**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

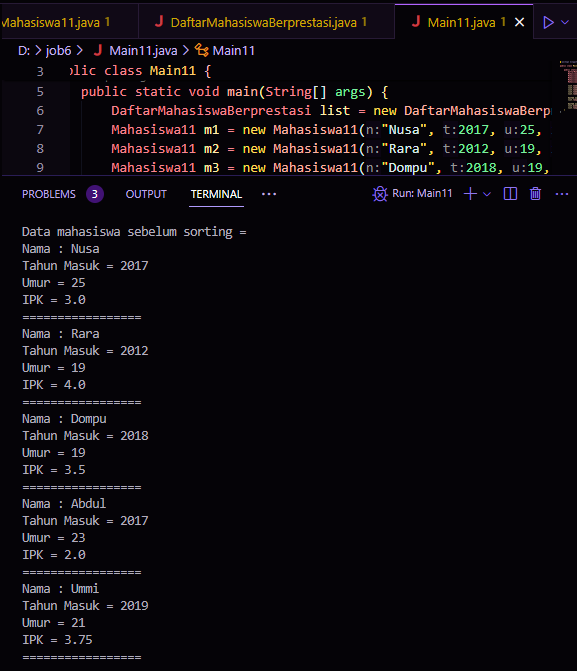
A picture containing text, receipt

Description automatically generated

A picture containing text, receipt

Description automatically generated

**Output :**





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

**Proses Bubble Sort ada dalam metode bubbleSort() di kelas DaftarMahasiswaBerprestasi. Metode ini digunakan untuk mengurutkan array listMhs, yang berisi objek-objek Mahasiswa11, berdasarkan nilai IPK secara menurun (descending).**Di dalam method bubbleSort(), terdapat baris program seperti di bawah ini:

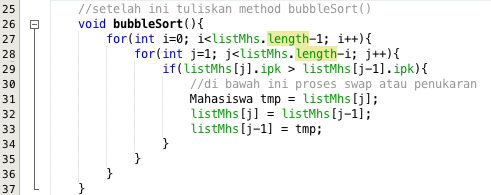
Text

Description automatically generated

Untuk apakah proses tersebut?

**Proses ini merupakan bagian dari algoritma Bubble Sort yang digunakan untuk membandingkan nilai IPK dari dua elemen berdekatan dalam array `listtMhs`. Jika nilai IPK elemen ke-j lebih kecil dari nilai IPK elemen ke-(j+1), maka posisi kedua elemen tersebut akan ditukar. Dengan begitu, elemen dengan nilai IPK tertinggi akan "menggelembung" ke posisi yang lebih awal dalam array setiap kali perulangan dilakukan, sehingga pada akhirnya array akan diurutkan secara menurun berdasarkan nilai IPK.**

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



* 1. Apakah perbedaan antara kegunaan perulangan i dan perulangan j?

**• Perulangan `i` digunakan untuk mengontrol jumlah iterasi secara keseluruhan. Pada setiap iterasi `i`, elemen terbesar dari sisa array akan "menggelembung" ke posisi yang benar.**

**• Perulangan `j` digunakan untuk membandingkan dan menukar elemen-elemen berdekatan dalam array, dimulai dari indeks 1 hingga `listtMhs.length - i`. Hal ini dilakukan untuk mengurutkan elemen-elemen dari yang terbesar ke yang terkecil. Dengan menggunakan kedua perulangan ini bersama-sama, algoritma Bubble Sort dapat mengurutkan array.**

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

**Kondisi perulangan `i < listMhs.length - 1` digunakan karena algoritma Bubble Sort membutuhkan `length - 1` iterasi untuk mengurutkan array dengan benar. Pada setiap iterasi, elemen terbesar akan "menggelembung" ke posisi yang benar di akhir array. Setelah `length - 1` iterasi, semua elemen akan berada pada posisi yang benar dari yang terkecil hingga yang terbesar, sehingga iterasi tambahan tidak diperlukan.**

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

**Syarat perulangan `j < listMhs.length - i` digunakan karena setiap iterasi `i` mengurangi jumlah elemen yang perlu dibandingkan di iterasi `j`. Pada awalnya, semua elemen dibandingkan, tetapi setelah beberapa iterasi `i`, elemen terbesar sudah berada pada posisi yang benar di ujung kanan array, sehingga tidak perlu lagi dibandingkan. Dengan membatasi perulangan `j`, kita hanya membandingkan elemen yang belum terurut.**

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

**Dalam kasus dengan 50 data di dalam `listMhs`, perulangan `i` akan berlangsung sebanyak 49 kali. Hal ini karena algoritma Bubble Sort memerlukan `n - 1` iterasi untuk mengurutkan array dengan `n` elemen. Tahap Bubble Sort yang ditempuh adalah sebanyak 49 tahap, di mana pada setiap tahap, satu elemen terbesar dari sisa array akan "menggelembung" ke posisi yang benar. Setelah 49 tahap, semua elemen akan berada pada posisi yang benar dari terkecil hingga terbesar.**

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

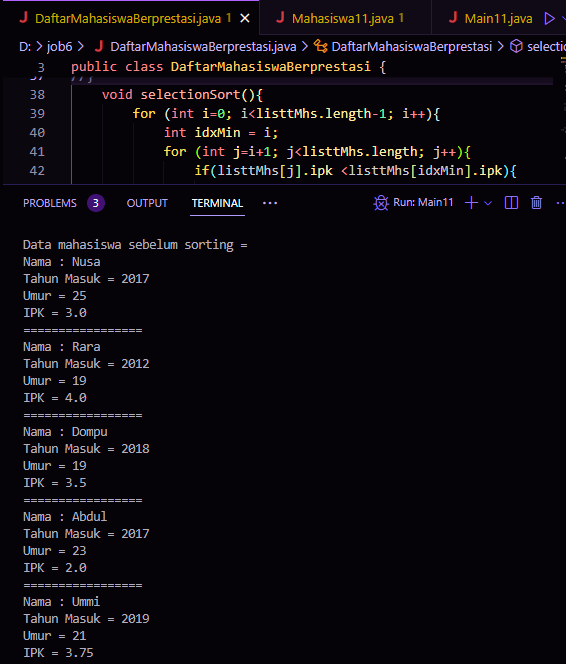
A picture containing text, receipt

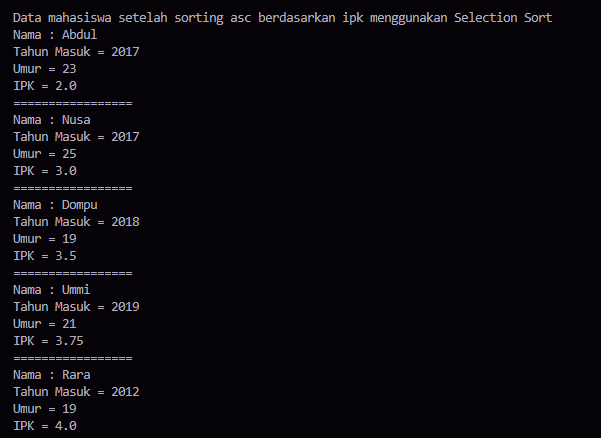
Description automatically generated

A picture containing text, receipt

Description automatically generated

**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 ini merupakan bagian dari algoritma Selection Sort yang digunakan untuk mencari elemen terkecil di antara elemen-elemen yang belum terurut dalam array berdasarkan nilai IPK. Pada setiap iterasi, jika nilai IPK dari elemen ke-j lebih kecil dari nilai IPK elemen terkecil yang telah ditentukan sebelumnya, maka indeks elemen terkecil diperbarui menjadi j. Setelah iterasi selesai, elemen terkecil ditukar dengan elemen pada indeks i, sehingga elemen terkecil akan berada di posisi yang benar dalam array setelah iterasi ke-i selesai.Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort**

**5.4 Waktu : 30 menit**

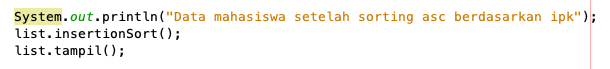
Yang terakhir akan diimplementasikan Teknik sorting menggunakan Insertion Sort, dengan mengurutkan IPK mahasiswa secara ascending.

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?
2. **Verifikasi Hasil Percobaan**

Pastikan output yang ditampilkan sudah benar seperti di bawah ini

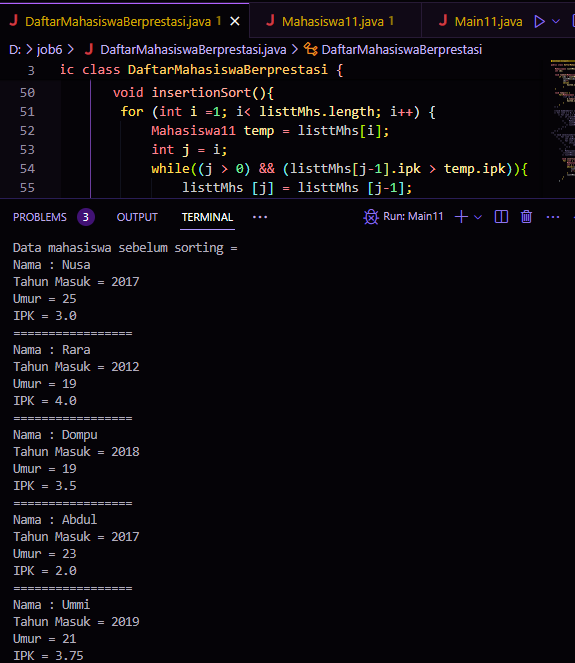
A picture containing text, receipt

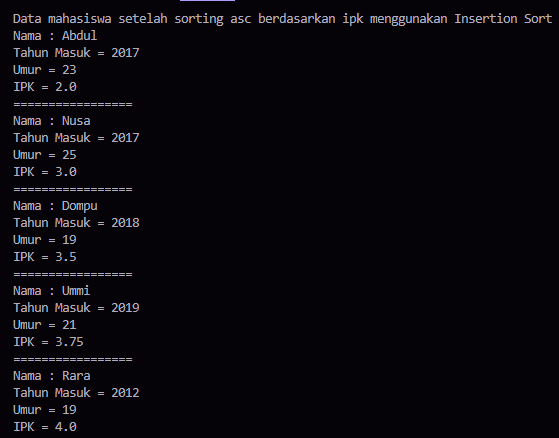
Description automatically generated

A picture containing text, receipt

Description automatically generated

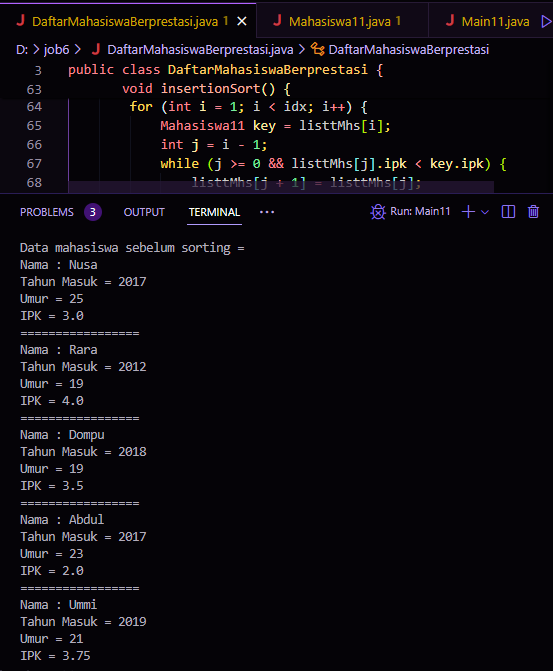
**Output :**

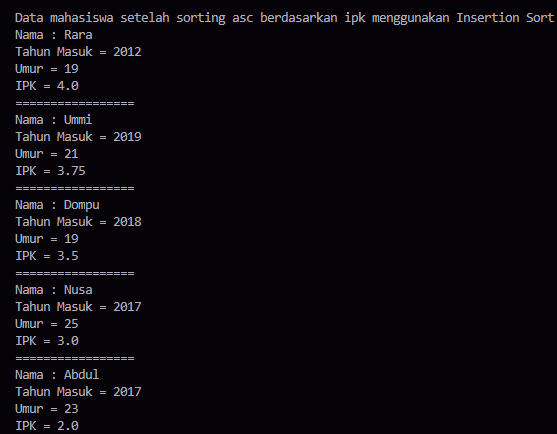




1. **Pertanyaan**

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



****

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**.

