

## PRAKTIKUM

1. Ada di line 28 sampai 39

```
27 //setelah ini tuliskan method bubbleSort ()
28 void bubbleSort () {
29     for (int i = 0; i < listMhs.length-1; i++) {
30         for (int j = 1; j < listMhs.length-i; j++) {
31             if (listMhs [j].ipk >listMhs [j-1].ipk){
32                 // dibawah ini proses swap atau penukaran
33                 Mahasiswa tmp = listMhs [j];
34                 listMhs [j] = listMhs [j-1];
35                 listMhs [j-1] = tmp;
36             }
37         }
38     }
39 }
```

2. Ada di line 40 sampai 52

```
40 void selectionSort() {
41     for (int i = 0; i < listMhs.length; i++) {
42         int idxMin = i;
43         for (int j = i+1; j <listMhs.length; j++) {
44             if ( listMhs [j].ipk < listMhs[idxMin].ipk)
45                 idxMin = j;
46         }
47         //swap
48         Mahasiswa tmp = listMhs [idxMin];
49         listMhs [idxMin] = listMhs [i];
50         listMhs [i] = tmp;
51     }
52 }
```

3. Proses Swap yaitu proses penukaran dengan cara mencari nilai terkecil dari deretan data terlebih dahulu untuk kemudian dilakukan proses penukaran (Swap).

```
48         //swap
49         Mahasiswa tmp = listMhs [idxMin];
50         listMhs [idxMin] = listMhs [i];
51         listMhs [i] = tmp;
52     }
53 }
```

4. Code program  
Swap ( penukaran)

```
32 // dibawah ini proses swap atau penukaran
33 Mahasiswa tmp = listMhs[j];
34 listMhs[j] = listMhs[j-1];
35 listMhs[j-1] = tmp;
36 }
37 }
38 }
39 }
```

Untuk melakukan penukaran dimana Jumlah Iterasi sebesar banyaknya Data. Diatas jumlah datanya ialah 5 maka, jumlah iterasinya ialah 5. Selain itu, setiap iterasi terdapat proses yang jumlahnya ialah sebesar banyaknya Data. Diatas jumlah datanya ialah 5 maka, jumlah proses setiap iterasinya ialah 5. Dan untuk iterasi berikutnya harus dikurang 1.

Maksudnya yaitu dimana program ini seolah-olah menggeser satu demi satu elemen dari kanan ke kiri atau dari kiri ke kanan tergantung pada jenis pengurutannya. Jenis pengurutan sorting ada 2 yaitu ascending dan descending. Dimana ascending itu mengurut data dari kecil ke besar dan descending itu mengurut data dari besar ke kecil. Jika semua elemen sudah diperiksa oleh fungsi bubble sort, dan tidak ada pertukaran lagi atau semua nilai sudah sesuai, maka saat itu program bubble sort akan berhenti bekerja. Pola Bubble sort yaitu bertukar dengan data sebelahnya.

## 5. Code program

```
27 //setelah ini tuliskan method bubbleSort ()
28 void bubbleSort () {
29     for (int i = 0; i < listMhs.length-1; i++) {
30         for (int j = 1; j < listMhs.length-i; j++) {
```

- Perulangan i digunakan untuk pada perulangan i memulai perulangan j dari awal, Sedangkan Perulangan j digunakan untuk perulangan j berada di dalam untuk melakukan penukaran indeks.
- Perulangan i adalah  $i < \text{listMhs.length}-1$  memenuhi 2 proses perulangan namun dari index 0
- Perulangan j adalah  $j < \text{listMhs.length}-i$  untuk memenuhi 2 proses perulangan namun mulai dari index 1
- Perulangan i akan berlangsung sebanyak 49 kali. Banyak tahap bubbleSort ada 1225

## 6. Code program

```
40 //setelah ini tuliskan method selectionSort ()
41 void selectionSort() {
42     for (int i = 0; i < listMhs.length; i++) {
43         int idxMin = i;
44         for (int j = i+1; j < listMhs.length; j++) {
45             if (listMhs[j].ipk < listMhs[idxMin].ipk)
46                 idxMin = j;
47         }
```

Maksud proses tersebut untuk mencari nilai terkecil untuk dilakukan penukaran.

Selection sort merupakan metode pengurutan dengan mencari nilai data terkecil dimulai dari data diposisi 0 hingga diposisi N-1. Jika terdapat N data dan data terkoleksi dari urutan 0 sampai dengan N-1. Selama proses, perbandingan dan pengubahan, hanya dilakukan pada indeks perbandingnya saja, pertukaran data secara fisik terjadi pada akhir proses.

## 7. Code program

```
54 //setelah ini tuliskan method insertSort ()
55 void insertionSort(boolean ascending ){
56     int i,j;
57     for (i = 1; i <listMhs.length; i++) {
58         Mahasiswa temp = new Mahasiswa();
59         temp = listMhs [i];
60         j = i;
61         if (ascending){
62             while ((j>0)&& (listMhs[j-1].ipk>temp.ipk)){
63                 listMhs [j] = listMhs [j-1];
64                 j--;
65             }
66             listMhs [j]= temp;
67         }else {
68             while ((j>0)&& (listMhs[j-1].ipk>temp.ipk)){
69                 listMhs [j] = listMhs [j-1];
70                 j--;
71             }
72         }
73         listMhs [j]= temp;
74     }
75 }
76 }
77 }
```