

Nama : Siti Nikmatus Sholihah

NIM : 244107020014

Kelas : TI 1B

Percobaan 1 : Mengimplementasikan Sorting menggunakan Object

1. Sorting – Bubble Sort

- Buat folder baru bernama jobsheet6 didalam repository Praktikum ASD.
- Buat class Sorting22.java, kemudian tambahkan atribut.

```
jobsheet6 > J Sorting22.java
1 public class Sorting22 {
2
3     int [] data;
4     int jumData;
5 }
```

- Buatlah konstruktor dengan parameter Data[] dan jumData.

```
jobsheet6 > J Sorting22.java > S Sorting22 > S Sorting22(int[], int)
1 public class Sorting22 {
2
3     int [] data;
4     int jumData;
5
6     Sorting22 (int Data[], int jmlDat){
7         jumData = jmlDat;
8         data = new int[jmlDat];
9         for (int i = 0; i < jumData; i++) {
10             data[i] = Data[i];
11         }
12     }
13 }
```

- Buatlah method bubbleSort bertipe void dan deklarasi isinya menggunakan algoirtma Bubble Sort.

```
jobsheet6 > J Sorting22.java > S Sorting22 > S bubbleSort()
1 public class Sorting22 {
2
3     int [] data;
4     int jumData;
5
6     Sorting22 (int Data[], int jmlDat){
7         jumData = jmlDat;
8         data = new int[jmlDat];
9         for (int i = 0; i < jumData; i++) {
10             data[i] = Data[i];
11         }
12     }
13
14     void bubbleSort (){
15         int temp = 0;
16         for (int i = 0; i < jumData-1; i++) {
17             for (int j = 1; j < jumData-i; j++) {
18                 if (data[j-1]>data[j]) {
19                     temp=data[j];
20                     data[j]=data[j-1];
21                     data[j-1]=temp;
22                 }
23             }
24         }
25     }
26 }
```

- e. Buatlah method tampil bertipe void dan deklarasikan isi method tersebut.

```
jobsheet6 > J Sorting22.java > Sorting22 > tampil()
1 public class Sorting22 {
2
3     int [] data;
4     int jumData;
5
6     Sorting22 (int Data[], int jmlDat){
7         jumData = jmlDat;
8         data = new int[jmlDat];
9         for (int i = 0; i < jumData; i++) {
10             data[i] = Data[i];
11         }
12     }
13
14     void bubbleSort (){
15         int temp = 0;
16         for (int i = 0; i < jumData-1; i++) {
17             for (int j = 1; j < jumData-i; j++) {
18                 if (data[j-1]>data[j]) {
19                     temp=data[j];
20                     data[j]=data[j-1];
21                     data[j-1]=temp;
22                 }
23             }
24         }
25     }
26
27     void tampil(){
28         for (int i = 0; i < jumData; i++) {
29             System.out.print(data[i]+" ");
30         }
31         System.out.println();
32     }
33 }
```

- f. Buat class SortingMain kemudian deklarasikan array dengan nama a[] kemudian isi array tersebut.

```
jobsheet6 > J SortingMain22.java > SortingMain22 > main(String[])
1 public class SortingMain22 {
2     Run | Debug
3     public static void main(String[] args) {
4
5         int a[]={20, 10, 2, 7, 12};
6     }
7 }
```

- g. Buatlah objek baru dengan nama dataurut1 yang merupakan instansiasi dari class Sorting, kemudian isi parameterinya.

```
jobsheet6 > J SortingMain22.java > SortingMain22 > main(String[])
1 public class SortingMain22 {
2     Run | Debug
3     public static void main(String[] args) {
4
5         int a[]={20, 10, 2, 7, 12};
6         Sorting22 dataurut1 = new Sorting22(a, a.length);
7     }
8 }
```

- h. Lakukan pemanggilan method bubbleSort dan tampil.

```
jobsheet6 > J SortingMain22.java > SortingMain22 > main(String[])
1 public class SortingMain22 {
2     Run | Debug
3     public static void main(String[] args) {
4
5         int a[]={20, 10, 2, 7, 12};
6
7         Sorting22 dataurut1 = new Sorting22(a, a.length);
8         System.out.println(x:"Data awal 1");
9         dataurut1.tampil();
10        dataurut1.bubbleSort();
11        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT(ASC)");
12        dataurut1.tampil();
13    }
```

- i. Verifikasi hasil percobaan.

```
D:\Praktikum ASD>cd "d:\Praktikum ASD\jobsheet6" && javac SortingMain22.java && java SortingMain22
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT(ASC)
2 7 10 12 20
```

2. Sorting – Selection Sort

- a. Pada class Sorting22 yang sudah dibuat di praktikum sebelumnya tambahkan method SelectionSort yang mengimplementasikan pengurutan menggunakan algoritma selection sort.

```
jobsheet6 > J Sorting22.java > Sorting22 > bubbleSort()
1 public class Sorting22 {
34 void SelectionSort(){
35     for (int i = 0; i < jumData-1; i++) {
36         int min = i;
37         for (int j = i+1; j < jumData; j++) {
38             if (data[j]<data[min]) {
39                 min =j;
40             }
41         }
42         int temp = data[i];
43         data[i]=data[min];
44         data[min]=temp;
45     }
46 }
47 }
```

- b. Deklarasikan array dengan nama b[] pada kelas SortingMain kemudian isi array tersebut.

```
jobsheet6 > J SortingMain22.java > ...
1 public class SortingMain22 {
2     Run | Debug
3     public static void main(String[] args) {
4
5         int a[]={20, 10, 2, 7, 12};
6         int b[]={30, 20, 2, 8, 14};
7
8         Sorting22 dataurut1 = new Sorting22(a, a.length);
9         System.out.println(x:"Data awal 1");
10        dataurut1.tampil();
11        dataurut1.bubbleSort();
12        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT(ASC)");
13        dataurut1.tampil();
14    }
```

- c. Buatlah objek baru dengan nama dataurut2 yang merupakan instansiasi dari class Sorting, kemudian isi parameteranya.

```

jobsheet6 > J SortingMain22.java > SortingMain22 > main(String[])
1  public class SortingMain22 {
    Run | Debug
2      public static void main(String[] args) {
3
4          int a[]={20, 10, 2, 7, 12};
5          int b[]={30, 20, 2, 8, 14};
6
7          Sorting22 dataurut1 = new Sorting22(a, a.length);
8          System.out.println(x:"Data awal 1");
9          dataurut1.tampil();
10         dataurut1.bubbleSort();
11         System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT(ASC)");
12         dataurut1.tampil();
13
14         Sorting22 dataurut2 = new Sorting22(b, b.length);|
15
16     }
17 }

```

- d. Lakukan pemanggilan method SelectionSort dan tampil.

```

jobsheet6 > J SortingMain22.java > SortingMain22 > main(String[])
1  public class SortingMain22 {
    Run | Debug
2      public static void main(String[] args) {
3
4          int a[]={20, 10, 2, 7, 12};
5          int b[]={30, 20, 2, 8, 14};
6
7          Sorting22 dataurut1 = new Sorting22(a, a.length);
8          System.out.println(x:"Data awal 1");
9          dataurut1.tampil();
10         dataurut1.bubbleSort();
11         System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT(ASC)");
12         dataurut1.tampil();
13
14         Sorting22 dataurut2 = new Sorting22(b, b.length);
15         System.out.println(x:"Data awal 2");
16         dataurut2.tampil();
17         dataurut2.bubbleSort();
18         System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT(ASC)");
19         dataurut2.tampil();|
20
21     }

```

- e. Jalankan program dan amati hasilnya!

```

d:\Praktikum-ASD\jobsheet6>cd "d:\Praktikum-ASD\
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT(ASC)
2 7 10 12 20
Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT(ASC)
2 8 14 20 30

```

3. Sorting – Insertion Sort

- a. Pada class Sorting22 yang sudah dibuat di praktikum sebelumnya tambahkan method insertionSort yang mengimplementasikan pengurutan menggunakan algoritma insertion sort.

```
jobsheet6 > J Sorting22.java > Sorting22 > SelectionSort()
1 public class Sorting22 {
48     void InsertionSort(){
49         for (int i = 1; i < data.length-1; i++) {
50             int temp = data[i];
51             int j=i-1;
52             while (j>=0 && data[j]>temp) {
53                 data[j+1] = data[j];
54                 j--;
55             }
56             data[j+1]=temp;
57         }
58     }
59 }
```

- b. Deklarasikan array dengan nama c[] pada kelas SortingMain kemudian isi array tersebut.

```
jobsheet6 > J SortingMain22.java > SortingMain22 > main(String[])
1 public class SortingMain22 {
2     Run | Debug
3     public static void main(String[] args) {
4         int a[]={20, 10, 2, 7, 12};
5         int b[]={30, 20, 2, 8, 14};
6         int c[]={40, 10, 4, 9, 3};
7
8         Sorting22 dataurut1 = new Sorting22(a, a.length);
9         System.out.println(x:"Data awal 1");
10        dataurut1.tampil();
11        dataurut1.bubbleSort();
12        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT(ASC)");
13        dataurut1.tampil();
14
15        Sorting22 dataurut2 = new Sorting22(b, b.length);
16        System.out.println(x:"Data awal 2");
17        dataurut2.tampil();
18        dataurut2.bubbleSort();
19        System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT(ASC)");
20        dataurut2.tampil();
21    }
22 }
```

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

```

jobsheet6 > J SortingMain22.java > SortingMain22 > main(String[])
1 public class SortingMain22 {
2     Run | Debug
3     public static void main(String[] args) {
4
5         int a[]={20, 10, 2, 7, 12};
6         int b[]={30, 20, 2, 8, 14};
7         int c[]={40, 10, 4, 9, 3};
8
9         Sorting22 dataurut1 = new Sorting22(a, a.length);
10        System.out.println(x:"Data awal 1");
11        dataurut1.tampil();
12        dataurut1.bubbleSort();
13        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT(ASC)");
14        dataurut1.tampil();
15
16        Sorting22 dataurut2 = new Sorting22(b, b.length);
17        System.out.println(x:"Data awal 2");
18        dataurut2.tampil();
19        dataurut2.bubbleSort();
20        System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT(ASC)");
21        dataurut2.tampil();
22
23        Sorting22 dataurut3 = new Sorting22(c, c.length);
24    }

```

- d. Lakukan pemanggilan method insertionSort dan tampil.

```

jobsheet6 > J SortingMain22.java > SortingMain22 > main(String[])
1 public class SortingMain22 {
2     Run | Debug
3     public static void main(String[] args) {
4
5         int a[]={20, 10, 2, 7, 12};
6         int b[]={30, 20, 2, 8, 14};
7         int c[]={40, 10, 4, 9, 3};
8
9         Sorting22 dataurut1 = new Sorting22(a, a.length);
10        System.out.println(x:"Data awal 1");
11        dataurut1.tampil();
12        dataurut1.bubbleSort();
13        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT(ASC)");
14        dataurut1.tampil();
15
16        Sorting22 dataurut2 = new Sorting22(b, b.length);
17        System.out.println(x:"Data awal 2");
18        dataurut2.tampil();
19        dataurut2.bubbleSort();
20        System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT(ASC)");
21        dataurut2.tampil();
22
23        Sorting22 dataurut3 = new Sorting22(c, c.length);
24        System.out.println(x:"Data awal 3");
25        dataurut3.tampil();
26        dataurut3.bubbleSort();
27        System.out.println(x:"Data sudah diurutkan dengan INSERTION SORT(ASC)");
28        dataurut3.tampil();
29    }

```

- e. Jalankan program dan amati hasilnya!

```
D:\Praktikum-ASD>cd "d:\Praktikum-ASD\jobsheet6\"
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT(ASC)
2 7 10 12 20
Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT(ASC)
2 8 14 20 30
Data awal 3
40 10 4 9 3
Data sudah diurutkan dengan INSERTION SORT(ASC)
3 4 9 10 40
```

Pertanyaan

1. Jelaskan fungsi kode proram berikut

```
if (data[j-1]>data[j]) {
    temp=data[j];
    data[j]=data[j-1];
    data[j-1]=temp;
}
```

Jawab: Kode tersebut merupakan bagian dari algoritma Bubble Sort yang digunakan untuk mengurutkan data dalam array (biasanya secara ascending)

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

Jawab:

```
int idxMin=i;
for (int j = i+1; j < listMhs.length; j++) {
    if (listMhs[j].ipk<listMhs[idxMin].ipk) {
        idxMin = j;
    }
}
```

3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan

```
while (j>=0 && data[j]>temp)
```

Jawab:

1. $j \geq 0$ → Memastikan perulangan hanya berjalan selama indeks j masih dalam batas array (tidak negatif).
2. $data[j] > temp$ → Memeriksa apakah elemen terurut ($data[j]$) lebih besar dari elemen yang akan disisipkan ($temp$).
3. **Tujuan:** Menggeser elemen-elemen yang lebih besar ke kanan sampai menemukan posisi yang tepat untuk $temp$.
4. Pada Insertion sort, apakah tujuan dari perintah

```
data[j+1] = data[j];
```

Jawab: Perintah ini digunakan untuk **menggeser elemen ke kanan** satu posisi guna membuka ruang (slot kosong) untuk penyisipan elemen yang sedang diproses (temp).

Percobaan 2 : Sorting Menggunakan Array of Object

1. Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort)

- Buatlah class dengan nama Mahasiswa22.java.
- Tulis struktur dasar class java dan deklarasikan nama, nim, kelas menggunakan tipe data String dan ipk dengan tipe data double.

```
jobsheet6 > J Mahasiswa22.java > Mahasiswa22
1 public class Mahasiswa22 {
2     String nim, nama, kelas;
3     double ipk;
4 }
5
```

- Tambahkan konstruktor default.

```
jobsheet6 > J Mahasiswa22.java > ...
1 public class Mahasiswa22 {
2     String nim, nama, kelas;
3     double ipk;
4
5     Mahasiswa22(){
6
7     }
8 }
```

- Tambahkan konstruktor berparameter nama, nim, kelas, dan ipk.

```
jobsheet6 > J Mahasiswa22.java > ...
1 public class Mahasiswa22 {
2     String nim, nama, kelas;
3     double ipk;
4
5     Mahasiswa22(){
6
7     }
8
9     Mahasiswa22(String nm, String name, String kls, double ip){
10         nim = nm;
11         nama = name;
12         kelas = kls;
13         ipk = ip;
14     }
15 }
```

- Tambahkan method untuk menampilkan informasi.

```
jobsheet6 > J Mahasiswa22.java > Mahasiswa22 > tampilInformasi()
1 public class Mahasiswa22 {
2     String nim, nama, kelas;
3     double ipk;
4
5     Mahasiswa22(){
6
7     }
8
9     Mahasiswa22(String nm, String name, String kls, double ip){
10         nim = nm;
11         nama = name;
12         kelas = kls;
13         ipk = ip;
14     }
15
16     void tampilInformasi(){
17         System.out.println("Nama: " + nama);
18         System.out.println("NIM: " + nim);
19         System.out.println("Kelas: " + kelas);
20         System.out.println("IPK: " + ipk);
21     }
22 }
```

- f. Buat class MahasiswaBerprestasi22.java dan deklarasikan array Mahasiswa[].

```
jobsheet6 > J MahasiswaBerprestasi22.java > MahasiswaBerprestasi22 > idx
1 public class MahasiswaBerprestasi22 {
2     Mahasiswa22[] listMhs = new Mahasiswa22[5];
3     int idx;
4 }
```

- g. Tambahkan method tambah() didalam class tersebut yang fungsinya untuk menambahkan objek dari class Mahasiswa22 ke dalam atribut listMahasiswa.

```
jobsheet6 > J MahasiswaBerprestasi22.java > ...
1 public class MahasiswaBerprestasi22 {
2     Mahasiswa22[] listMhs = new Mahasiswa22[5];
3     int idx;
4
5     void tambah (Mahasiswa22 m){
6         if (idx<listMhs.length) {
7             listMhs[idx]=m;
8             idx++;
9         } else {
10             System.out.println(x:"Data Sudah Penuh");
11         }
12     }
13 }
```

- h. Tambahkan method tampil() di class tersebut yang digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada dalam class tersebut.

```
jobsheet6 > J MahasiswaBerprestasi22.java > ...
1 public class MahasiswaBerprestasi22 {
2     Mahasiswa22[] listMhs = new Mahasiswa22[5];
3     int idx;
4
5     void tambah (Mahasiswa22 m){
6         if (idx<listMhs.length) {
7             listMhs[idx]=m;
8             idx++;
9         } else {
10             System.out.println(x:"Data Sudah Penuh");
11         }
12     }
13
14     void tampil(){
15         for (Mahasiswa22 m : listMhs) {
16             m.tampilInformasi();
17             System.out.println(x:"-----");
18         }
19     }
20 }
```

- i. Tambahkan method BubbleSort() di dalam class tersebut.

```
jobsheet6 > J MahasiswaBerprestasi22.java > MahasiswaBerprestasi22 > bubbleSort()
1 public class MahasiswaBerprestasi22 {
2     Mahasiswa22[] listMhs = new Mahasiswa22[5];
3     int idx;
4
5     void tambah (Mahasiswa22 m){
6         if (idx<listMhs.length) {
7             listMhs[idx]=m;
8             idx++;
9         } else {
10             System.out.println(x:"Data Sudah Penuh");
11         }
12     }
13
14     void tampil(){
15         for (Mahasiswa22 m : listMhs) {
16             m.tampilInformasi();
17             System.out.println(x:"-----");
18         }
19     }
20
21     void bubbleSort(){
22         for (int i = 0; i < listMhs.length-1; i++) {
23             for (int j = 1; j < listMhs.length-i; j++) {
24                 if (listMhs[j].ipk>listMhs[j-1].ipk) {
25                     Mahasiswa22 tmp = listMhs[j];
26                     listMhs[j]=listMhs[j-1];
27                     listMhs[j-1]=tmp;
28                 }
29             }
30         }
31     }
32 }
```

- j. Buat class MahasiswaDemo22.java, lalu buatlah sebuah objek MahasiswaBerprestasi22 dan buatlah 5 objek mahasiswa22 kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek MahasiswaBerprestasi22. 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.

```
jobsheet6 > J MahasiswaDemo22.java > MahasiswaDemo22 > main(String[])
1 public class MahasiswaDemo22 {
2     public static void main(String[] args) {
3
4         MahasiswaBerprestasi22 list = new MahasiswaBerprestasi22();
5         Mahasiswa22 m1 = new Mahasiswa22(nm:"123", name:"Haydar", kls:"2A", ip:3.8);
6         Mahasiswa22 m2 = new Mahasiswa22(nm:"124", name:"Bintang", kls:"2A", ip:3.5);
7         Mahasiswa22 m3 = new Mahasiswa22(nm:"125", name:"Husnil", kls:"2A", ip:3.4);
8         Mahasiswa22 m4 = new Mahasiswa22(nm:"126", name:"Biyyah", kls:"2A", ip:3.9);
9         Mahasiswa22 m5 = new Mahasiswa22(nm:"127", name:"Alif", kls:"2A", ip:3.7);
10
11         list.tambah(m1);
12         list.tambah(m2);
13         list.tambah(m3);
14         list.tambah(m4);
15         list.tambah(m5);
16
17         System.out.println(x:"Data Mahasiswa Belum sorting: ");
18         list.tampil();
19
20         System.out.println(x:"Data Mahasiswa setelah sorting berdasarkan IPK(DESC): ");
21         list.bubbleSort();
22         list.tampil();
23     }
24 }
```

k. Jalankan kode program dan lihat hasilnya.

```
D:\Praktikum-ASD>cd "d:\Praktikum-ASD\jobsheet6\" && java
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -D
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -D
Data Mahasiswa Belum sorting:
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Nama: Alif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Data Mahasiswa setelah sorting berdasarkan IPK(DESC):
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Alif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
```

2. Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

- a. Pada class MahasiswaBerprestasi22.java, tambahkan method selectionSort() didalamnya. Method ini akan melakukan proses sorting secara ascending.

```
jobsheet6 > J MahasiswaBerprestasi22.java > MahasiswaBerprestasi22 > bubbleSort()
1 public class MahasiswaBerprestasi22 {
33 void SelectionSort(){
34     for (int i = 0; i < listMhs.length-1; i++) {
35         int idxMin=i;
36         for (int j = i+1; j < listMhs.length; j++) {
37             if (listMhs[j].ipk<listMhs[idxMin].ipk) {
38                 idxMin = j;
39             }
40         }
41         Mahasiswa22 tmp = listMhs[idxMin];
42         listMhs[idxMin]=listMhs[i];
43         listMhs[i]= tmp;
44     }
45 }
46 }
```

- b. 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!

```
jobsheet6 > J MahasiswaDemo22.java > MahasiswaDemo22 > main(String[])
1 public class MahasiswaDemo22 {
2     public static void main(String[] args) {
3
4         MahasiswaBerprestasi22 list = new MahasiswaBerprestasi22();
5         Mahasiswa22 m1 = new Mahasiswa22(nm:"123", name:"Haydar", kls:"2A", ip:3.8);
6         Mahasiswa22 m2 = new Mahasiswa22(nm:"124", name:"Bintang", kls:"2A", ip:3.5);
7         Mahasiswa22 m3 = new Mahasiswa22(nm:"125", name:"Husnil", kls:"2A", ip:3.4);
8         Mahasiswa22 m4 = new Mahasiswa22(nm:"126", name:"Bilyyah", kls:"2A", ip:3.9);
9         Mahasiswa22 m5 = new Mahasiswa22(nm:"127", name:"Alif", kls:"2A", ip:3.7);
10
11         list.tambah(m1);
12         list.tambah(m2);
13         list.tambah(m3);
14         list.tambah(m4);
15         list.tambah(m5);
16
17         System.out.println(x:"Data Mahasiswa Belum sorting: ");
18         list.tampil();
19
20         System.out.println(x:"Data Mahasiswa setelah sorting berdasarkan IPK(DESC): ");
21         list.bubblesort();
22         list.tampil();
23
24         System.out.println(x:"Data yang sudah terurut menggunakan Selection Sort(ASC): ");
25         list.selectionSort();
26         list.tampil();
27     }
28 }
```

c. Jalankan kode program dan lihat hasilnya.

```
d:\Praktikum ASD\jobsheet6> cd "d:\Praktikum ASD\jobsheet6"
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstd
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstd
Data Mahasiswa Belum sorting:
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Nama: Aliif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Data Mahasiswa setelah sorting berdasarkan IPK(DESC):
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Aliif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
Data yang sudah terurut menggunakan Selection Sort(ASC):
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Aliif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
```

3. Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

a. Pada class MahasiswaBerprestasi22.java, tambahkan method InsertionSort() didalamnya. Method ini akan melakukan proses sorting secara ascending.

```
jobsheet6 > J MahasiswaBerprestasi22.java > MahasiswBerprestasi22 > SelectionSort()
1 public class MahasiswaBerprestasi22 {
47     void InsertionSort(){
48         for (int i = 1; i < listMhs.length; i++) {
49             Mahasiswa22 temp = listMhs[i];
50             int j = i;
51             while (j>0 && listMhs[j-1].ipk>temp.ipk) {
52                 listMhs[j]=listMhs[j-1];
53                 j--;
54             }
55             listMhs[j]=temp;
56         }
57     }
58 }
```

- b. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method InsertionSort() tersebut, kemudian panggil method tampil() untuk menampilkan data yang sudah diurutkan!

```
jobsheet6 > J MahasiswaDemo22.java > MahasiswaDemo22 > main(String[])
1  public class MahasiswaDemo22 {
2      Run | Debug
3      public static void main(String[] args) {
4          MahasiswaBerprestasi22 list = new MahasiswaBerprestasi22();
5          Mahasiswa22 m1 = new Mahasiswa22(nm:"123", name:"Haydar", kls:"2A", ip:3.8);
6          Mahasiswa22 m2 = new Mahasiswa22(nm:"124", name:"Bintang", kls:"2A", ip:3.5);
7          Mahasiswa22 m3 = new Mahasiswa22(nm:"125", name:"Husnil", kls:"2A", ip:3.4);
8          Mahasiswa22 m4 = new Mahasiswa22(nm:"126", name:"Biyyah", kls:"2A", ip:3.9);
9          Mahasiswa22 m5 = new Mahasiswa22(nm:"127", name:"Alif", kls:"2A", ip:3.7);
10
11         list.tambah(m1);
12         list.tambah(m2);
13         list.tambah(m3);
14         list.tambah(m4);
15         list.tambah(m5);
16
17         System.out.println(x:"Data Mahasiswa Belum sorting: ");
18         list.tampil();
19
20         System.out.println(x:"Data Mahasiswa setelah sorting berdasarkan IPK(DESC): ");
21         list.bubbleSort();
22         list.tampil();
23
24         System.out.println(x:"Data yang sudah terurut menggunakan Selection Sort(ASC): ");
25         list.SelectionSort();
26         list.tampil();
27
28         System.out.println(x:"Data yang sudah terurut menggunakan Insertion Sort(ASC): ");
29         list.InsertionSort();
30         list.tampil();
31     }
32 }
```

c. Jalankan kode program dan lihat hasilnya.

```
d:\Praktikum-ASD\jobsheet6>cd "d:\Praktikum-ASD\jobsheet6\" &&
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.
Data Mahasiswa belum sorting:
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Nama: Alif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Data Mahasiswa setelah sorting berdasarkan IPK(DESC):
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Alif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
Data yang sudah terurut menggunakan Selection Sort(ASC):
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Alif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
Data yang sudah terurut menggunakan Insertion Sort(ASC):
Nama: Husnil
NIM: 125
Kelas: 2A
IPK: 3.4
-----
Nama: Bintang
NIM: 124
Kelas: 2A
IPK: 3.5
-----
Nama: Alif
NIM: 127
Kelas: 2A
IPK: 3.7
-----
Nama: Haydar
NIM: 123
Kelas: 2A
IPK: 3.8
-----
Nama: Biyyah
NIM: 126
Kelas: 2A
IPK: 3.9
-----
```

Pertanyaan

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

Jawab:

Sebelum :

```
void InsertionSort(){
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa22 temp = listMhs[i];
        int j = i;
        while (j>0 && listMhs[j-1].ipk>temp.ipk) {
            listMhs[j]=listMhs[j-1];
            j--;
        }
        listMhs[j]=temp;
    }
}
```

Sesudah :

```
void InsertionSort(){
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa22 temp = listMhs[i];
        int j = i;
        while (j>0 && listMhs[j-1].ipk<temp.ipk) {
            listMhs[j]=listMhs[j-1];
            j--;
        }
        listMhs[j]=temp;
    }
}
```