**JOBSHEET - 5**

**SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)**

1. **Tujuan Praktikum**

Setelah melakukan praktikum ini diharapkan mahasiswa mampu:

1. Mahasiswa mampu  membuat algoritma searching bubble sort, selection sort dan insertion sort
2. Mahasiswa mampu  menerapkan algoritma searching bubble sort, selection sort dan insertion sort pada program
3. **Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Bubble Sort**

**Waktu : 50 menit**

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

|  |
| --- |
| Mahasiswa |
| nama: String  thnMasuk: int  umur: int  ipk: double |
| Mahasiswa(n: String, t: int, u: int, i: double)  tampil(): void |

Berdasarkan class diagram di atas, kita akan membuat sebuah 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.

|  |
| --- |
| DaftarMahasiswaBerprestasi |
| listMhs: Mahasiswa[5]  idx: int |
| tambah(mhs: Mahasiswa): void  tampil(): void  bubbleSort(): 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, dan juga untuk mengurutkan menggunakan Teknik bubble sort berdasarkan nilai IPK mahasiswa.

1. **Langkah-langkah Percobaan**
   1. Buat project baru dengan nama “bubble-selection-insertion”, kemudian buat package dengan nama “jobsheet6”.
   2. Buatlah sebuah class dengan nama Mahasiswa
   3. Sesuaikan class Mahasiswa dengan melihat class diagram di atas dengan menambahkan attribute, konstruktor, dan fungsi atau method. Untuk lebih jelasnya class tersebut dapat dilihat pada potongan kode di bawah ini

Graphical user interface, text

Description automatically generated

* 1. Buat class DaftarMahasiswaBerprestasi seperti di bawah ini!

Graphical user interface, text, application

Description automatically generated

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

Graphical user interface, text, application

Description automatically generated

* 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.

Text

Description automatically generated with medium confidence

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

Text

Description automatically generated

* 1. Buat class Main dan didalamnya buat method main() seperti di bawah ini!

Graphical user interface, text, application

Description automatically generated

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

Text

Description automatically generated with medium confidence

1. **Verifikasi Hasil Percobaan**

Cocokan hasilnya dengan yang terdapat pada tampilan di bawah ini

A picture containing text, receipt

Description automatically generated

A picture containing text, receipt

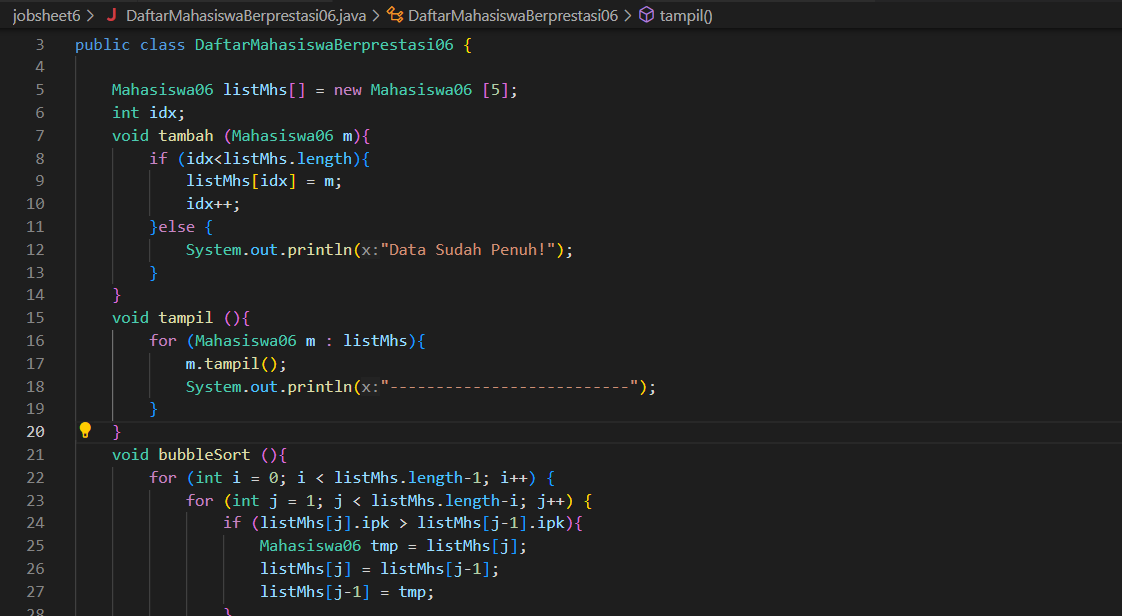
Description automatically generated

* **HASIL INPUT :**

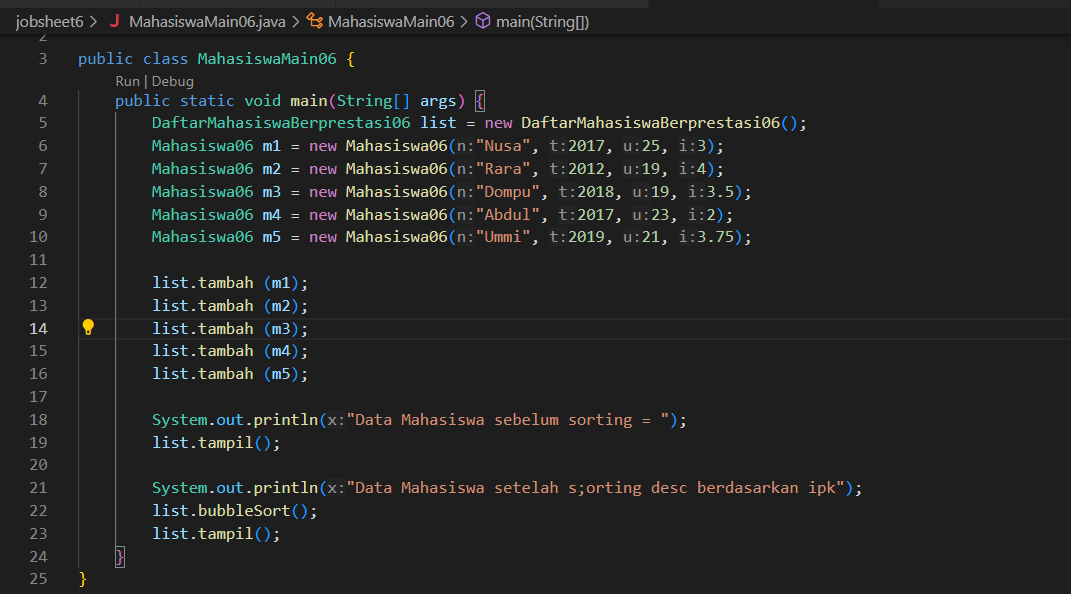
**Class Mahasiswa**

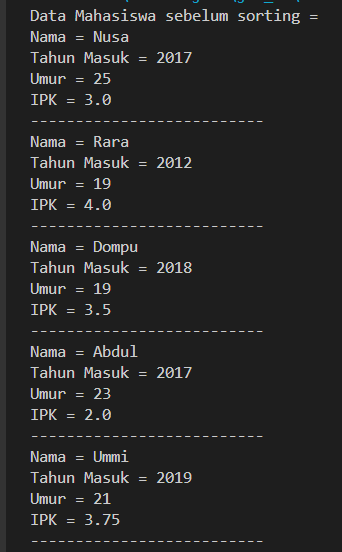


**Class Daftar Mahasiswa Berprestasi**



**Class Main**



* **HASIL OUTPUT :**

1. **Pertanyaan**
2. **Terdapat di method apakah proses bubble sort?**

* Iya, method fungsi `bubbleSort()` merupakan proses bubble sort. bubble sort pada kode program tersebut diimplementasikan dalam dua loop bersarang. Loop pertama digunakan untuk melewati elemen-elemen dalam array, dan loop kedua digunakan untuk membandingkan elemen-elemen berdekatan dan menukar mereka jika diperlukan.

1. **Di dalam method bubbleSort(), terdapat baris program seperti di bawah ini:**

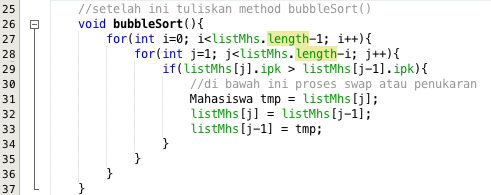
Text

Description automatically generated

**Untuk apakah proses tersebut?**

* Proses tersebut adalah langkah pertukaran (swap) dalam bubble sort nilai ipk mahasiswa dari besar ke kecil, langkah ini bertujuan untuk memastikan bahwa elemen yang memiliki nilai yang lebih besar `listMhs[j]` bergerak ke posisi yang lebih tinggi dalam array, sedangkan elemen yang memiliki nilai yang lebih kecil `listMhs[j-1]` bergerak ke posisi yang lebih rendah.

1. **Perhatikan perulangan di dalam bubbleSort() di bawah ini:**



* 1. **Apakah perbedaan antara kegunaan perulangan i dan perulangan j?**
* **Perulangan i** digunakan untuk iterasi luar yang menangani satu langkah pengurutan atau "lompatan" yang menempatkan elemen terbesar (atau terkecil) di ujung yang benar dari array
* **Perulangan j** digunakan untuk membandingkan pasangan elemen yang berdekatan dalam setiap iterasi i dan memeriksa dan menukar elemen-elemen yang tidak terurut.
  1. **Mengapa syarat dari perulangan i adalah i<listMhs.length-1 ?**
* Syarat dari perulangan i adalah i < listMhs.length - 1 karena dalam bubble sort, pada setiap iterasi dari perulangan i, dilakukan pengurangan satu pada bagian akhir dari array yang belum terurut.
  1. **Mengapa syarat dari perulangan j adalah j<listMhs.length-i ?**
* Syarat dari perulangan j dalam bubble sort adalah j < listMhs.length – i menghindari mempertimbangkan kembali elemen-elemen yang sudah diurutkan dalam setiap iterasi berikutnya, sehingga efisiensi algoritma bubble sort meningkat.
  1. **Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?**
* Jika jumlah data dalam listMhs adalah 50, maka perulangan i pada algoritma bubble sort akan berlangsung sebanyak 49 kali. Hal ini karena setiap iterasi luar (iterasi i) akan mengurangi jumlah elemen yang belum terurut di ujung array, dimulai dari 0 hingga 48. Pada iterasi ke-49, hanya ada satu elemen yang perlu dipertimbangkan, sehingga tidak perlu dilakukan iterasi lagi.

1. **Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Selection Sort**

**Waktu : 30 menit**

Jika pada praktikum yang sebelumnya kita telah mengurutkan data mahasiwa berdasarkan IPK menggunakan Bubble Sort secara descending, pada kali ini kita akan mencoba untuk menambahkan fungsi pengurutan menggunakan Selection Sort.

1. **Langkah-langkah Percobaan.**
2. Lihat kembali class DaftarMahasiswaBerprestasi, dan tambahkan method selectionSort() di dalamnya! Method ini juga akan melakukan proses sorting secara **ascending**, tetapi menggunakan pendekatan selection sort.

Text

Description automatically generated

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



1. Coba jalankan kembali class Main, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?
2. **Verifikasi Hasil Percobaan**

Pastikan output yang ditampilkan sudah benar seperti di bawah ini

A picture containing text, receipt

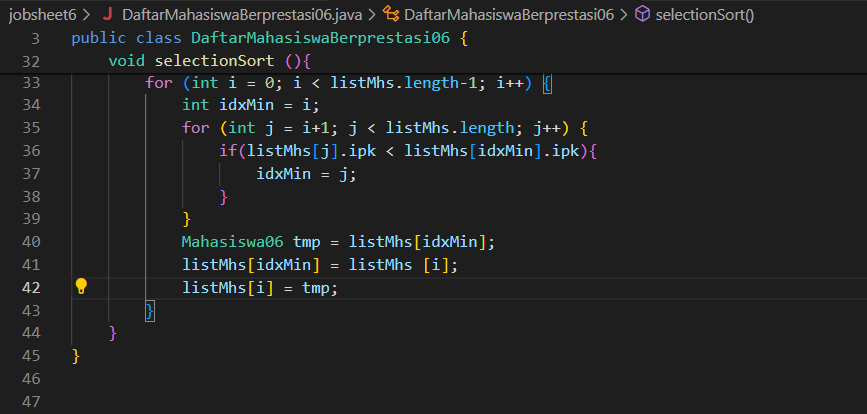
Description automatically generated

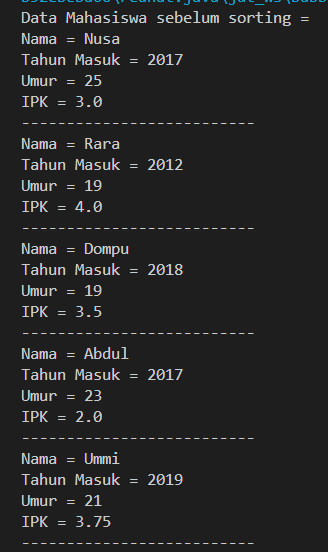
A picture containing text, receipt

Description automatically generated

* **HASIL INPUT :**

Menambahkan **method Selection Sort** pada Class **Daftar Mahasiswa Berprestasi**

****

* **HASIL OUTPUT**

1. **Pertanyaan**

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

Text

Description automatically generated

Untuk apakah proses tersebut, jelaskan!

* Proses tersebut adalah langkah selection sort yang digunakan untuk mencari indeks dari elemen terkecil dalam array yang belum diurutkan. (yaitu mengurutkan nilai ipk mahasiswa dari yang terkecil ke terbesar)

1. **Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort**

**Waktu : 30 menit**

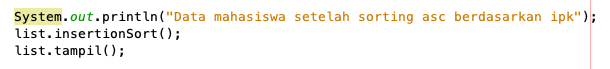
Yang terakhir akan diimplementasikan Teknik sorting menggunakan Insertion Sort, dengan mengurutkan IPK mahasiswa secara ascending.

1. **Langkah-langkah Percobaan**
2. Lihat kembali class DaftarMahasiswaBerprestasi, dan tambahkan method insertionSort() di dalamnya. Method ini juga akan melakukan proses sorting secara ***ascending***, tetapi menggunakan pendekatan Insertion Sort.

Text

Description automatically generated

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



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

* Iya, hasil dari method insertion sort ascending yaitu urut berdasarkan nilai IPK dari terkecil ke terbesar

1. **Verifikasi Hasil Percobaan**

Pastikan output yang ditampilkan sudah benar seperti di bawah ini

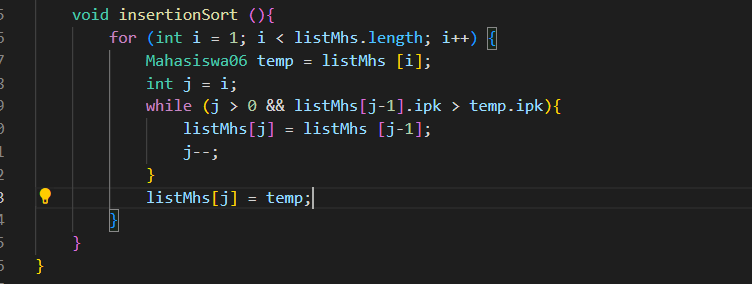
A picture containing text, receipt

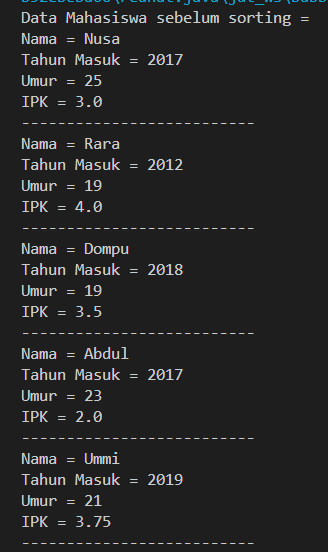
Description automatically generated

A picture containing text, receipt

Description automatically generated

* **HASIL INPUT :**

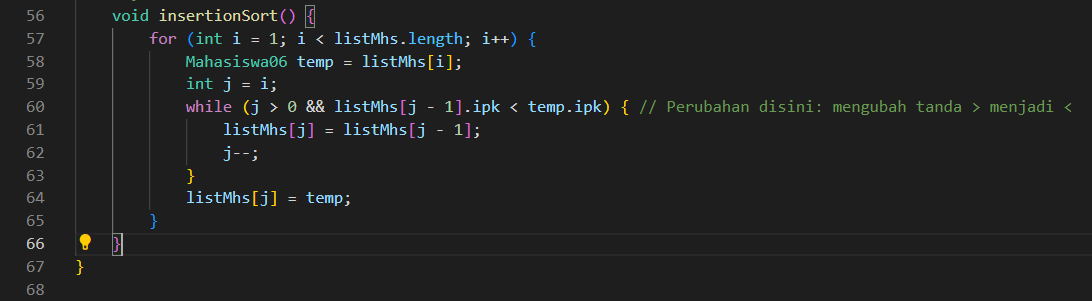


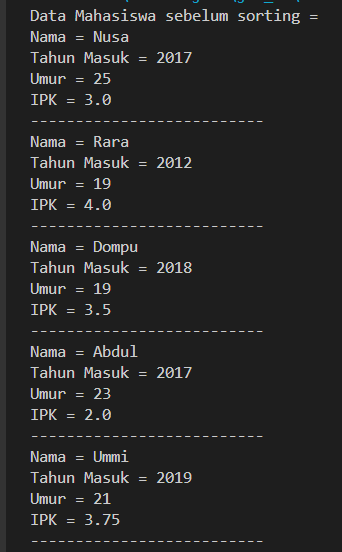
* **HASIL OUTPUT**

1. **Pertanyaan**

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

* Tampilan Kode Program **InsertionSort Descending**



* **Maka Hasil Output**

1. **Latihan Praktikum**

**Waktu : 90 Menit**

Sebuah platform travel yang menyediakan layanan pemesanan kebutuhan travelling sedang mengembangkan backend untuk sistem pemesanan/reservasi akomodasi (penginapan), salah satu fiturnya adalah menampilkan daftar penginapan yang tersedia berdasarkan pilihan filter yang diinginkan user. Daftar penginapan ini harus dapat disorting berdasarkan

* + - 1. Harga dimulai dari harga termurah ke harga tertinggi.
      2. Rating bintang penginapan dari bintang tertinggi (5) ke terendah (1)

Buatlah proses sorting data untuk kedua filter tersebut dengan menggunakan algoritma **bubble sort** dan **selection sort**.

