**Nama : Rizqi Dewanto**

**Number : 17**

**Class : SIB - 1G**

**Github : https://github.com/Rizxqi**

**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**

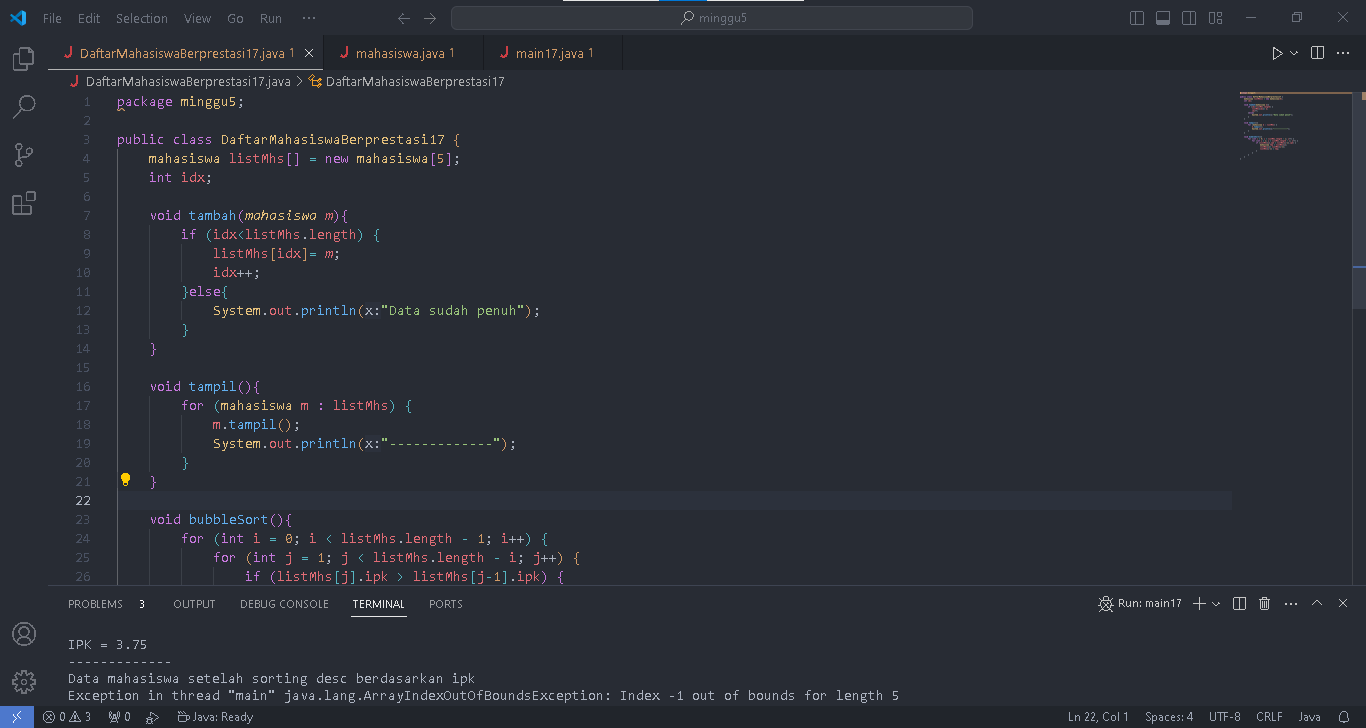
Cocokan hasilnya dengan yang terdapat pada tampilan di bawah ini

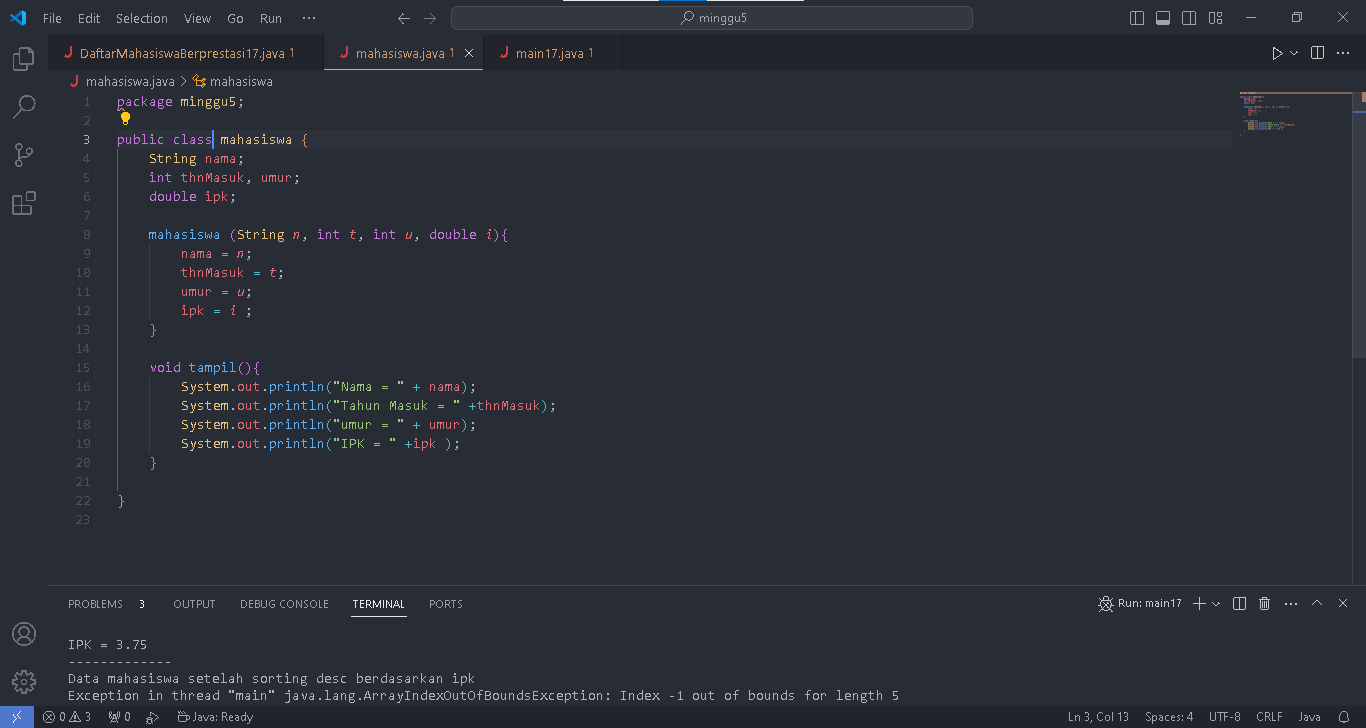
A picture containing text, receipt

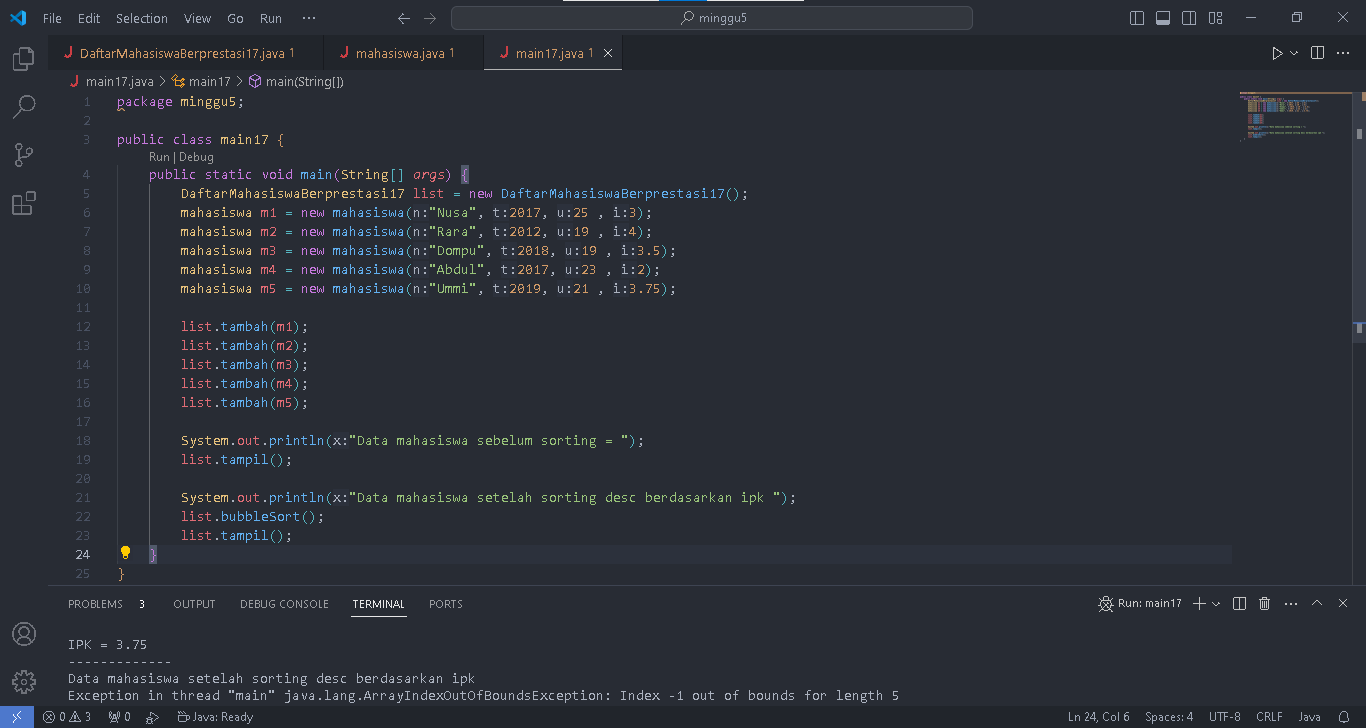
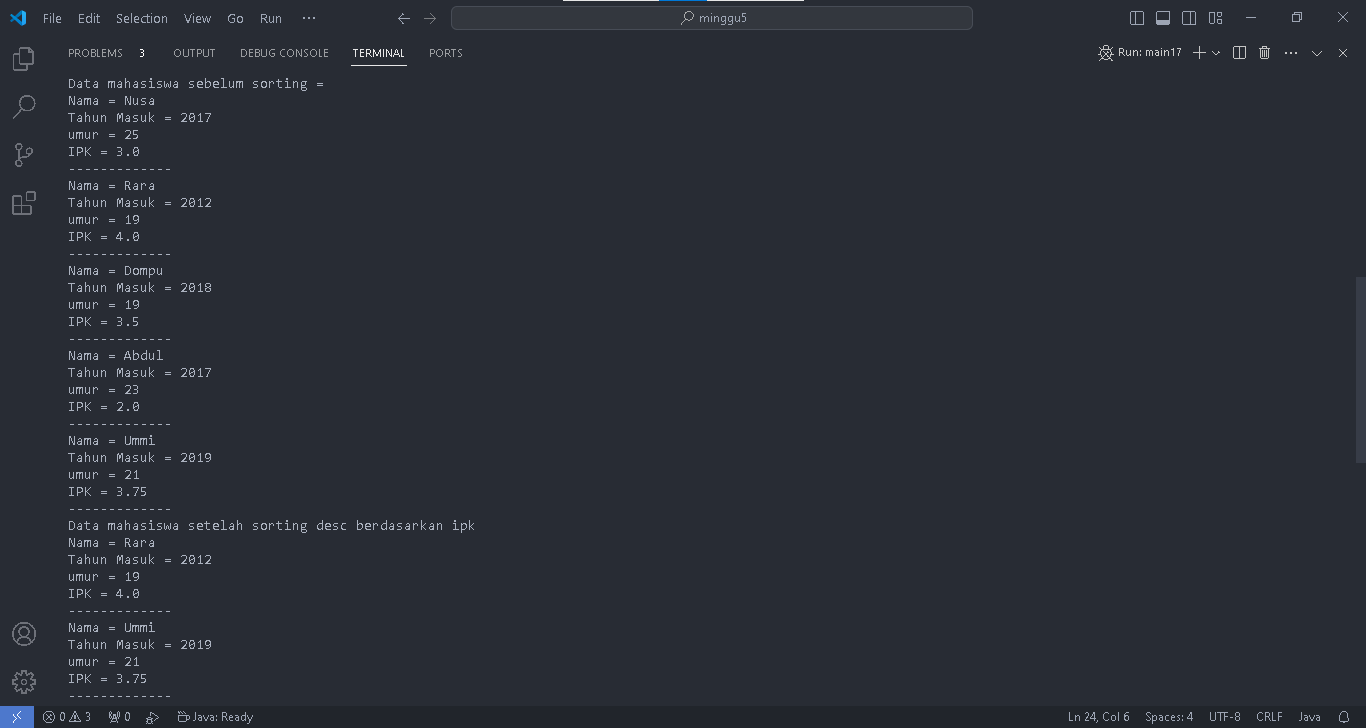
Description automatically generated

A picture containing text, receipt

Description automatically generated







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

**Proses bubble sort terdapat pada method `bubbleSort()` dalam kelas `DaftarMahasiswaBerprestasi17`. Metode ini menggunakan algoritma bubble sort untuk mengurutkan objek `mahasiswa` dalam array `listMhs` berdasarkan nilai `ipk` (indeks prestasi kumulatif) dari yang tertinggi ke yang terendah.**

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

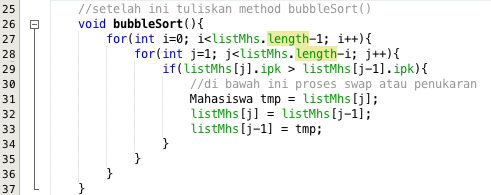
Text

Description automatically generated

Untuk apakah proses tersebut?

**Proses tersebut merupakan bagian dari algoritma bubble sort yang membandingkan nilai `ipk` antara dua elemen dalam array `listMhs`, kemudian menukar posisi jika nilai `ipk` pada indeks `j` lebih besar dari nilai `ipk` pada indeks `j-1`. Selama iterasi algoritma, nilai `ipk` tertinggi akan bergerak ke akhir array, menghasilkan array terurut dari nilai `ipk` tertinggi hingga terendah.**

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



* 1. Apakah perbedaan antara kegunaan perulangan i dan perulangan j?
  2. Mengapa syarat dari perulangan i adalah i<listMhs.length-1 ?
  3. Mengapa syarat dari perulangan j adalah j<listMhs.length-i ?
  4. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?

**a. Perulangan i mengontrol iterasi keseluruhan pengurutan, sementara perulangan j digunakan untuk membandingkan dan menukar elemen.**

**b. Perulangan i menggunakan syarat `i < listMhs.length - 1` karena proses bubble sort hanya memerlukan n - 1 iterasi.**

**c. Syarat perulangan j adalah `j < listMhs.length - i` karena setiap iterasi i memindahkan elemen terbesar ke posisi yang benar, membatasi perulangan j hingga elemen yang belum diurutkan saja.**

**d. Dalam kasus 50 data, perulangan i berlangsung 49 kali dengan 49 tahap bubble sort untuk memindahkan elemen terbesar ke posisi yang benar.**

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!

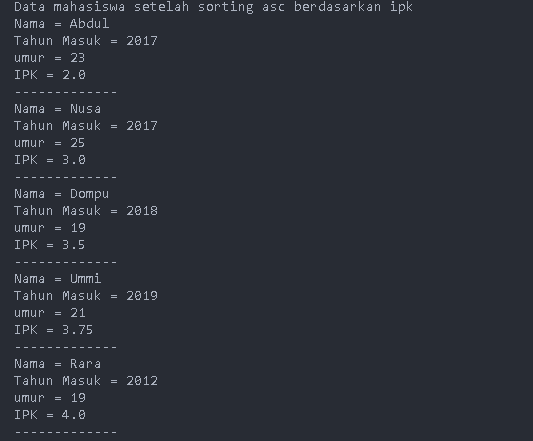


1. Coba jalankan kembali class Main, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?
2. **Verifikasi Hasil Percobaan**

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

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 bagian dari algoritma selection sort yang mencari indeks dari elemen dengan nilai ipk terkecil dalam bagian array yang belum diurutkan.**

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

**Waktu : 30 menit**

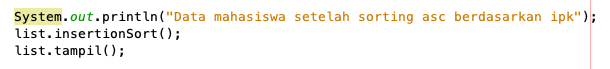
Yang terakhir akan diimplementasikan Teknik sorting menggunakan Insertion Sort, dengan mengurutkan IPK mahasiswa secara ascending.

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

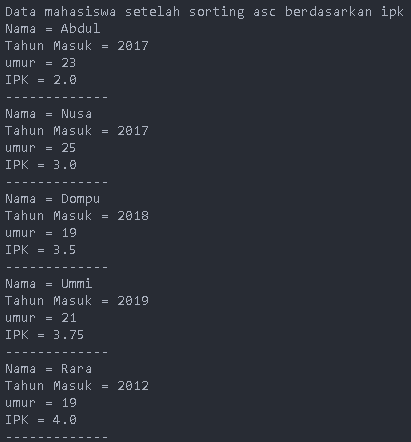
Description automatically generated

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?
2. **Verifikasi Hasil Percobaan**

Pastikan output yang ditampilkan sudah benar seperti di bawah ini

A picture containing text, receipt

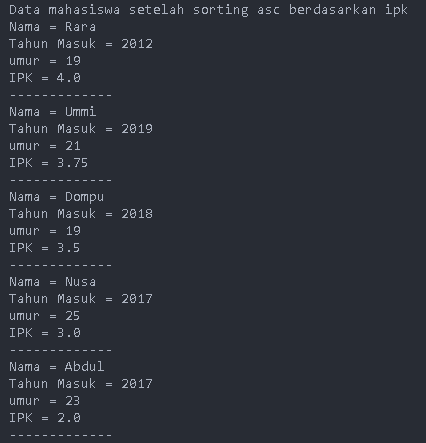
Description automatically generated

A picture containing text, receipt

Description automatically generated

1. **Pertanyaan**

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

****

while (j > 0 && listMhs[ j - 1 ].ipk < temp.ipk) {

                listMhs[j] = listMhs[j-1];

                j--;

           }

**Ubah kondisi** while (j > 0 && listMhs[ j - 1 ].ipk > temp.ipk) {

Menjadi while (j > 0 && listMhs[ j - 1 ].ipk < temp.ipk) {

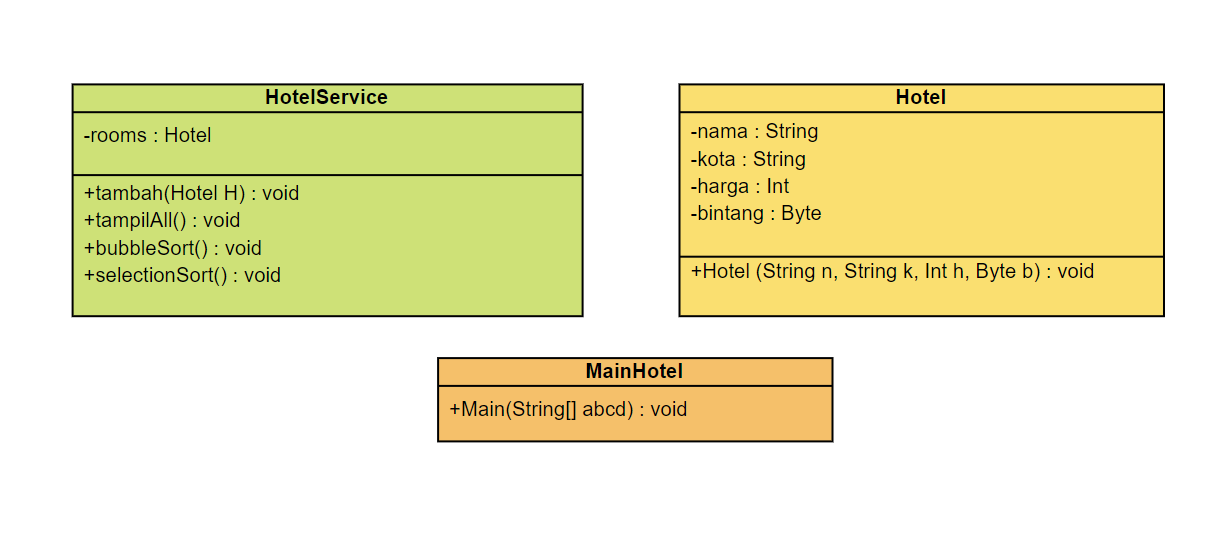
1. **Latihan Praktikum**

**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**.



Code :

package tugasPrak17;

public class hotel17 {

    String nama;

    String kota;

    int harga;

    byte bintang;

    hotel17(String *n*, String *k*, int *h*, byte *b*) {

        nama = *n*;

        kota = *k*;

        harga = *h*;

        bintang = *b*;

    }

}

package tugasPrak17;

public class hotelService {

    hotel17[] rooms;

    int count;

    public hotelService(int *capacity*) {

        rooms = new hotel17[*capacity*];

        count = 0;

    }

    public void tambah(*hotel17* *H*) {

        if (count < rooms.length) {

            rooms[count] = *H*;

            count++;

        } else {

            System.out.println("Daftar hotel sudah penuh");

        }

    }

    public void hapusAll() {

        rooms = new hotel17[rooms.length];

        count = 0;

    }

    public void bubbleSortByPrice() {

        for (int i = 0; i < count - 1; i++) {

            for (int j = 0; j < count - 1 - i; j++) {

                if (rooms[j].harga > rooms[j + 1].harga) {

                    hotel17 temp = rooms[j];

                    rooms[j] = rooms[j + 1];

                    rooms[j + 1] = temp;

                }

            }

        }

    }

    public void bubbleSortByRating() {

        for (int i = 0; i < count - 1; i++) {

            for (int j = 0; j < count - 1 - i; j++) {

                if (rooms[j].bintang < rooms[j + 1].bintang) {

                    hotel17 temp = rooms[j];

                    rooms[j] = rooms[j + 1];

                    rooms[j + 1] = temp;

                }

            }

        }

    }

    public hotel17[] getRooms() {

        return rooms;

    }

}

package tugasPrak17;

public class mainHotel {

    public static void main(String[] *args*) {

        hotelService hotelService = new hotelService(10);

*// Menambahkan beberapa hotel ke dalam daftar*

  hotelService.tambah(new hotel17("Hotel A", "Kota A", 150, (byte) 4));

  hotelService.tambah(new hotel17("Hotel B", "Kota B", 200, (byte) 3));

  hotelService.tambah(new hotel17("Hotel C", "Kota C", 100, (byte) 5));

*// Menampilkan daftar hotel sebelum diurutkan*

    System.out.println("Daftar Hotel Sebelum Diurutkan:");

    hotel17[] daftarSebelumUrut = hotelService.getRooms();

    for (hotel17 hotel : daftarSebelumUrut) {

        if (hotel != null) {

     System.out.println(hotel.nama + " - $" + hotel.harga + " - " + hotel.bintang + " bintang");

        }

    }

*// Mengurutkan daftar hotel berdasarkan harga*

    hotelService.bubbleSortByPrice();

*// Menampilkan daftar hotel setelah diurutkan*

    System.out.println("\nDaftar Hotel Setelah Diurutkan Berdasarkan Harga:");

    hotel17[] daftarSetelahUrut = hotelService.getRooms();

    for (hotel17 hotel : daftarSetelahUrut) {

        if (hotel != null) {

            System.out.println(hotel.nama + " - $" + hotel.harga + " - " + hotel.bintang + " bintang");

        }

    }

}

}

OUTPUT :

