

Nama : Zaskia Maulidina Mutiara Hati

NIM : 244107060056

Kelas : SIB 1A

JOBSHEET 6 ALGORITMA STRUKTUR DATA

Percobaan 1

➤ Class

```
jobsheet_6 > 🦋 Sorting24.java > ...
1  package jobsheet_6;
2
3  public class Sorting24 {
4      int [] data;
5      int jumData;
6
7      Sorting24 (int Data[], int jmlData){
8          jumData = jmlData;
9          data = new int[jmlData];
10         for (int i = 0; i < jumData; i++){
11             data[i] = Data[i];
12         }
13     }
14
15     void bubblesort(){
16         int temp = 0;
17         for (int i = 0; i < jumData-1; i++){
18             for (int j = 1; j < jumData-i; j++){
19                 if (data[j-1] > data[j]){
20                     temp = data[j];
21                     data[j] = data[j-1];
22                     data[j-1] = temp;
23                 }
24             }
25         }
26     }
27
28     void tampil(){
29         for(int i = 0; i < jumData; i++){
30             System.out.print(data[i]+" ");
31         }
32         System.out.println();
33     }
34 }
```

➤ Main

```
jobsheet_6 > SortingMain24.java > SortingMain24
1 package jobsheet_6;
2
3 public class SortingMain24 {
4     Run | Debug
5     public static void main(String[] args) {
6         int a[] = {20, 10, 2, 7, 12};
7         Sorting24 dataurut1 = new Sorting24(a, a.length);
8
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    }
```

➤ Output

```
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
```

• Selection Sort

```
28 void SelectionSort(){
29     for (int i=0; i < jumData-1; i++){
30         int min = i;
31         for (int j = i +1; j < jumData; j++){
32             if(data[j] < data[min]){
33                 min = j;
34             }
35         }
36         int temp = data[i];
37         data[i] = data [min];
38         data[min] = temp;
39     }
40 }
```

Main

```
14 int b[] = {30, 20, 2, 8, 14};
15 Sorting24 dataurut2 = new Sorting24(b, b.length);
16
17 System.out.println(x:"Data awal 2");
18 dataurut2.tampil();
19 dataurut2.SelectionSort();
20 System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT (ASC)");
21 dataurut2.tampil();
22 }
```

Output

```
Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30
```

- Insertion Sort

```
42 void insertionSort(){
43     for (int i = 1; i <= data.length-1; i++){
44         int temp = data[i];
45         int j = i-1;
46         while (j >= 0 && data[j] > temp){
47             data[j+1] = data[j];
48             j--;
49         }
50         data[j+1] = temp;
51     }
52 }
```

Main

```
23 int c[] = {40, 10, 4, 9, 3};
24 Sorting24 dataurut3 = new Sorting24(c, c.length);
25
26 System.out.println(x:"Data awal 3");
27 dataurut3.tampil();
28 dataurut3.insertionSort();
29 System.out.println(x:"Data sudah diurutkan dengan INSERTION SORT (ASC)");
30 dataurut3.tampil();
31
```

Output

```
Data awal 3
40 10 4 9 3
Data sudah diurutkan dengan INSERTION SORT (ASC)
3 4 9 10 40
```

Pertanyaan 1

1. Jelaskan fungsi kode program berikut!

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

- If (data[j-1] > data[j]) digunakan untuk cek apakah angka kiri lebih besar dari angka kanan, jika iya maka angka harus di swap menggunakan temp agar nilainya tidak hilang. Jadi jika ada dua angka yang urutannya salah, langsung di swap agar angka yang kecil ke kiri, dan angka yang besar ke kanan.

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

```
int min = i;  
for (int j = i + 1; j < jumData; j++){  
    if(data[j] < data[min]){  
        min = j;  
    }  
}
```

-

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

While (j >= 0 && data[j] > temp)

- Digunakan untuk cek apakah angka perlu bergeser atau tidak.

4. Pada Insertion sort, apakah tujuan dari perintah data[j + 1] = data [j];

- Digunakan untuk menggeser angka yang lebih besar ke kanan.

Percobaan 2 (Bubble Sort)

➤ Class Mahasiswa

```
jobsheet_6 > 🍷 Mahasiswa24.java > 🚧 Mahasiswa24
1  package jobsheet_6;
2
3  public class Mahasiswa24 {
4      String nim;
5      String nama;
6      String kelas;
7      double ipk;
8
9      Mahasiswa24(){
10
11     }
12
13     Mahasiswa24(String nm, String name, String kls, double ip){
14         nim = nm;
15         nama = name;
16         kelas = kls;
17         ipk = ip;
18     }
19
20     void tampilInformasi(){
21         System.out.println("Nama : "+ nama);
22         System.out.println("NIM : "+ nim);
23         System.out.println("IPK : "+ ipk);
24         System.out.println("Kelas : "+ kelas);
25     }
26 }
```

➤ Class Mahasiswa Berprestasi

```
jobsheet_6 > 🍷 MahasiswaBerprestasi24.java > 🚧 MahasiswaBerprestasi24 > 🏠 bubble
1  package jobsheet_6;
2
3  public class MahasiswaBerprestasi24 {
4      Mahasiswa24 [] listMhs = new Mahasiswa24 [5];
5      int idx;
6
7      void tambah(Mahasiswa24 m){
8          if (idx < listMhs.length){
9              listMhs[idx] = m;
10             idx++;
11         } else {
12             System.out.println(x:"Data sudah penuh!!");
13         }
14     }
15
16     void tampil(){
17         for (Mahasiswa24 m : listMhs){
18             m.tampilInformasi();
19             System.out.println(x:"-----");
20         }
21     }
22
23     void bubbleSort(){
24         for (int i = 0; i < listMhs.length-1; i++){
25             for (int j = 1; j < listMhs.length-i; j++){
26                 if (listMhs[j].ipk > listMhs[j-1].ipk){
27                     Mahasiswa24 tmp = listMhs[j];
28                     listMhs[j] = listMhs[j-1];
29                     listMhs[j-1] = tmp;
30                 }
31             }
32         }
33     }
34 }
```

➤ Mahasiswa Demo

```
jobsheet_6 > MahasiswaDemo24.java > MahasiswaDemo24
1  package jobsheet_6;
2
3  public class MahasiswaDemo24 {
    Run | Debug
4      public static void main(String[] args) {
5          MahasiswaBerprestasi24 list = new MahasiswaBerprestasi24();
6          Mahasiswa24 m1 = new Mahasiswa24(nm:"123", name:"Zidan", kls:"2A", ip:3.2);
7          Mahasiswa24 m2 = new Mahasiswa24(nm:"124", name:"Ayu", kls:"2A", ip:3.5);
8          Mahasiswa24 m3 = new Mahasiswa24(nm:"125", name:"Sofi", kls:"2A", ip:3.1);
9          Mahasiswa24 m4 = new Mahasiswa24(nm:"126", name:"Sita", kls:"2A", ip:3.9);
10         Mahasiswa24 m5 = new Mahasiswa24(nm:"127", name:"Miki", kls:"2A", ip:3.7);
11
12         list.tambah(m1);
13         list.tambah(m2);
14         list.tambah(m3);
15         list.tambah(m4);
16         list.tambah(m5);
17
18         System.out.println(x:"Data mahasiswa sebelum sorting: ");
19         list.tampil();
20
21         System.out.println(x:"Data mahasiswa setelah sorting berdasarkan IPK (DESC): ");
22         list.bubbleSort();
23         list.tampil();
24     }
25 }
```

➤ Output

```
Data mahasiswa sebelum sorting:
Nama : Zidan
NIM : 123
IPK : 3.2
Kelas : 2A
-----
Nama : Ayu
NIM : 124
IPK : 3.5
Kelas : 2A
-----
Nama : Sofi
NIM : 125
IPK : 3.1
Kelas : 2A
-----
Nama : Sita
NIM : 126
IPK : 3.9
Kelas : 2A
-----
Nama : Miki
NIM : 127
IPK : 3.7
Kelas : 2A
-----
```

24

```
Data mahasiswa setelah sorting berdasarkan IPK (DESC):
Nama : Sita
NIM : 126
IPK : 3.9
Kelas : 2A
-----
Nama : Miki
NIM : 127
IPK : 3.7
Kelas : 2A
-----
Nama : Ayu
NIM : 124
IPK : 3.5
Kelas : 2A
-----
Nama : Zidan
NIM : 123
IPK : 3.2
Kelas : 2A
-----
Nama : Sofi
NIM : 125
IPK : 3.1
Kelas : 2A
-----
```

Pertanyaan 2 (Bubble Sort)

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

```
for (int i=0; i<listMhs.length-1; i++){  
    for (int j=1; j<listMhs.length-i; j++){
```

- Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$?
 - Karena di algoritma sorting, setelah $\text{listMhs.length} - 1$ iterasi, array sudah pasti terurut, jadi tidak perlu cek sampai elemen terakhir.
 - Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?
 - Karena setiap iterasi i, elemen terbesar sudah pasti berada di tempatnya, jadi tiap iterasi mengurangi jumlah perbandingan j untuk efisiensi.
 - Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?
 - Perulangan akan berlangsung 49 kali, dan 49 tahap.
2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyboard) yang terdiri dari nim, nama, kelas, dan ipk!

```
4      public static void main(String[] args) {  
5          MahasiswaBerprestasi24 list = new MahasiswaBerprestasi24();  
6          Scanner input24 = new Scanner(System.in);  
7  
8          System.out.print(s:"Masukkan jumlah data mahasiswa : ");  
9          int jumlah = input24.nextInt();  
10         input24.nextLine();  
11  
12         Mahasiswa24[] arrayOfMahasiswa = new Mahasiswa24[jumlah];  
13  
14         for (int i = 0; i < jumlah; i++) {  
15             System.out.println("Masukkan data mahasiswa ke-" + (i + 1));  
16             System.out.print(s:"NIM : ");  
17             String nim = input24.nextLine();  
18             System.out.print(s:"Nama : ");  
19             String nama = input24.nextLine();  
20             System.out.print(s:"IPK : ");  
21             double ipk = input24.nextDouble();  
22             input24.nextLine();  
23             System.out.print(s:"Kelas : ");  
24             String kelas = input24.nextLine();  
25             System.out.println(x:"-----");  
26             arrayOfMahasiswa[i] = new Mahasiswa24(nim, nama, kelas, ipk);  
27             list.tambah(arrayOfMahasiswa[i]);  
28         }  
29  
30  
31         System.out.println(x:"Data Mahasiswa sebelum sorting: ");  
32         list.tampil();  
33  
34         System.out.println(x:"Data Mahasiswa setelah sorting berdasarkan IPK (DESC) : ");  
35         list.bubbleSort();  
36         list.tampil();
```

Percobaan 2 (Selection Sort)

➤ Class Mahasiswa Berprestasi

```
35     void SelectionSort(){
36         for (int i = 0; i < listMhs.length-1; i++){
37             int idxMin = i;
38             for (int j = i+1; j < listMhs.length; j++){
39                 if (listMhs[j].ipk < listMhs[idxMin].ipk){
40                     idxMin = j;
41                 }
42             }
43             Mahasiswa24 tmp = listMhs[idxMin];
44             listMhs[idxMin] = listMhs[i];
45             listMhs[i] = tmp;
46         }
47     }
48 }
```

➤ Mahasiswa Demo

```
38     System.out.println(x:"Data yang sudah terurut menggunakan SELECTION SORT (ASC) : ");
39     list.SelectionSort();
40     list.tampil();
```

➤ Output

```
Masukkan jumlah data mahasiswa : 5
Masukkan data mahasiswa ke-1
NIM : 123
Nama : Ali
Kelas : 2B
IPK : 3,9
-----
Masukkan data mahasiswa ke-2
NIM : 124
Nama : ila
Kelas : 2B
IPK : 3,1
-----
Masukkan data mahasiswa ke-3
NIM : 125
Nama : agus
Kelas : 2B
IPK : 3,6
-----
Masukkan data mahasiswa ke-4
NIM : 126
Nama : tika
Kelas : 2B
IPK : 3,3
-----
Masukkan data mahasiswa ke-5
NIM : 127
Nama : udin
Kelas : 2B
IPK : 3,2
-----
```



```
Data yang sudah terurut menggunakan SELECTION SORT (ASC) :
Nama : ila
NIM : 124
IPK : 3.1
Kelas : 2B
-----
Nama : udin
NIM : 127
IPK : 3.2
Kelas : 2B
-----
Nama : tika
NIM : 126
IPK : 3.3
Kelas : 2B
-----
Nama : agus
NIM : 125
IPK : 3.6
Kelas : 2B
-----
Nama : Ali
NIM : 123
IPK : 3.9
Kelas : 2B
-----
```

Pertanyaan 2 (Selection Sort)

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

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

Untuk apakah proses tersebut, jelaskan!

- Baris program di atas digunakan untuk **mencari nilai minimum dalam array** dengan membandingkan setiap elemen setelah indeks i. Jika ditemukan angka lebih kecil, indeksnya disimpan dalam idxMin. Proses ini digunakan dalam **Selection Sort** untuk menempatkan angka terkecil ke posisi yang benar secara bertahap hingga array terurut.

Percobaan 2 (Insertion Sort)

➤ Class Mahasiswa Berprestasi

```
49 void insertionSort(){
50     for (int i = 1; i < listMhs.length; i++){
51         Mahasiswa24 temp = listMhs[i];
52         int j = i;
53         while (j > 0 && listMhs[j-1].ipk > temp.ipk){
54             listMhs[j] = listMhs[j-1];
55             j--;
56         }
57         listMhs[j] = temp;
58     }
59 }
```

➤ Class Mahasiswa Demo

```
42 System.out.println(x:"Data yang sudah terurut menggunakan INSERTION SORT (ASC) : ");
43 list.insertionSort();
44 list.tampil();
```

➤ Output

```
Masukkan jumlah data mahasiswa : 5
Masukkan data mahasiswa ke-1
NIM : 111
Nama : ayu
Kelas : 2c
IPK : 3,7
-----
Masukkan data mahasiswa ke-2
NIM : 222
Nama : dika
Kelas : 2c
IPK : 3,0
-----
Masukkan data mahasiswa ke-3
NIM : 333
Nama : ila
Kelas : 2c
IPK : 3,8
-----
Masukkan data mahasiswa ke-4
NIM : 444
Nama : susi
Kelas : 2c
IPK : 3,1
-----
Masukkan data mahasiswa ke-5
NIM : 555
Nama : yayuk
Kelas : 2c
IPK : 3,4
-----
```

```
Data yang sudah terurut menggunakan INSERTION SORT (ASC) :
Nama : dika
NIM : 222
IPK : 3,0
Kelas : 2c
-----
Nama : susi
NIM : 444
IPK : 3,1
Kelas : 2c
-----
Nama : yayuk
NIM : 555
IPK : 3,4
Kelas : 2c
-----
Nama : ayu
NIM : 111
IPK : 3,7
Kelas : 2c
-----
Nama : ila
NIM : 333
IPK : 3,8
Kelas : 2c
-----
```

Pertanyaan 2 (Insertion Sort)

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

```
49 void insertionSort(){
50     for (int i = 1; i < listMhs.length; i++){
51         Mahasiswa24 temp = listMhs[i];
52         int j = i;
53         while (j > 0 && listMhs[j-1].ipk < temp.ipk){
54             listMhs[j] = listMhs[j-1];
55             j--;
56         }
57         listMhs[j] = temp;
58     }
59 }
```

Latihan Praktikum

➤ Class Dosen

```
jobsheet_6 > Dosen24.java > Dosen24
1  package jobsheet_6;
2
3  public class Dosen24 {
4      String kode;
5      String nama;
6      Boolean jenisKelamin;
7      int usia;
8
9      Dosen24(){
10
11     }
12
13     Dosen24(String kd, String name, Boolean jk, int age){
14         kode = kd;
15         nama = name;
16         jenisKelamin = jk;
17         usia = age;
18     }
19
20     void tampil(){
21         System.out.println("Kode : "+ kode);
22         System.out.println("Nama : "+ nama);
23         System.out.println("Jenis Kelamin : "+ jenisKelamin);
24         System.out.println("Usia : "+ usia);
25     }
26 }
```

➤ Class Data Dosen

```
jobsheet_6 > DataDosen24.java > DataDosen24
1  package jobsheet_6;
2
3  public class DataDosen24 {
4      Dosen24 [] listDos = new Dosen24[10];
5      int idx;
6
7      void tambah(Dosen24 d){
8          if (idx < listDos.length){
9              listDos[idx] = d;
10             idx++;
11         } else {
12             System.out.println("Data sudah penuh!!");
13         }
14     }
15
16     void tampil(){
17         for (Dosen24 d : listDos){
18             d.tampil();
19             System.out.println("-----");
20         }
21     }
22
23     void bubbleSortASC(){
24         for (int i = 0; i < listDos.length-1; i++){
25             for (int j = 1; j < listDos.length-i; j++){
26                 if (listDos[j].usia < listDos[j-1].usia){
27                     Dosen24 tmp = listDos[j];
28                     listDos[j] = listDos[j-1];
29                     listDos[j-1] = tmp;
30                 }
31             }
32         }
33     }
```

```

35     void selectionSortDSC(){
36         for (int i = 0; i < listDos.length-1; i++){
37             int idxMax = i;
38             for (int j = i+1; j < listDos.length; j++){
39                 if (listDos[j].usia > listDos[idxMax].usia){
40                     idxMax = j;
41                 }
42             }
43             Dosen24 tmp = listDos[idxMax];
44             listDos[idxMax] = listDos[i];
45             listDos[i] = tmp;
46         }
47     }
48
49     void insertionSortDSC(){
50         for (int i = 1; i < listDos.length; i++){
51             Dosen24 tmp = listDos[i];
52             int j = i;
53             while (j > 0 && listDos[j-1].usia < tmp.usia){
54                 listDos[j] = listDos[j-1];
55                 j--;
56             }
57             listDos[j] = tmp;
58         }
59     }
60 }

```

➤ Class Dosen Demo

```

jobsheet_6 > DosenDemo24.java
1  package jobsheet_6;
2  import java.util.Scanner;
3
4  public class DosenDemo24 {
5      public static void main(String[] args) {
6          Scanner sc = new Scanner(System.in);
7          DataDosen24 dataDosen = new DataDosen24();
8
9          while (true) {
10             System.out.println("\nMenu:");
11             System.out.println("1. Tambah Data Dosen");
12             System.out.println("2. Tampilkan Data Dosen");
13             System.out.println("3. Sorting ASC (Usia Termuda ke Tertua)");
14             System.out.println("4. Sorting DSC (Usia Tertua ke Termuda)");
15             System.out.println("5. Keluar");
16             System.out.print("Pilih menu: ");
17             int pilihan = sc.nextInt();
18             sc.nextLine();

```

```

5      public static void main(String[] args) {
20          switch (pilihan) {
21              case 1:
22                  System.out.print(s:"Berapa banyak dosen yang ingin dimasukkan? ");
23                  int jumlah = sc.nextInt();
24                  sc.nextLine();
25
26                  for (int i = 0; i < jumlah; i++) {
27                      System.out.println("Data Dosen ke-" + (i + 1) + ":");
28                      System.out.print(s:"Masukkan kode: ");
29                      String kode = sc.nextLine();
30                      System.out.print(s:"Masukkan nama: ");
31                      String nama = sc.nextLine();
32                      System.out.print(s:"Masukkan jenis kelamin (true untuk pria, false untuk wanita): ");
33                      boolean jk = sc.nextBoolean();
34                      System.out.print(s:"Masukkan usia: ");
35                      int usia = sc.nextInt();
36                      sc.nextLine();
37
38                      Dosen24 dsn = new Dosen24(kode, nama, jk, usia);
39                      dataDosen.tambah(dsn);
40                  }
41                  break;
42
43                  case 2:
44                      dataDosen.tampil();
45                      break;
46                  case 3:
47                      dataDosen.bubbleSortASC();
48                      System.out.println(x:"Data berhasil diurutkan ASC (Termuda ke Tertua).\n");
49                      break;
50                  case 4:
51                      System.out.println(x:"Pilih metode sorting:");
52                      System.out.println(x:"1. Selection Sort");
53                      System.out.println(x:"2. Insertion Sort");
54                      System.out.print(s:"Pilihan: ");
55                      int sortMethod = sc.nextInt();
56
57                      if (sortMethod == 1) {
58                          dataDosen.selectionSortDSC();
59                      } else {
60                          dataDosen.insertionSortDSC();
61                      }
62                      System.out.println(x:"Data berhasil diurutkan DSC (Tertua ke Termuda).\n");
63                      break;
64                  case 5:
65                      System.out.println(x:"Keluar dari program...");
66                      sc.close();
67                      return;
68                  default:
69                      System.out.println(x:"Pilihan tidak valid!");

```