Nama : Dicky Darmawan

Kelas : TI 1B

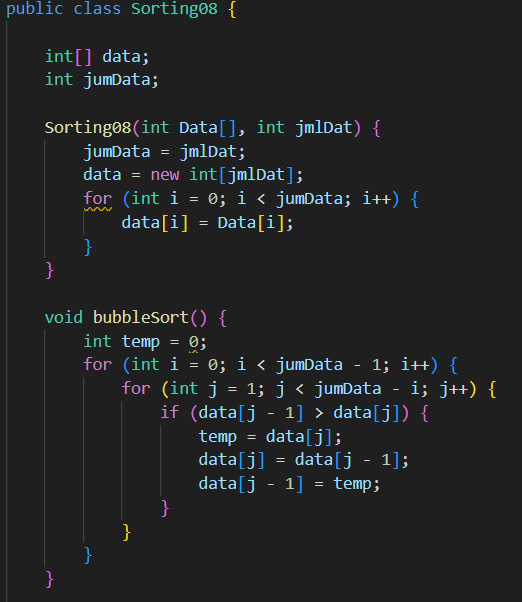
NIM : 244107020037

**Praktikum 1 - Mengimplementasikan Sorting menggunakan object**

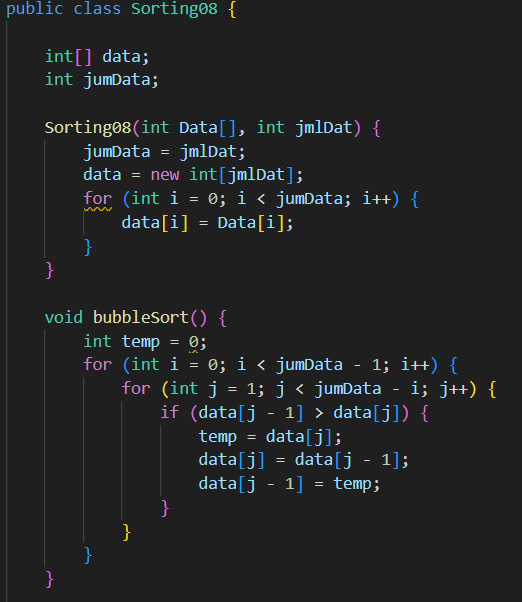
**Langkah Praktikum 1**

a. SORTING – BUBBLE SORT

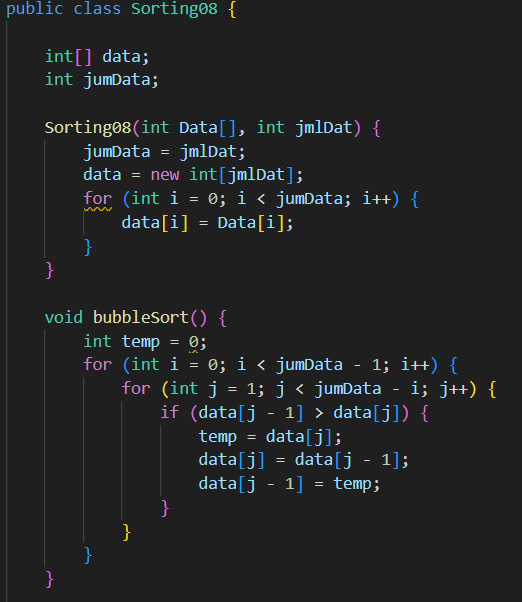
1. Buat folder baru bernama Jobsheet6 di dalam repository Praktikum ASD
2. Buat class Sorting<No Presensi>, kemudian tambahkan atribut sebagai berikut:



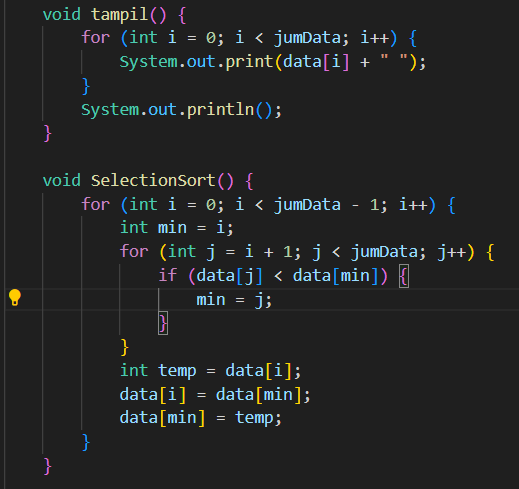
1. Buatlah konstruktor dengan parameter Data[] dan jmlDat



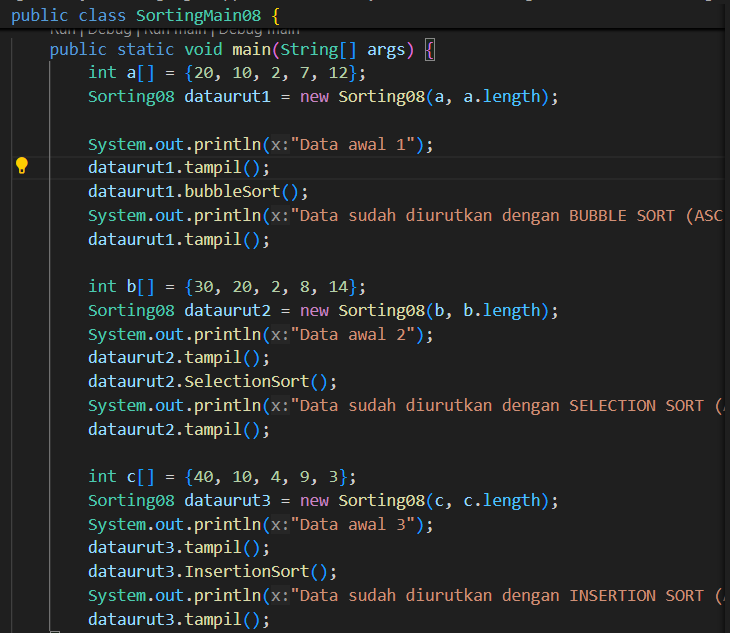
1. Buatlah method bubbleSort bertipe void dan deklarasikan isinya menggunakan algoritma Bubble Sort.



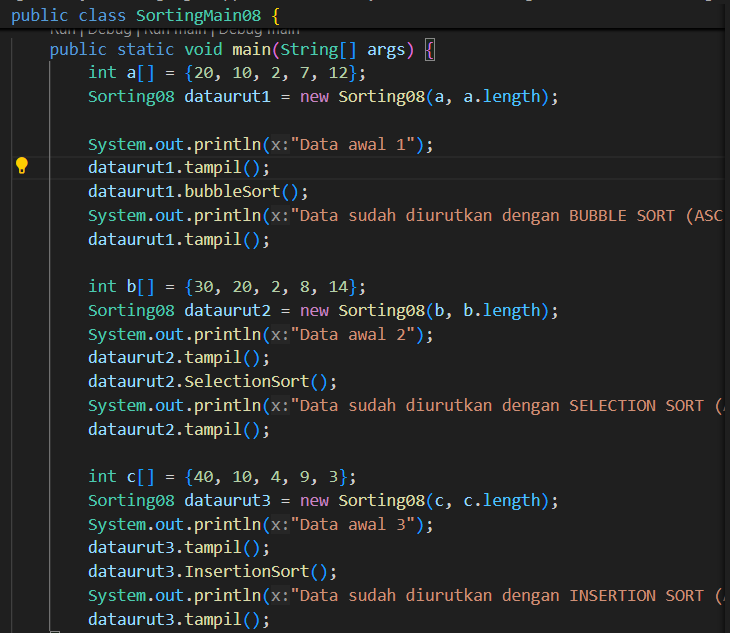
1. Buatlah method tampil bertipe void dan deklarasikan isi method tersebut.



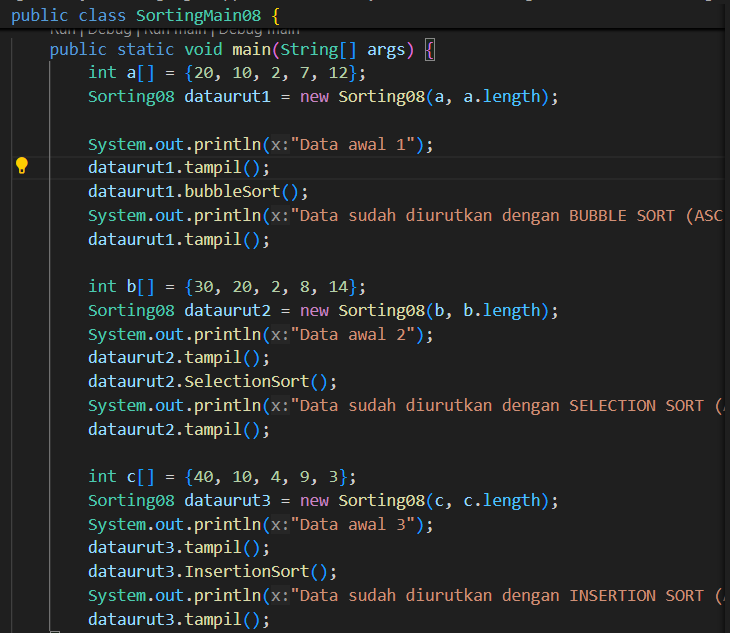
1. Buat class SortingMain<No Presensi> kemudian deklarasikan array dengan nama a[] kemudian isi array tersebut



1. Buatlah objek baru dengan nama dataurut1 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya

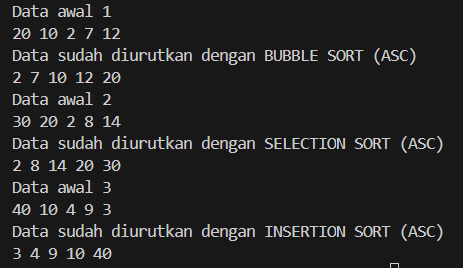


1. Lakukan pemanggilan method bubbleSort dan tampil



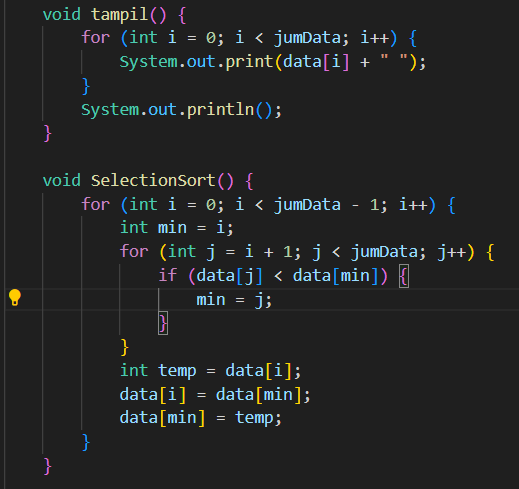
1. Jalankan program, dan amati hasilnya!

**Verifikasi Hasil Percobaan**

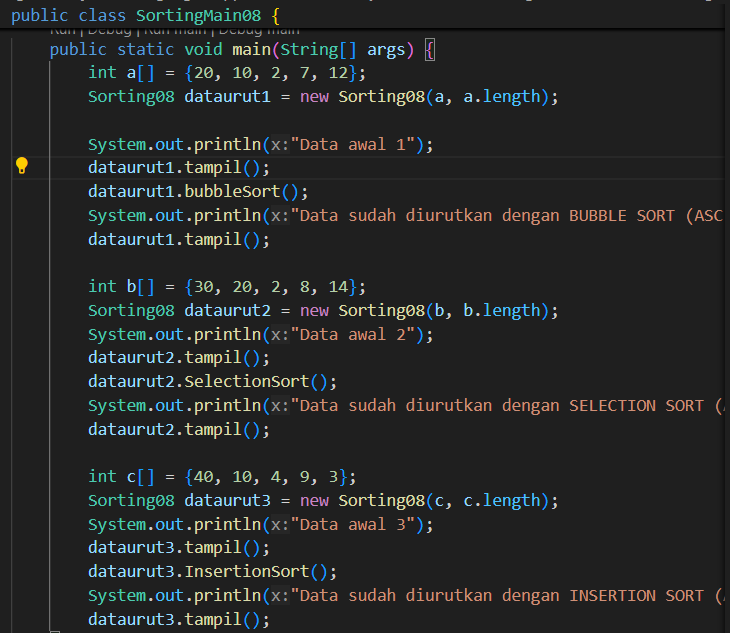
****

b. SORTING – SELECTION SORT

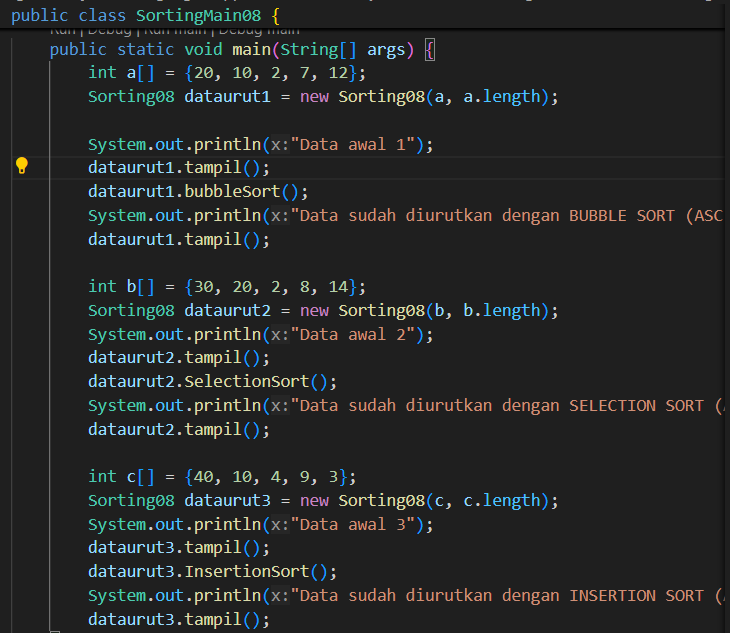
1. Pada class Sorting<No Presensi> yang sudah dibuat di praktikum sebelumnya tambahkan method SelectionSort yang mengimplementasikan pengurutan menggunakan algoritma selection sort.



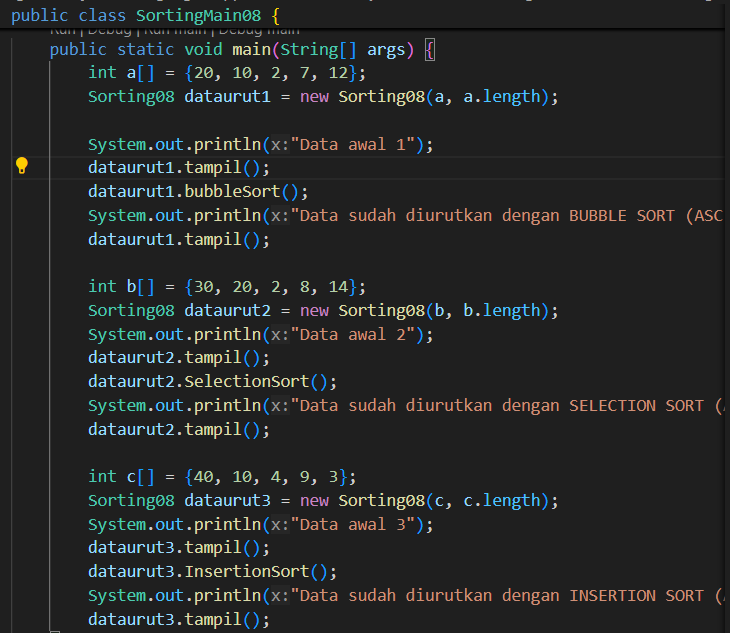
1. Deklarasikan array dengan nama b[] pada kelas SortingMain<No Presensi> kemudian isi array tersebut



1. Buatlah objek baru dengan nama dataurut2 yang merupakan instansiasi dari classSorting, kemudian isi parameternya

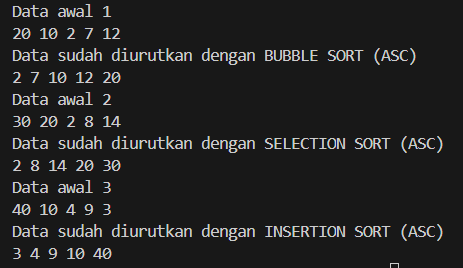


1. Lakukan pemanggilan method SelectionSort dan tampil



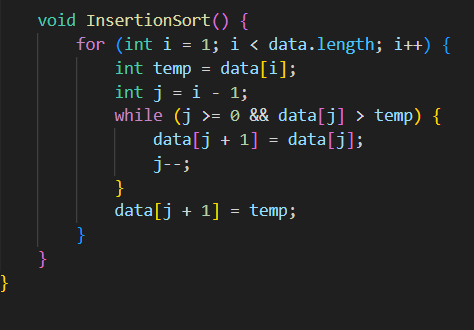
1. Jalankan program dan amati hasilnya!

**Verifikasi Hasil Percobaan**

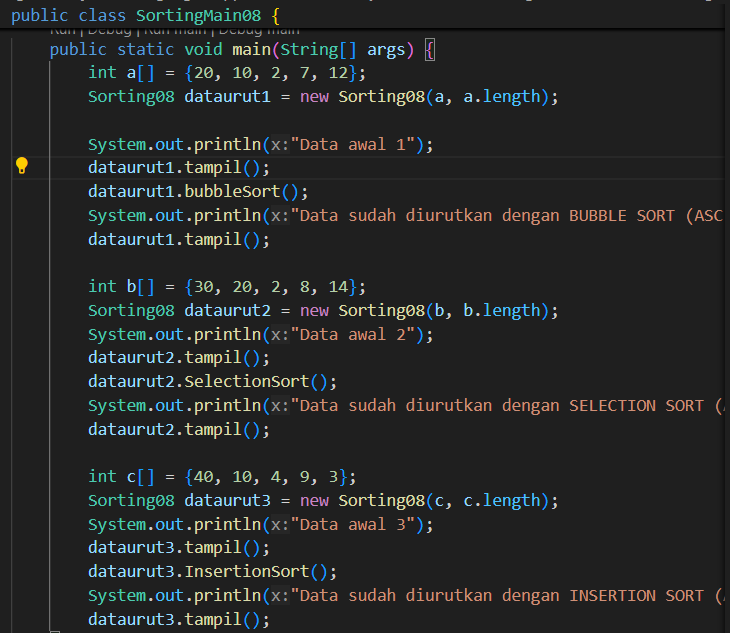


c. SORTING – INSERTION SORT

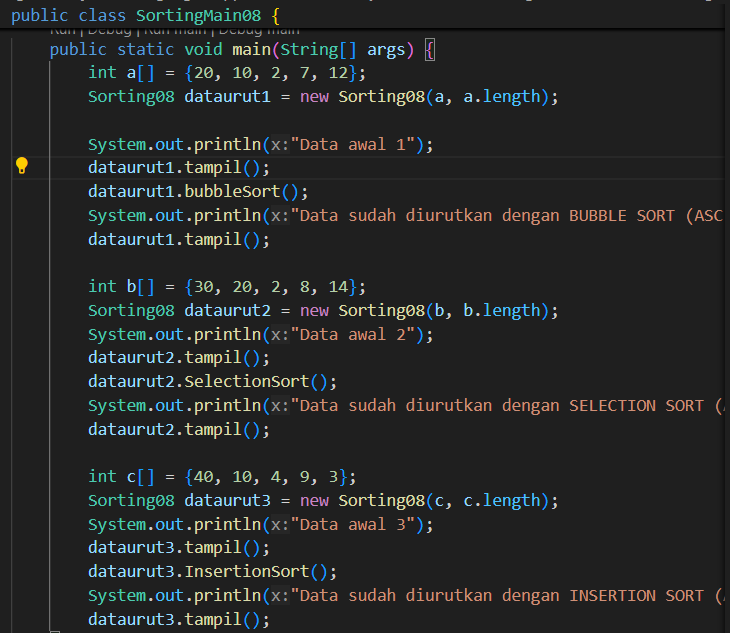
1. Pada class Sorting<No Presensi> yang sudah dibuat di praktikum sebelumnya tambahkan method insertionSort yang mengimplementasikan pengurutan menggunakan algoritma insertion sort.



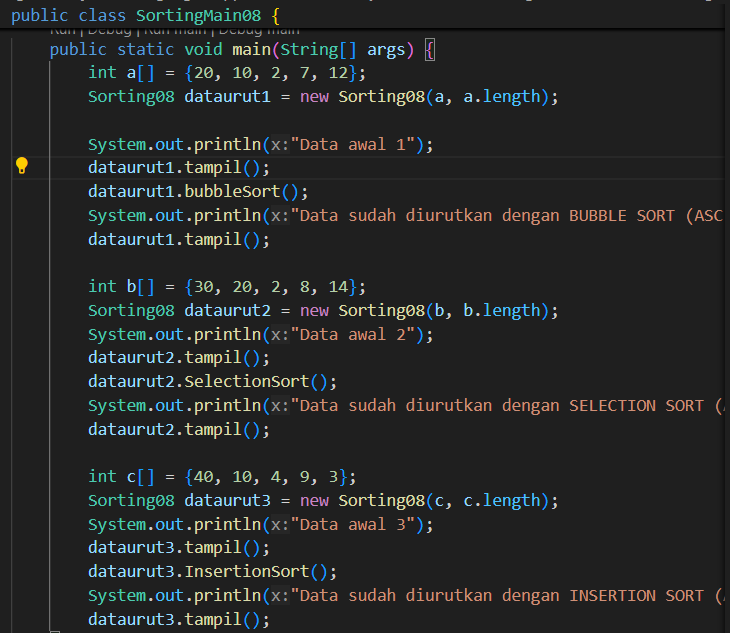
1. Deklarasikan array dengan nama c[] pada kelas SortingMain<No Presensi> kemudian isi array tersebut



1. Buatlah objek baru dengan nama dataurut3 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya

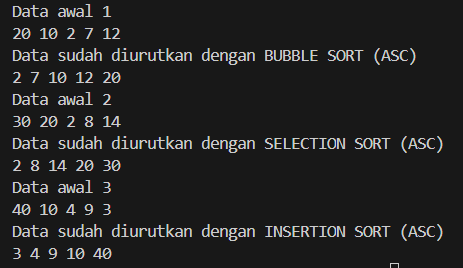


1. Lakukan pemanggilan method insertionSort dan tampil



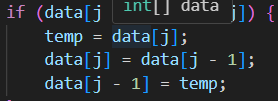
1. Jalankan program dan amati hasilnya!

**Verifikasi Hasil Percobaan**



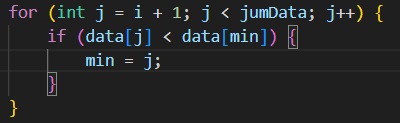
**Pertanyaan**

1. Jelaskan fungsi kode program tersebut!



if (data[j - 1] > data[j]), mengecek apakah elemen sebelumnya (data[j - 1]) lebih besar daripada elemen saat ini (data[j]). jika benar, maka dilakukan pertukaran (swap) antara data[j] dan data[j - 1].

1. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!



1. Pada Insertion sort, jelaskan maksud dari kondisi pada perulangan while (j>0 && data[j]>temp)

Selama j lebih besar dari 0 dan data[j] lebih besar dari temp maka program di dalam while akan terus dijalankan

1. Pada Insertion sort, apakah tujuan dari perintah data[j+1] = data[j];

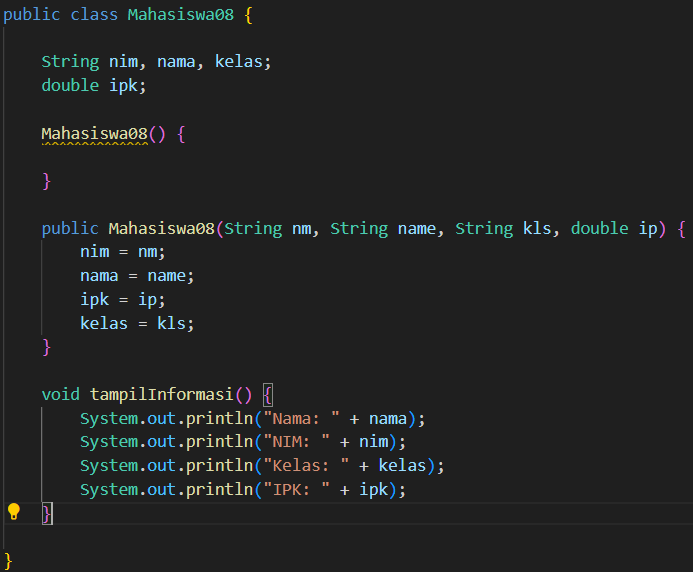
Untuk menggeser elemen ke kanan agar ada tempat kosong untuk menyisipkan elemen yang sedang diproses

**Praktikum 2- (Sorting Menggunakan Array of Object)**

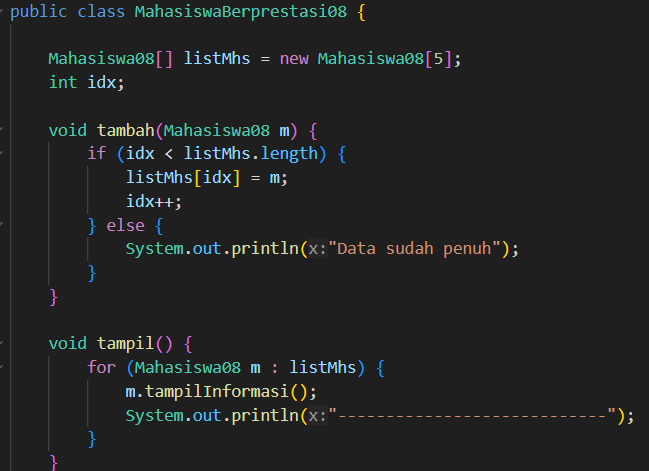
**Langkah Praktikum 2 - Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort)**

**Langkah-langkah Praktikum 2**

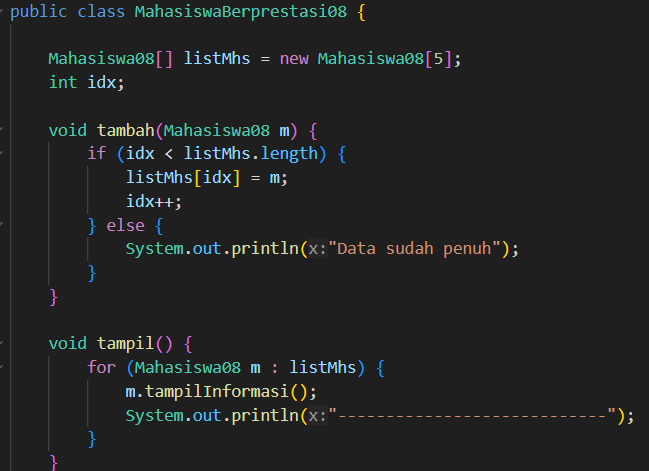
1. Buatlah class dengan nama Mahasiswa<No Presensi>.
2. Untuk lebih jelasnya class tersebut dapat dilihat pada potongan kode di bawah ini



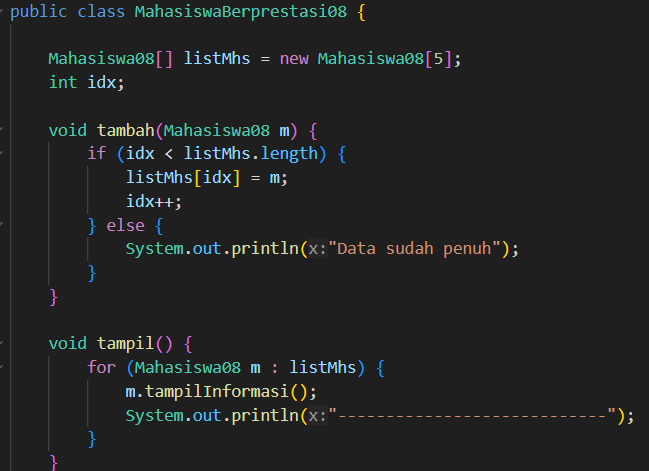
1. Buat class MahasiswaBerprestasi<No Presensi> seperti di bawah ini!



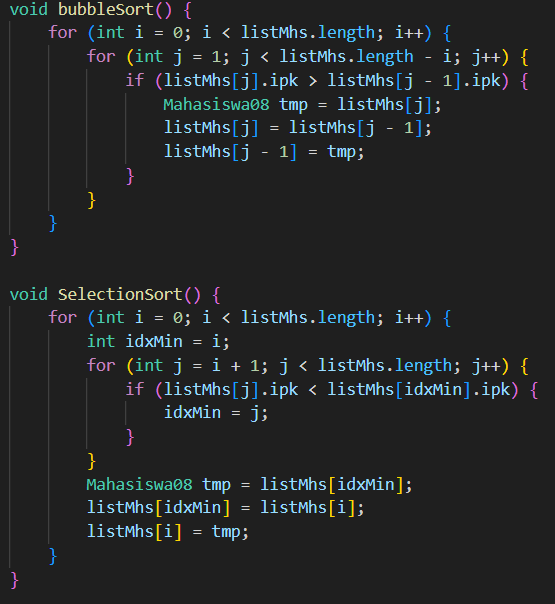
1. Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.



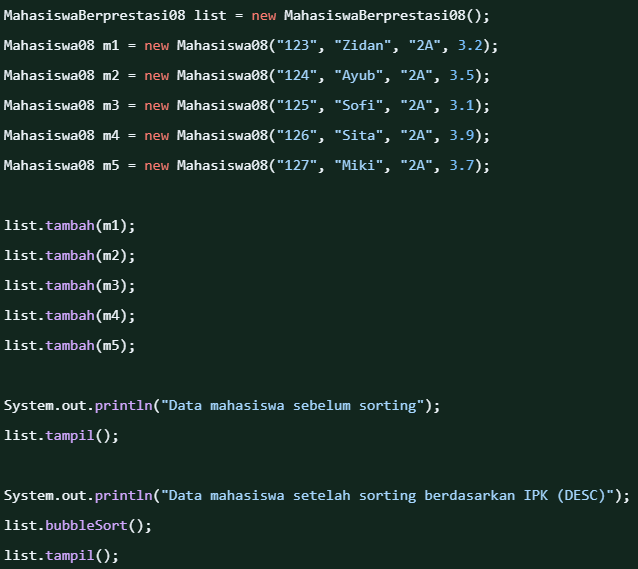
1. Tambahkan method tampil() di dalam class tersebut! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.



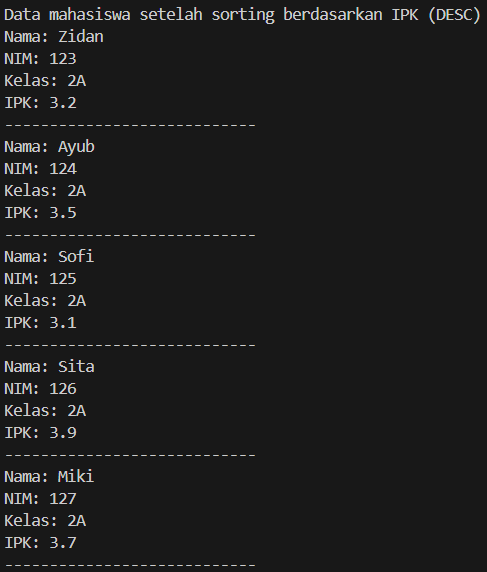
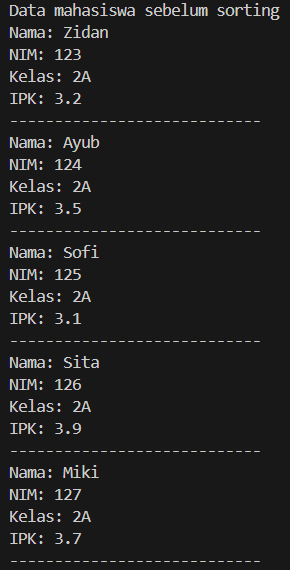
1. Tambahkan method bubbleSort() di dalam class tersebut!



1. Buat class MahasiswaDemo<No Presensi>, kemudian buatlah sebuah objek MahasiswaBerprestasi dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek MahasiswaBerprestasi. Silakan dipanggil fungsi tampil() untuk melihat semua data yang telah dimasukan, urutkan data tersebut dengan memanggil fungsi bubbleSort() dan yang terakhir panggil fungsi tampil kembali.



**Verifikasi Hasil Percobaan**



**Pertanyaan**

1. Perhatikan perulangan di dalam bubbleSort() di bawah ini:
2. Mengapa syarat dari perulangan i adalah i<listMhs.length-1 ?

Karena setelah n-1 tahap array sudah pasti terurut, maka cukup melakukan perulangan hingga i < listMhs.length - 1. Jika jumlah data dalam array adalah 5, maka hanya perlu 4 tahap sorting, karena elemen terakhir secara otomatis sudah berada di posisi yang benar.

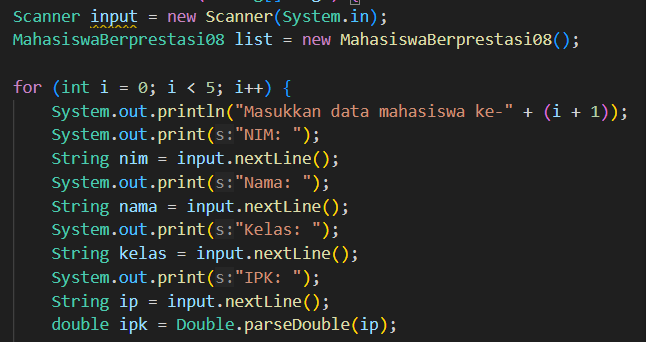
1. Mengapa syarat dari perulangan j adalah j<listMhs.length-i ?

Karena jumlah perbandingan berkurang setiap tahap, sehingga batas perulangan j dikurangi dengan nilai i

1. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?

Perulangan i akan berlangsung sebanyak 49 kali (karena i < 50 - 1)

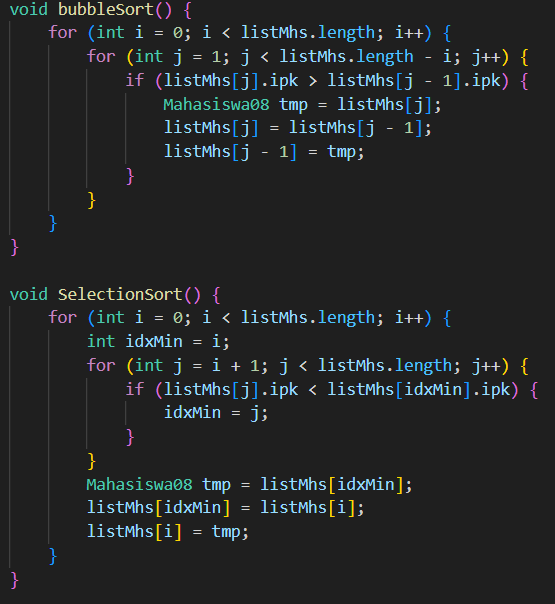
1. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyborad) yang terdiri dari nim, nama, kelas, dan ipk!



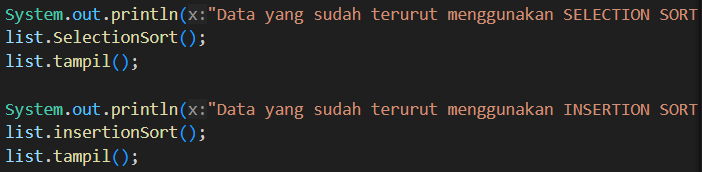
**Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)**

**Langkah-langkah Percobaan**

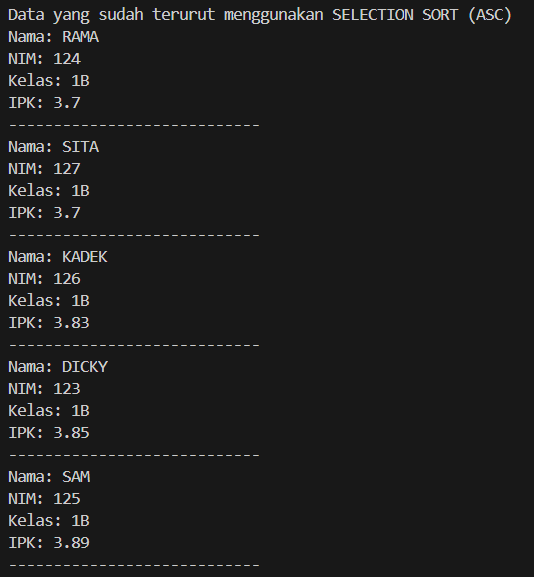
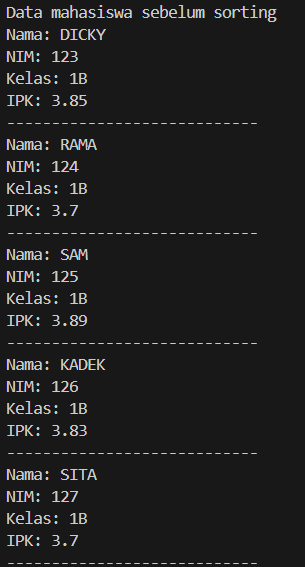
1. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method selectionSort() di dalamnya! Method ini juga akan melakukan proses sorting secara ascending, tetapi menggunakan pendekatan selection sort.



1. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method selectionSort() tersebut, kemudian panggil method tampil() untuk menampilkan data yang sudah diurutkan!Coba jalankan kembali class MahasiswaDemo, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?

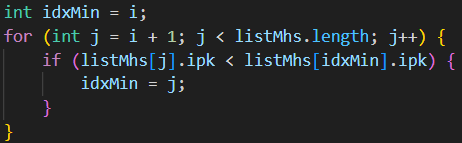


**Verifikasi Hasil Percobaan**



**Pertanyaan**

Di dalam method selection sort, terdapat baris program seperti di bawah ini:



Untuk apakah proses tersebut, jelaskan!

Int idxMIn = I, menyimpan indeks dari elemen minimum (nilai IPK terkecil) yang ditemukan. indeks elemen pertama dalam bagian yang belum terurut (i) dianggap sebagai elemen terkecil.

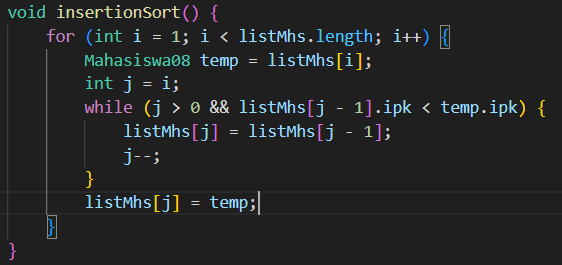
for (int j = i + 1; j < listMhs.length; j++), melakukan perulangan untuk mencari nilai terkecil dari array. j = i + 1 karena elemen indeks i sudah dianggap sebagai nilai minimum sementara.

if (listMhs[j].ipk < listMhs[idxMin].ipk), jika ditemukan elemen yang lebih kecil dari listMhs[idxMin].ipk maka akan elemen j akan diperbarui ke idxMin.

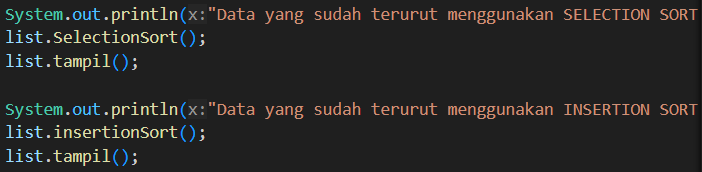
**Mengurutkan Data Mahasiswa Berdasarkan IPK menggunakan Insertion Sort**

**Langkah-langkah Percobaan**

1. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method insertionSort() di dalamnya. Method ini juga akan melakukan proses sorting secara ascending, tetapi menggunakan pendekatan Insertion Sort.

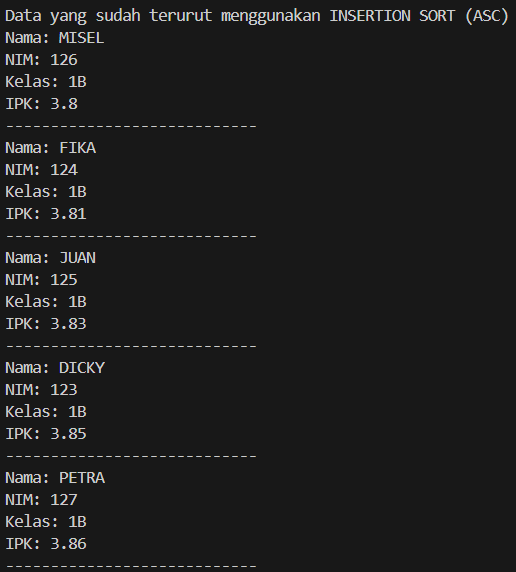
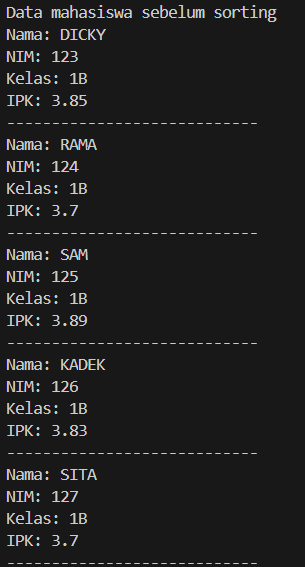


1. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method insertionSort() dan tampil () tersebut!



1. Coba jalankan kembali class MahasiswaDemo, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?

**Verifikasi Hasil Percobaan**



**Pertanyaan**

Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending.

