**LAPORAN PRAKTIKUM ALGOTIRMA DAN STRUKTUR DATA**

**JOBSHEET 6 SEARCHING**

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



Muhammad Afiq Firdaus

2341760189 / 21

SIB-1E

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

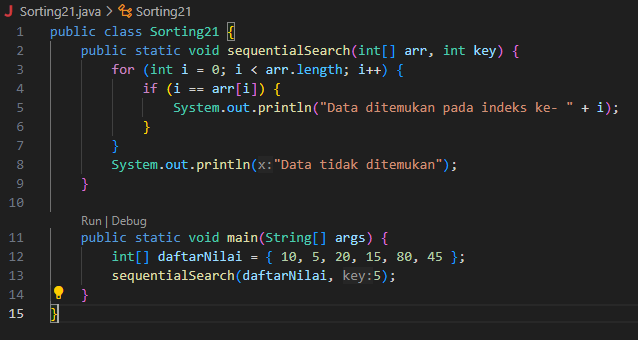
**POLITEKNIK NEGERI MALANG**

**2024**

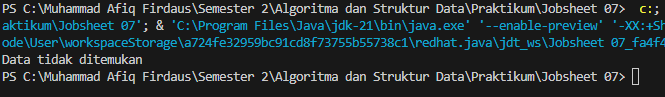
**6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search**

**6.2.1 Sequential Search Menggunakan Array**

Berikut adalah hasil code berdasarkan jobsheet Algoritma Sequential Search

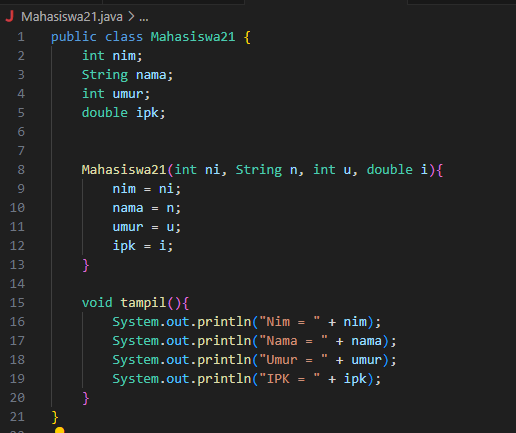


Berikut adalah hasil run dari code diatas

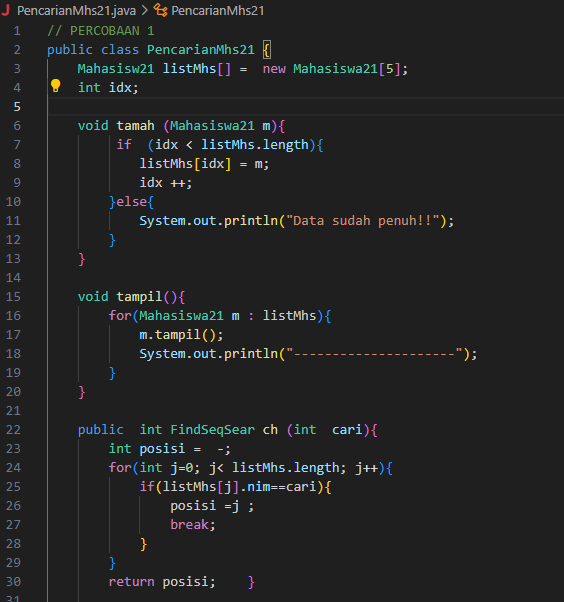


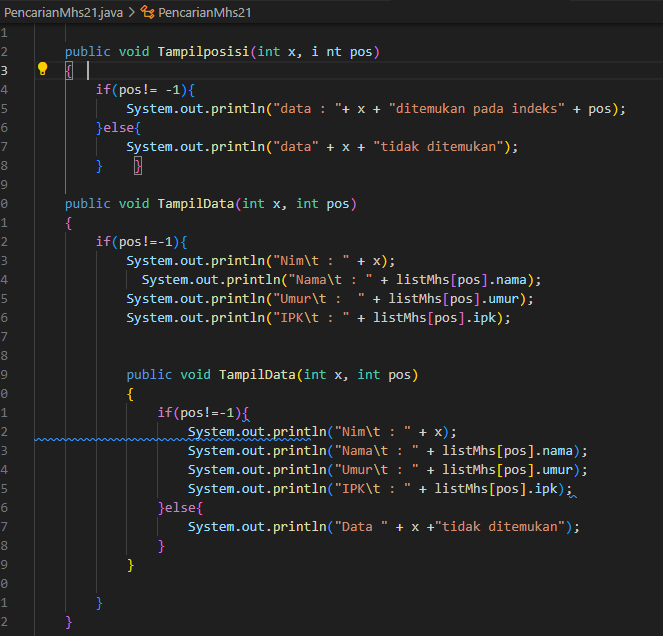
**6.2.2 Sequential Search Menggunakan Array of Object**

Berikut adalah code class Mahasiswa

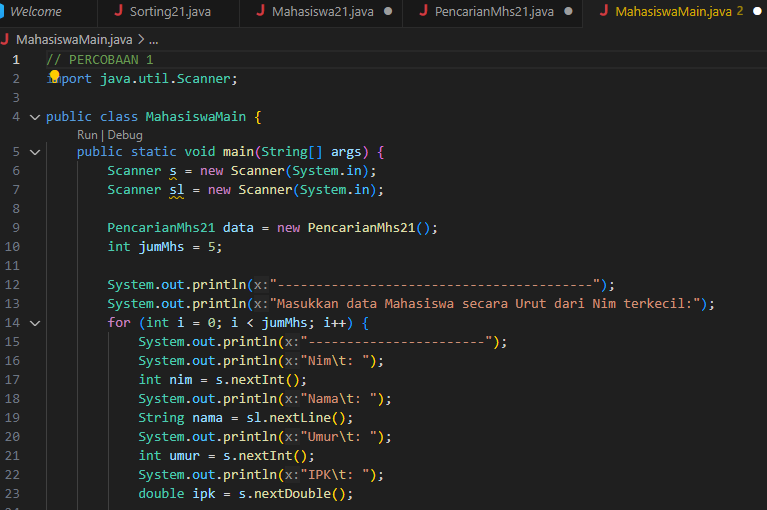


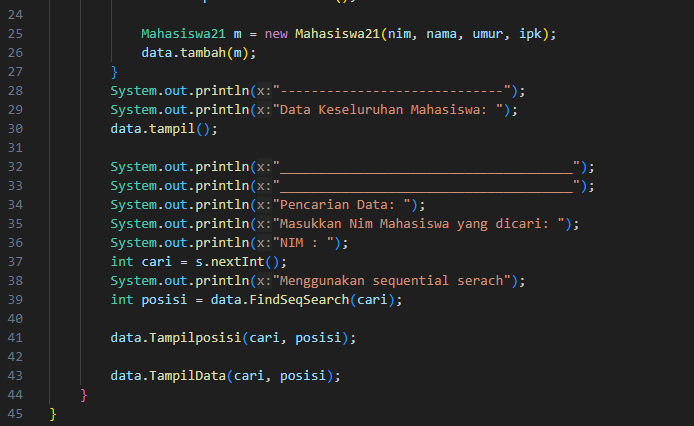
Berikut adalah code class pencarianMhs



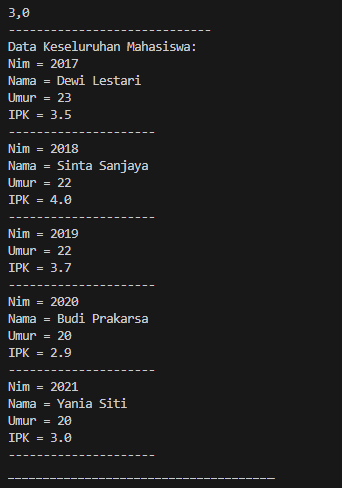
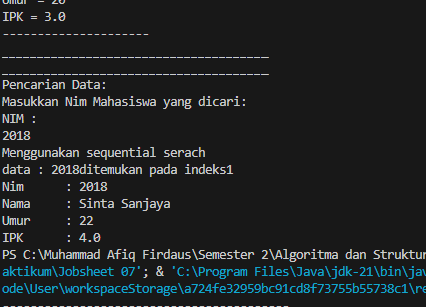


Berikut adalah code MahasiswaMain



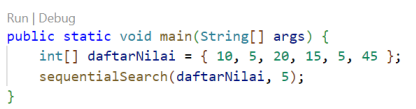


Dan berikut adalah hasil run dari code-code diatas

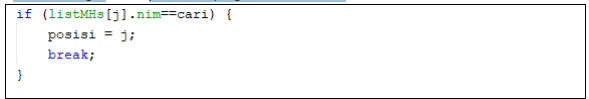


**Pertanyaan :**

1. Lakukan perubahan array daftarNilai pada fungsi main()



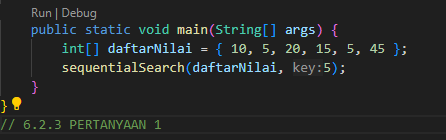
1. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs
2. Jelaskan fungsi break pada kode program dibawah ini!



1. 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.



2. perbedaan utama antara kedua metode tersebut adalah dalam informasi yang ditampilkan. Metode TampilData menampilkan seluruh informasi mahasiswa yang ditemukan, sedangkan metode Tampilposisi hanya menampilkan indeks posisi di mana data ditemukan.

3. Fungsi dari break; adalah untuk menghentikan iterasi loop secara paksa segera setelah suatu elemen yang memenuhi kondisi pencarian ditemukan. Ini menghindarkan loop dari melanjutkan iterasi ke elemen berikutnya setelah elemen yang sesuai dengan kriteria pencarian telah ditemukan, karena sudah tidak perlu lagi mencari. Dengan demikian, break; mengoptimalkan kinerja dan menghindari pemrosesan yang tidak perlu dalam pencarian.

4. Ya, program ini akan tetap berjalan meskipun data NIM yang dimasukkan tidak terurut dari kecil ke besar. Namun, hasil yang dikeluarkan mungkin tidak akan selalu benar.

Alasan program masih dapat berjalan adalah karena pencarian dilakukan secara sequential (berurutan) melalui seluruh elemen array listMhs untuk mencari NIM yang cocok dengan nilai yang dicari. Namun, jika data NIM tidak terurut, pencarian sequential ini mungkin memerlukan lebih banyak iterasi daripada jika data terurut, karena harus memeriksa setiap elemen dari awal hingga akhir array.

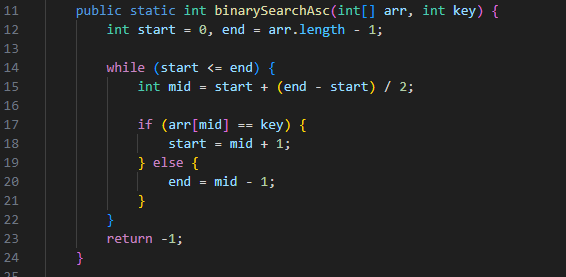
Ketika data tidak terurut, hasil dari pencarian tergantung pada posisi mana elemen yang sesuai ditemukan. Jika elemen yang sesuai berada di posisi awal array, hasilnya akan benar. Namun, jika elemen yang sesuai berada di posisi akhir array, hasilnya juga akan benar. Namun, jika elemen yang sesuai berada di tengah array atau tidak ada dalam array, hasilnya akan salah.

Jadi, meskipun program masih dapat berjalan, keakuratannya tergantung pada posisi elemen yang sesuai dalam array, dan jika data tidak terurut, hasilnya mungkin tidak selalu benar.

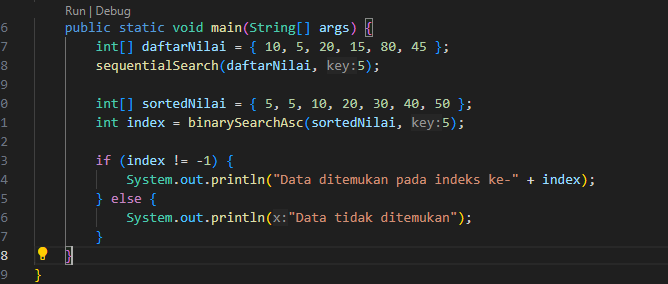
**6.3. Searching / Pencarian Menggunakan Binary Search**

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

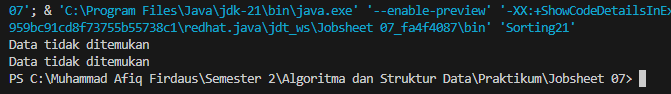
1. Tambahkan method binarySearchAsc() pada file Sorting.java



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

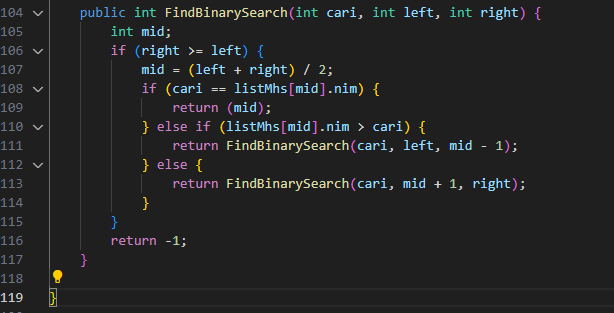


1. Run dan compile program

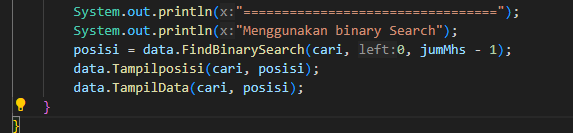


**6.3.2. Langkah-langkah Percobaan Binary Search menggunakan Array of Object**

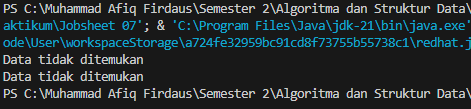
1. Pada percobaan 6.2.2 (sequential search) tambahkan method FindBinarySearch bertipe integer pada class PencarianMhs. Kemudian Deklarasikan isi method FindBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching.



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

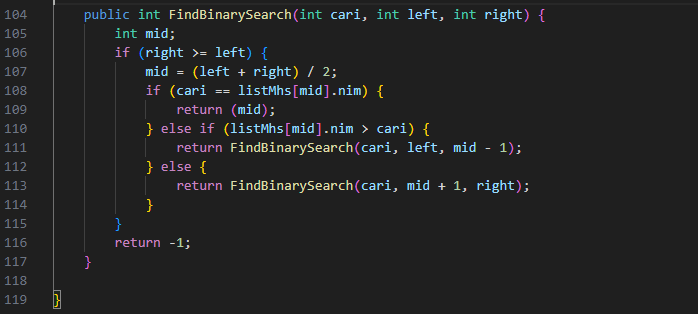


3. Jalankan dan amati hasilnya.

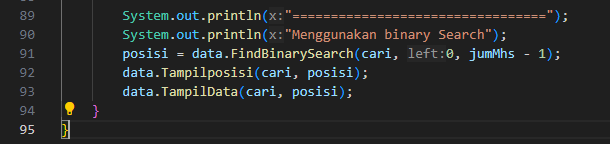


**6.3.2. Langkah-langkah Percobaan Binary Search menggunakan Array of Object**

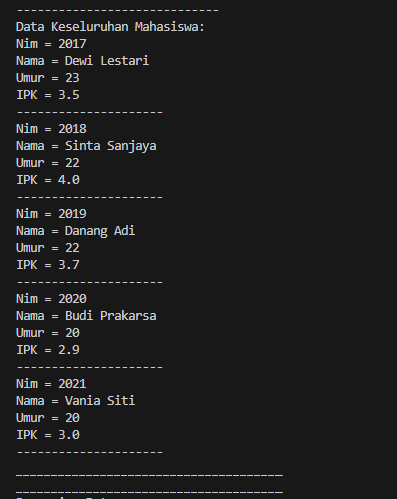
1. Pada percobaan 6.2.2 (sequential search) tambahkan method FindBinarySearch bertipe integer pada class PencarianMhs. Kemudian Deklarasikan isi method FindBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching.

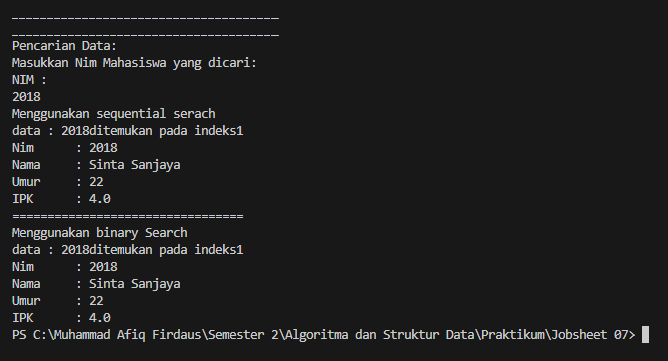


2. Panggil method FindBinarySearch terdapat pada class PencarianMhs di kelas Mahasiswamain. Kemudia panggil method tampilposisi dan tampilData



**6.3.2. Verifikasi Hasil Percobaan**

****

****

**6.3.3. Pertanyaan**

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

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

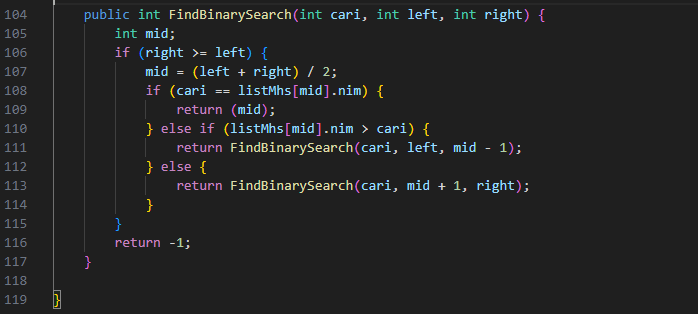
3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!

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. FindBinarySearch pada class PencarianMhs21. Berikut adalah potongan kode yang menunjukkan proses divide:



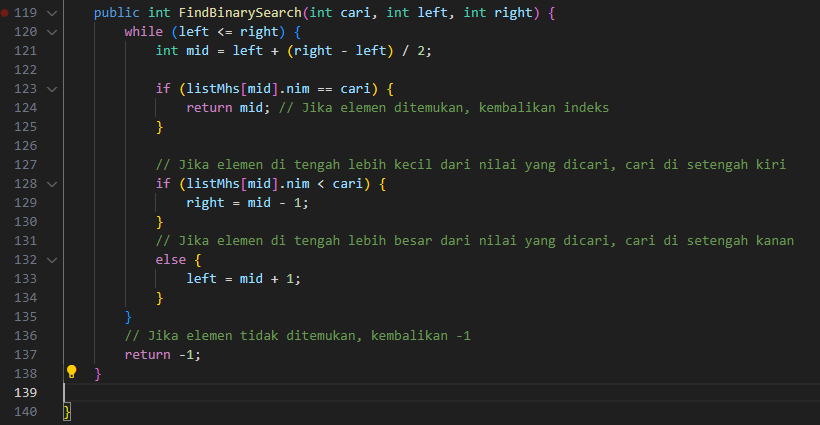
1. Proses conquer (penggabungan) dalam algoritma binary search dijalankan pada bagian berikut dari method FindBinarySearch pada class PencarianMhs21



1. Ya, program akan tetap berjalan meskipun data NIM yang dimasukkan tidak terurut. Namun, hasil yang dikeluarkan mungkin tidak akan selalu benar.
2. Jika data NIM yang dimasukkan sudah terurut dari terbesar ke terkecil, dan elemen yang dicari adalah 2020, hasil dari binary search akan tergantung pada implementasi algoritma pencarian. Dalam kasus ini, kita akan menggunakan implementasi binary search yang diberikan dalam kode program.

Namun, dalam implementasi yang diberikan dalam kode program, binary search tidak akan memberikan hasil yang benar jika data tidak terurut. Karena data yang dimasukkan sudah terurut dari terbesar ke terkecil, binary search kemungkinan besar akan memberikan hasil yang sesuai, yaitu menemukan bahwa elemen 2020 ada dalam array.

Berikut adalah hasil modifikasi berdasarkan code diatas



1. Berikut adalah hasil modifikasi dari class main yang mana jumlah mahasiswa yang di input sesuai berdasarkan masukan keyboard

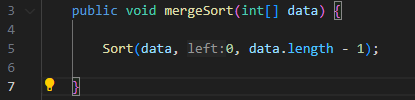
**6.4. Percobaan Pengayaan Divide and Conquer**

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

Tambahkan class MergeSorting pada package tersebut



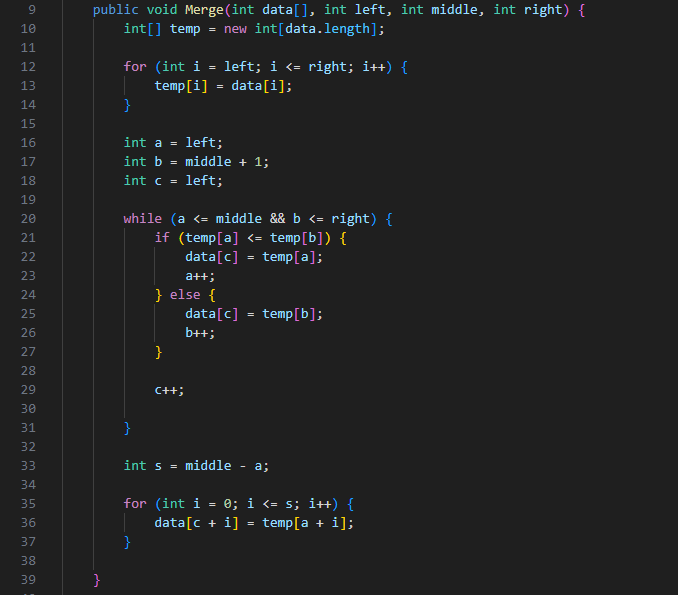
Pada class MergeSorting buatlah method mergeSort yang menerima parameter data array yang akan diurutkan



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



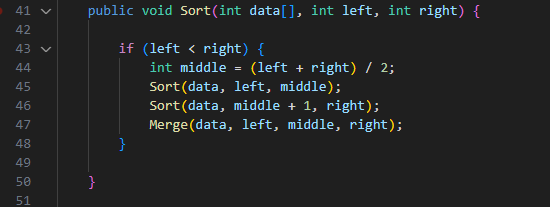
Implementasikan proses merge sebagai berikut.



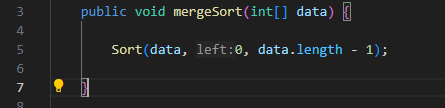
Buatlah method sort



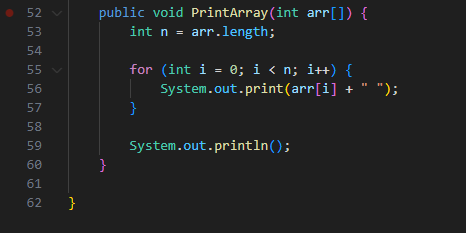
Implementasikan kode berikut pada method sort



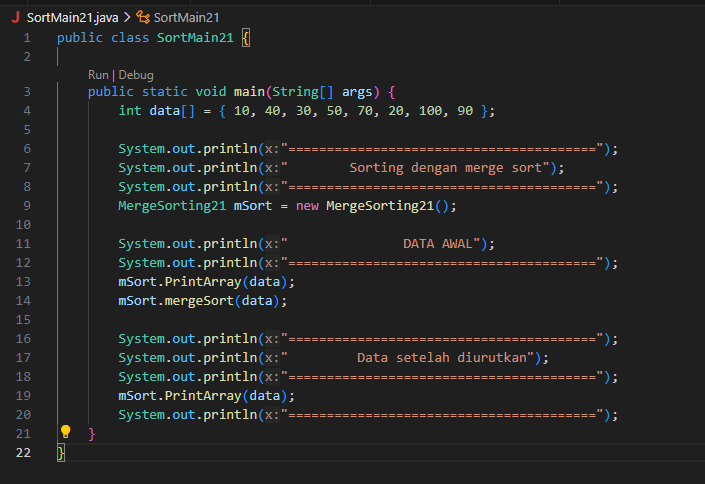
Pada method mergeSort, panggil method sort dengan parameter data yang ingin diurutkan serta range data awal sampai dengan akhir.



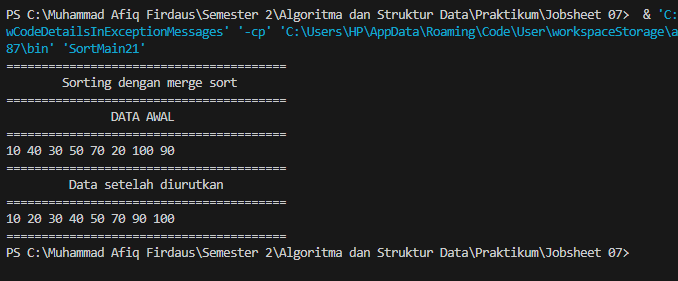
Tambahkan method printArray



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



**6.4.2. Verifikasi Hasil Percobaan**



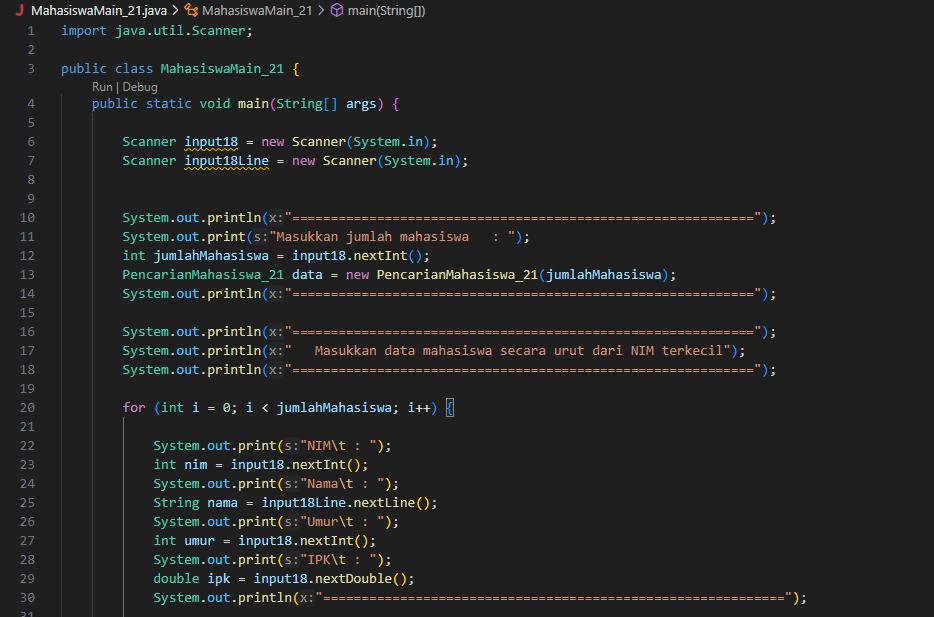
**6.5. Latihan Praktikum**

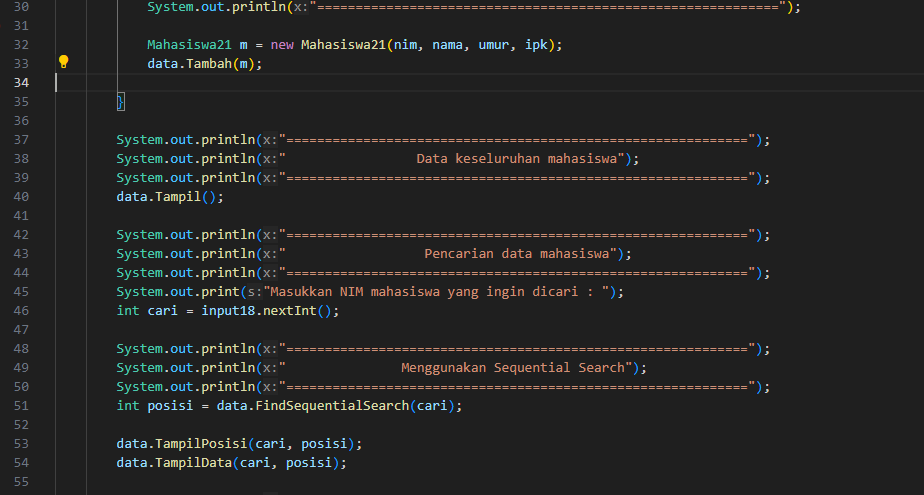
1. Modifikasi percobaan searching diatas yang menggunakan Searching array of object dengan ketentuan berikut ini

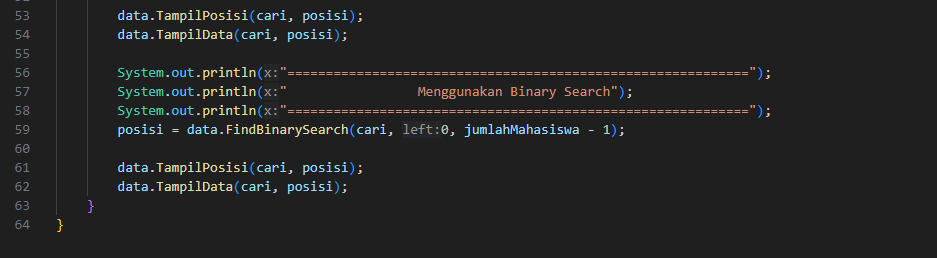
- Pencarian dilakukan berdasarkan Nama Mahasiswa (gunakan Algoritma binary Search)

- Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan

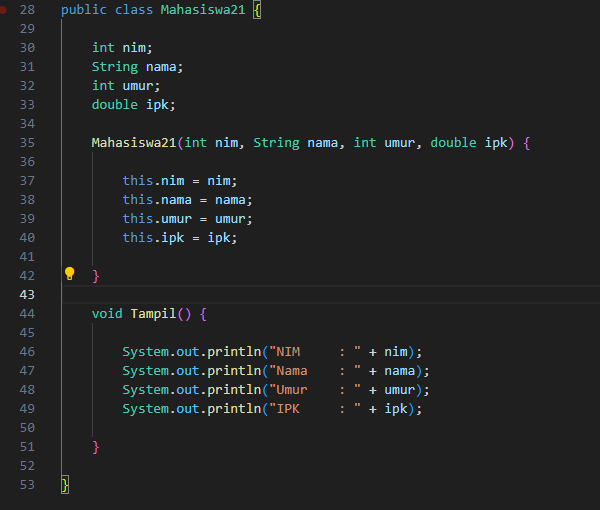
* **MahasiswaMain\_21.java**



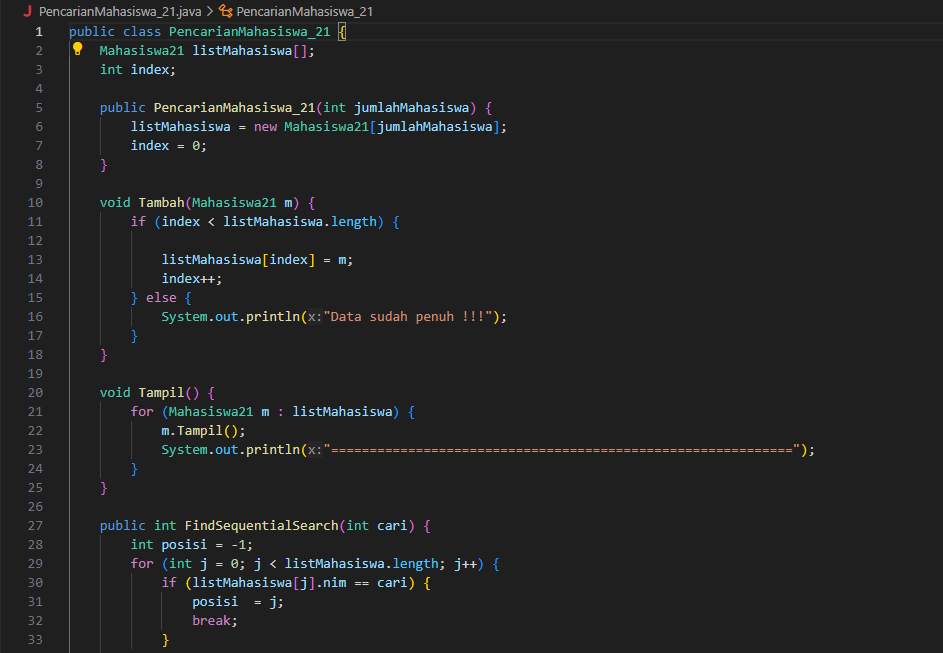




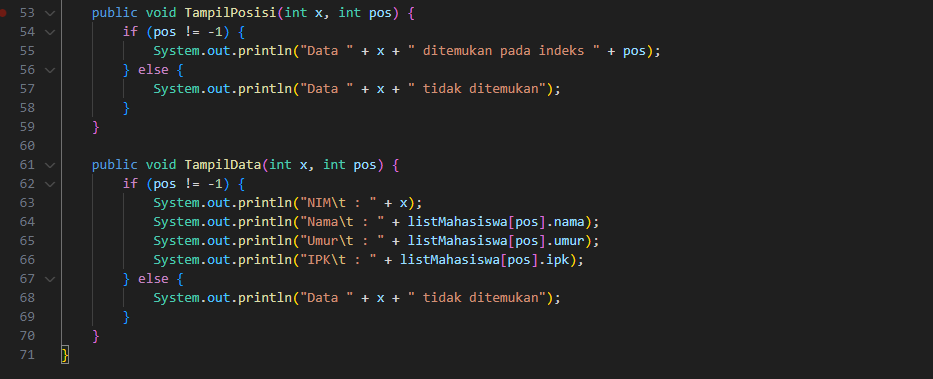
* **Mahasiswa21.java**

****

* **PencarianMahasiswa\_21.java**







* **OutPut**

