

Nama : Annisa Nugraheni

NIM : L200180066

Kelas : C

## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

### MODUL 5

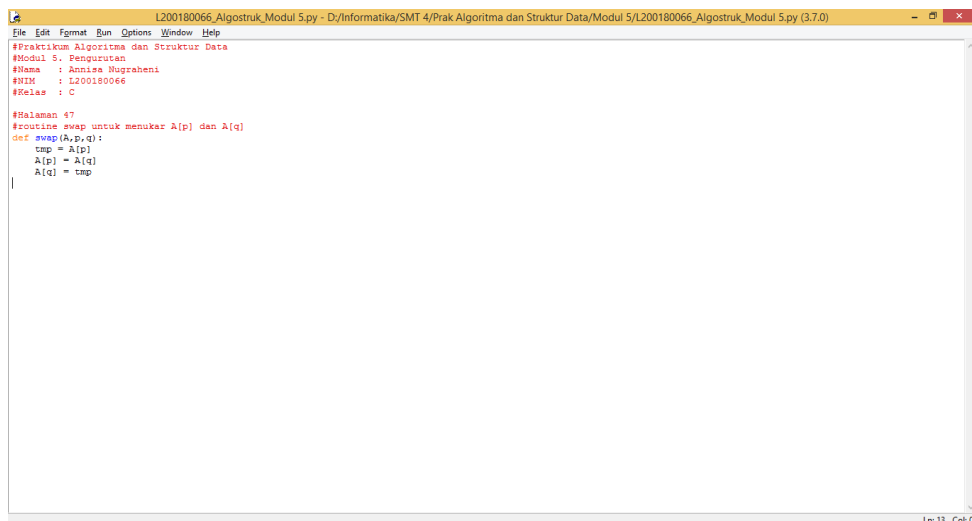
### PENGURUTAN

#### Latihan

#### ❖ Halaman 47

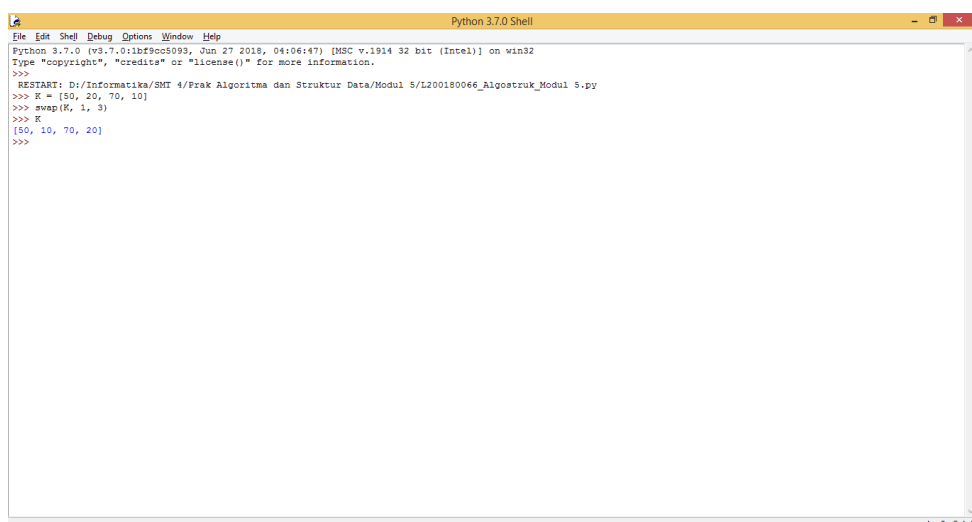
#### Routine swap untuk menukar A[p] dan A[q]

Berikut adalah screenshot program dan hasil:



```
L200180066_Algostruk_Modul 5.py - D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 5/L200180066_Algostruk_Modul 5.py (3.7.0)
File Edit Format Run Options Window Help
#Praktikum Algoritma dan Struktur Data
#Modul 5. Pengurutan
#Nama : Annisa Nugraheni
#NIM : L200180066
#Kelas : C

#Halaman 47
#routine swap untuk menukar A[p] dan A[q]
def swap(A,p,q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp
```



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:01bf90c5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 5/L200180066_Algostruk_Modul 5.py
>>> K = [50, 20, 70, 10]
>>> swap(K, 1, 3)
>>> K
[50, 10, 70, 20]
>>>
```

## ❖ Halaman 48

### Routine untuk mencari index dari elemen yang terkecil

Berikut adalah screenshot dari program dan hasil:

```
#Halaman 48
#routine untuk mencari index dari elemen yang terkecil
def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1, sampaiSini):
        if A[i] < A[posisiYangTerkecil]:
            posisiYangTerkecil = i
    return posisiYangTerkecil

#-> anggap ini yang terkecil
#-> cari di sisa list
#-> kalo menemukan yang lebih kecil
#-> anggapan diubah

>>>
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 5/L200180066_Algostruk_Modul 5.py
>>> A = [18, 13, 44, 25, 66, 107, 78, 89]
>>> j = cariPosisiYangTerkecil(A, 2, len(A))
>>> j
3
>>>
```

## ❖ Halaman 49

### 5.1 Bubble Sort

Berikut adalah screenshot dari program dan hasil :

```
#Halaman 49
#5.1 Bubble Sort
def bubbleSort(A):
    n = len(A)
    for i in range(n-1):
        for j in range(n-i-1):
            if A[j] > A[j+1]:
                swap(A, j, j+1)

#-> lakukan operasi gelembung sebanyak n-1
#-> dorong elemen terbesar ke ujung kanan
#-> jika dikiri lebih besar dari di kanannya
#-> tukar posisi elemen ke j dengan ke j+1

'''
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 5/L200180066_Algostruk_Modul 5.py
>>> A = [18, 13, 44, 25, 66, 107, 78, 89]
>>> bubbleSort(A)
>>> A
[13, 18, 25, 44, 66, 78, 89, 107]
>>> L = [10,51,2,18,4,31,13,5,23,64,29]
>>> 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]:  $O(n^2)$

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

## ❖ Halaman 50

### 5.2 Selection Sort

Berikut adalah screenshot dari program dan hasil :

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

```
'''
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 5/L200180066_Algostruk_Modul 5.py
>>> A = [18, 13, 44, 25, 66, 107, 78, 89]
>>> selectionSort(A)
>>> A
[13, 18, 25, 44, 66, 78, 89, 107]
>>> L = [10,51,2,18,4,31,13,5,23,64,29]
>>> selectionSort(L)
>>> L
[10, 2, 18, 4, 31, 13, 5, 23, 51, 29, 64]
>>> |
```

Ln: 148 Col: 4

## ❖ Halaman 52

### 5.3 Insertion Sort

Berikut adalah screenshot dari program dan hasil :

```
#5.3 Insertion Sort
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
            # dan geser ke kanan terus
            A[pos] = A[pos-1]
            pos = pos - 1
            # nilai - nilai yang lebih besar
            #-> pada posisi ini tempatkan nilai elemen ke i
        A[pos] = nilai
```

Ln: 34 Col: 98

```
'''
RESTART: D:/Informatika/SMT 4/Prak Algoritma dan Struktur Data/Modul 5/L200180066_Algostruk_Modul 5.py
>>> A = [18, 13, 44, 25, 66, 107, 78, 89]
>>> insertionSort(A)
>>> A
[13, 18, 25, 44, 66, 78, 89, 107]
>>> L = [10,51,2,18,4,31,13,5,23,64,29]
>>> insertionSort(L)
>>> L
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>>
```

Ln: 166 Col: 4