# JOBSHEET - 5

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

**Nama: Rizqi Bagus Andrean**

**Kelas: TI-1D**

**Absen: 25**

# 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

# 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 |

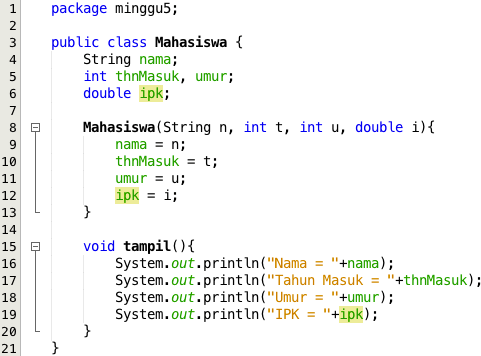
Berdasarkan class diagram di atas, kita akan membuat sebuah class Mahasiswa yang berfunsi untuk membuat objek mahasiswa yang akan dimasukan ke dalam sebuah array. Terdapat sebuah konstruktor berparameter dan juga fungsi tampil() untuk menampilkan semua attribute yang ada.

|  |
| --- |
| DaftarMahasiswaBerprestasi |
| listMhs: Mahasiswa[5]  idx: int |
| tambah(mhs: Mahasiswa): void tampil(): void  bubbleSort(): void |

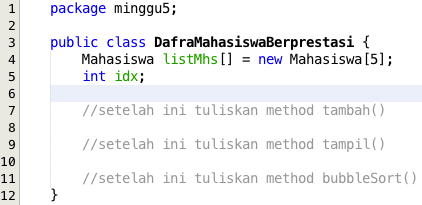
Selanjutnya class diagram di atas merupakan representasi dari sebuah class yang berfungsi untuk melakukan operasi-operasi dari objek array mahasiswa, misalkan untuk menambahkan objek mahasiswa, menampilkan semua data mahasiswa, dan juga untuk mengurutkan menggunakan Teknik bubble sort berdasarkan nilai IPK mahasiswa.

# Langkah-langkah Percobaan

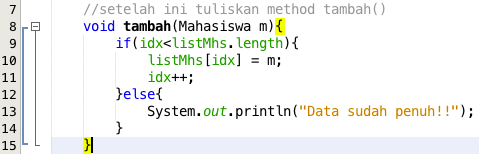
1. Buat project baru dengan nama “bubble-selection-insertion”, kemudian buat package dengan nama “jobsheet6”.
2. Buatlah sebuah class dengan nama Mahasiswa
3. Sesuaikan class Mahasiswa dengan melihat class diagram di atas dengan menambahkan attribute, konstruktor, dan fungsi atau method. Untuk lebih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



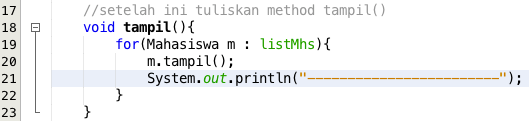
1. Buat class DaftarMahasiswaBerprestasi seperti di bawah ini!



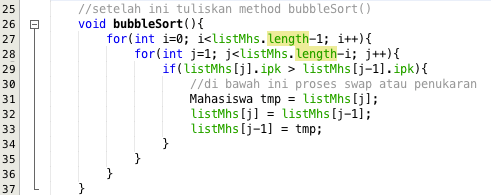
1. Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.



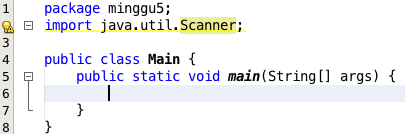
1. Tambahkan method tampil() di dalam class tersebut! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.



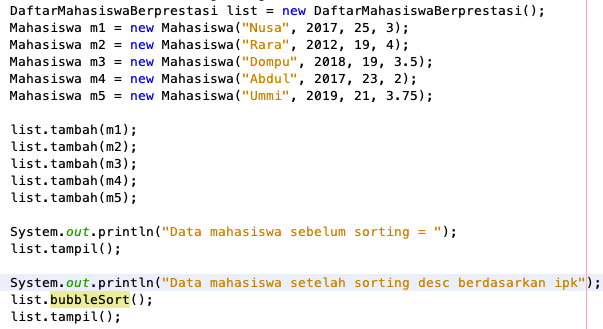
1. Tambahkan method bubbleSort() di dalam class tersebut!



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

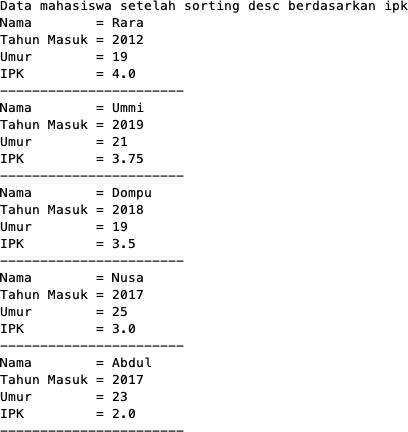
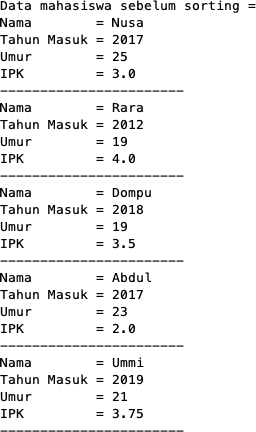


1. **Di dalam method main()**, buatlah sebuah objek DaftarMahasiswaBerprestasi dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek DaftarMahasiswaBerprestasi. Silakan dipanggil fungsi tampil() untuk melihat semua data yang telah dimasukan, urutkan data tersebut dengan memanggil fungsi bubbleSort() dan yang terakhir panggil fungsi tampil kembali.



# Verifikasi Hasil Percobaan

Cocokan hasilnya dengan yang terdapat pada tampilan di bawah ini

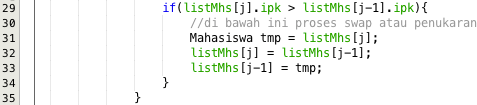


# Pertanyaan

1. Terdapat di method apakah proses bubble sort?

Method bubbleSort.

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



Untuk apakah proses tersebut?

Bagian kode ini menukar elemen dalam bubble sort. Membandingkan IPK elemen terdekat (listMMhs[j] dan listMMhs[3j-1]). Jika IPK elemen sekarang lebih besar, maka ditukar menggunakan variabel temporary temp. Ini memastikan elemen terurut sesuai IPK (naik dalam kasus ini).

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



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

Perulangan i dan j bekerja sama untuk mengurutkan elemen dalam listMMhs berdasarkan IPK. Perulangan i mengontrol berapa kali perbandingan dan pertukaran dilakukan, sedangkan perulangan j melakukan perbandingan dan pertukaran tersebut.

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

Syarat i < listMMhs.length - 1 memastikan perulangan i berhenti sebelum mencapai batas array dan elemen terakhir diurutkan dengan benar.

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

Syarat j < listMMhs.length - i memastikan perulangan j hanya membandingkan elemen yang belum terurut dan meningkatkan efisiensi bubble sort

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

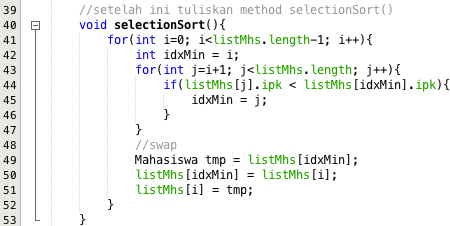
Perulangan i dalam bubble sort dengan 50 data akan berlangsung sebanyak **49** kali

# 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.

# Langkah-langkah Percobaan.

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



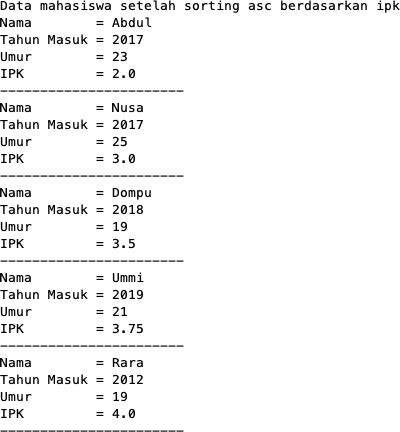
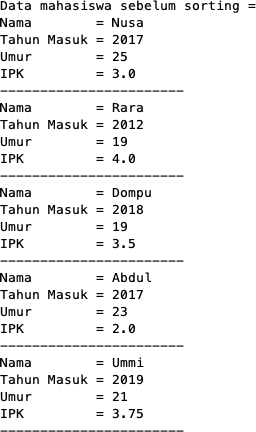
1. Setelah itu, buka kembali class Main, dan di dalam method main() tambahkan baris program untuk memanggil method selectionSort() tersebut!



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

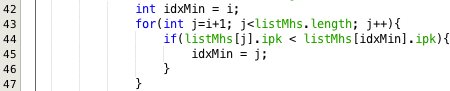
# Verifikasi Hasil Percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini



# Pertanyaan

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



Untuk apakah proses tersebut, jelaskan!

Bagian kode itu mencari indeks mahasiswa dengan IPK terkecil dalam sub-list yang belum terurut. Looping j iterasi di seluruh sub-list dan membandingkan IPK. Jika ditemukan IPK yang lebih kecil, indeks minimum (idxMin) diperbarui.

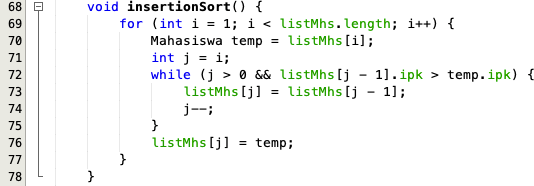
Bagian kode itu mencari indeks mahasiswa dengan IPK terkecil dalam sub-list yang belum terurut. Looping j iterasi di seluruh sub-list dan membandingkan IPK. Jika ditemukan IPK yang lebih kecil, indeks minimum (idxMin) diperbarui.

# Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort Waktu : 30 menit

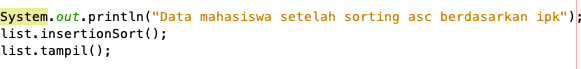
Yang terakhir akan diimplementasikan Teknik sorting menggunakan Insertion Sort, dengan mengurutkan IPK mahasiswa secara ascending.

# Langkah-langkah Percobaan

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



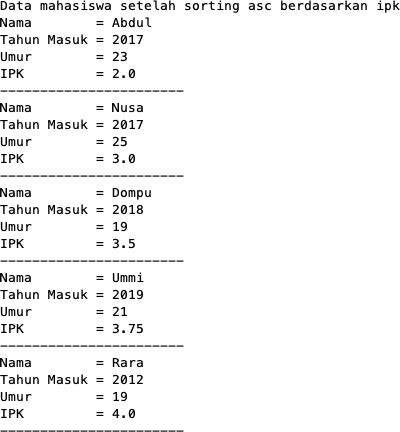
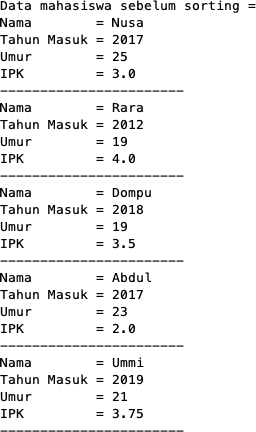
1. Setelah itu, buka kembali class Main, dan di dalam method main() tambahkan baris program untuk memanggil method insertionSort() tersebut!



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

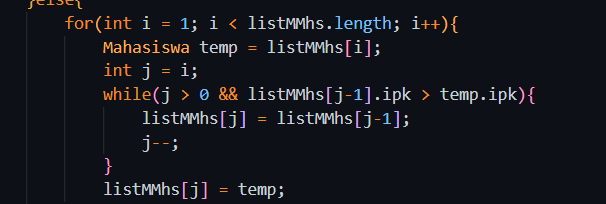
# Verifikasi Hasil Percobaan

Pastikan output yang ditampilkan sudah benar seperti di bawah ini



# Pertanyaan

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



# 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.

