

Nama : Malik

Muhammad

NIM : L200180072

Kelas : C

## Modul 5

### Laporan Praktikum

Routine swap untuk menukar A[p] dan A[q]

Berikut adalah screenshoot dari program yang saya buat:

```
##Swap
def swap(A,p,q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp
```

Berikut adalah screenshoot saat program dijalankan:

```
===== RESTART: C:/Users
>>> K = [50,20,70,10]
>>> swap(K,1,3)
>>> K
[50, 10, 70, 20]
>>> |
```

Routine untuk mencari index dari elemen yang

terkecil Berikut adalah screenshoot dari program

yang saya buat:

```
##cariPosisiTerkecil
def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini          #-> anggap ini yang terkecil
    for i in range(dariSini+1, sampaiSini): #-> cari di sisa list
        if A[i] < A[posisiYangTerkecil]:    #-> kalau menemukan yang lebih kecil
            posisiYangTerkecil = i          #-> anggap dirubah
    return posisiYangTerkecil
```

Berikut adalah screenshoot saat program dijalankan:

```
>>> A = [18,13,44,25,66,107,78,89]
>>> j = cariPosisiYangTerkecil(A,2,len(A))
>>> j
3
>>> |
```

## Bubble Sort

Berikut adalah screenshoot dari program yang saya buat:

```
def bubbleSort(A):  
    n = len(A)  
    for i in range(n-1):          #-> Lakukan operasi gelmbung sebanyak n-1  
        for j in range(n-i-1):    #->   Dorong elemen terbesar ke kanan  
            if A[j] > A[j+1]:      #->   Jika di kiri lebih besar dari di kanannya,  
                swap(A,j,j+1)      #->   tukakar posisi elemen ke j dengan ke j+1
```

Berikut adalah screenshoot saat program dijalankan:

```
>>> A = [18,13,44,25,66,107,78,89]  
>>> L = [10,51,2,18,4,31,13,5,23,64,29]  
>>> bubbleSort(A)  
>>> A  
[13, 18, 25, 44, 66, 78, 89, 107]  
>>> bubbleSort(L)  
>>> L  
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]  
>>>
```

Pertanyaan :

Dengan elemen sebanyak n, berapa banyakkah operasi perbandingan dan pertukaran yang dilakukan oleh algoritma bubble sort ini? Selidiki nilainya untuk worst-case, average-case dan best-case scenario.

Jawab :

Rumus :

Worst Case Time Complexity [ Big-O ]:

$O(n^2)$  Best Case Time Complexity [Big-

omega]:  $O(n)$  Average Time Complexity

[Big-theta]:  $O(n^2)$

Bersadarkan rumus tersebut, maka algoritma bubble sort ini ada 107 operasi perbandingan dan pertukaran.

## Selection Sort

Berikut adalah screenshoot dari program yang saya buat:

```
--  
##SelectionSort  
def selectionSort(A):  
    n = len(A)  
    for i in range(n-1):  
        indexKecil = cariPosisiYangTerkecil(A, i, n)  
        if indexKecil != i:  
            swap(A, i, indexKecil)
```

Berikut adalah screenshoot saat program dijalankan:

```
>>> A = [18,13,44,25,66,107,78,89]  
>>> L = [10,51,2,18,4,31,13,5,23,64,29]  
>>> selectionSort(A)  
>>> A  
[13, 18, 25, 44, 66, 78, 89, 107]  
>>> selectionSort(L)  
>>> L  
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]  
>>> |
```

Pertanyaan :

Dengan elemen sebanyak  $n$ , berapa banyakkah operasi pembandingan dan pertukaran yang dilakukan oleh algoritma selection sort ini? Selidiki nilainya untuk worst-case, average-case dan best-case scenario.

Jawab :

## Insertion Sort

Berikut adalah screenshoot dari program yang saya buat:

```
--  
##InsertionSort  
def insertionSort(A):  
    n = len(A)  
    for i in range(1,n):  
        nilai = A[i]  
        pos = i  
        while pos > 0 and nilai < A[pos - 1]: #-> cari posisi yang tepat  
            A[pos]=A[pos-1] # dan geser ke kanan terus  
            pos = pos - 1 # nilai-nilai yang lebih besar  
        A[pos]=nilai #-> pada posisi ini tempatkan nilai elemen ke i
```