

## PRAKTIKUM

1. Ada di line 28 sampai 38

```
27  
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                 Mahasiswa tmp = listMhs[j];  
33                 listMhs[j] = listMhs[j - 1];  
34                 listMhs[j - 1] = tmp;  
35             }  
36         }  
37     }  
38 }
```

2. Ada di line 39 sampai 51

```
38 }  
39 void selectionSort(){  
40     for (int i=0; i<listMhs.length; i++){  
41         int idxMin = i;  
42         for (int j=i+1; j<listMhs.length; j++){  
43             if (listMhs[j].ipk < listMhs[idxMin].ipk){  
44                 idxMin = j;  
45             }  
46         }  
47         Mahasiswa tmp = listMhs[idxMin];  
48         listMhs [idxMin] = listMhs [i];  
49         listMhs [i] = tmp;  
50     }  
51 }
```

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

```
47     Mahasiswa tmp = listMhs[idxMin];  
48     listMhs [idxMin] = listMhs [i];  
49     listMhs [i] = tmp;  
50 }  
51 }
```

4. Untuk melakukan penukaran sebesar banyaknya Data. Jumlah datanya diatas 5 maka, jumlah iterasinya ialah 5. Setiap iterasi terdapat proses yang jumlahnya ialah sebesar banyaknya Data. Jumlah datanya diatas 5 maka, jumlah proses setiap iterasinya ialah 5. Dan untuk iterasi berikutnya harus dikurang 1. 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 asscending dan descending. Dimana asscending 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.

```

32 |         Mahasiswa tmp = listMhs[j];
33 |         listMhs[j] = listMhs[j - 1];
34 |         listMhs[j - 1] = tmp;
35 |     }
36 | }

```

5. - 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

```

27 |
28 | void bubbleSort() {
29 |     for (int i = 0; i < listMhs.length - 1; i++) {
30 |         for (int j = 1; j < listMhs.length - i; j++) {

```

6. Untuk melakukan penukaran/swap dari nilai terkecil. Selection sort merupakan teknik sorting yang paling sederhana, hal pertama yang akan dilakukan algoritma selection sort adalah menemukan elemen terkecil dalam array kita dan menukarnya (swap) dengan elemen yang ada di posisi pertama, kemudian algoritma ini akan mengulangi hal yang sama lagi yaitu mencari elemen terkecil yang ada di dalam array dan kemudian menukarnya (swap) dengan elemen yang ada di posisi kedua (mengingat elemen di posisi pertama sudah berhasil kita sorting). Proses ini akan terus berlanjut sampai semua elemen yang ada di dalam array telah berhasil kita sorting.

```

39 | void selectionSort(){
40 |     for (int i=0; i<listMhs.length; i++){
41 |         int idxMin = i;
42 |         for (int j=i+1; j<listMhs.length; j++){
43 |             if (listMhs[j].ipk < listMhs[idxMin].ipk){
44 |                 idxMin = j;
45 |             }
46 |         }

```

7.

```
52 void insertionSort(){  
53     int i, j;  
54     for (i=1; i<listMhs.length; i++){  
55         Mahasiswa temp = new Mahasiswa ();  
56         temp = listMhs[i];  
57         j = i;  
58         while ((j>0) && (listMhs[j - 1].ipk > temp.ipk)){  
59             listMhs[j] = listMhs[j - 1];  
60             j--;  
61         }  
62         listMhs[j] = temp;  
63     }  
64 }  
65 }  
66 }
```