

Nama : Afiq Tri Nugraha

NIM : L200180080

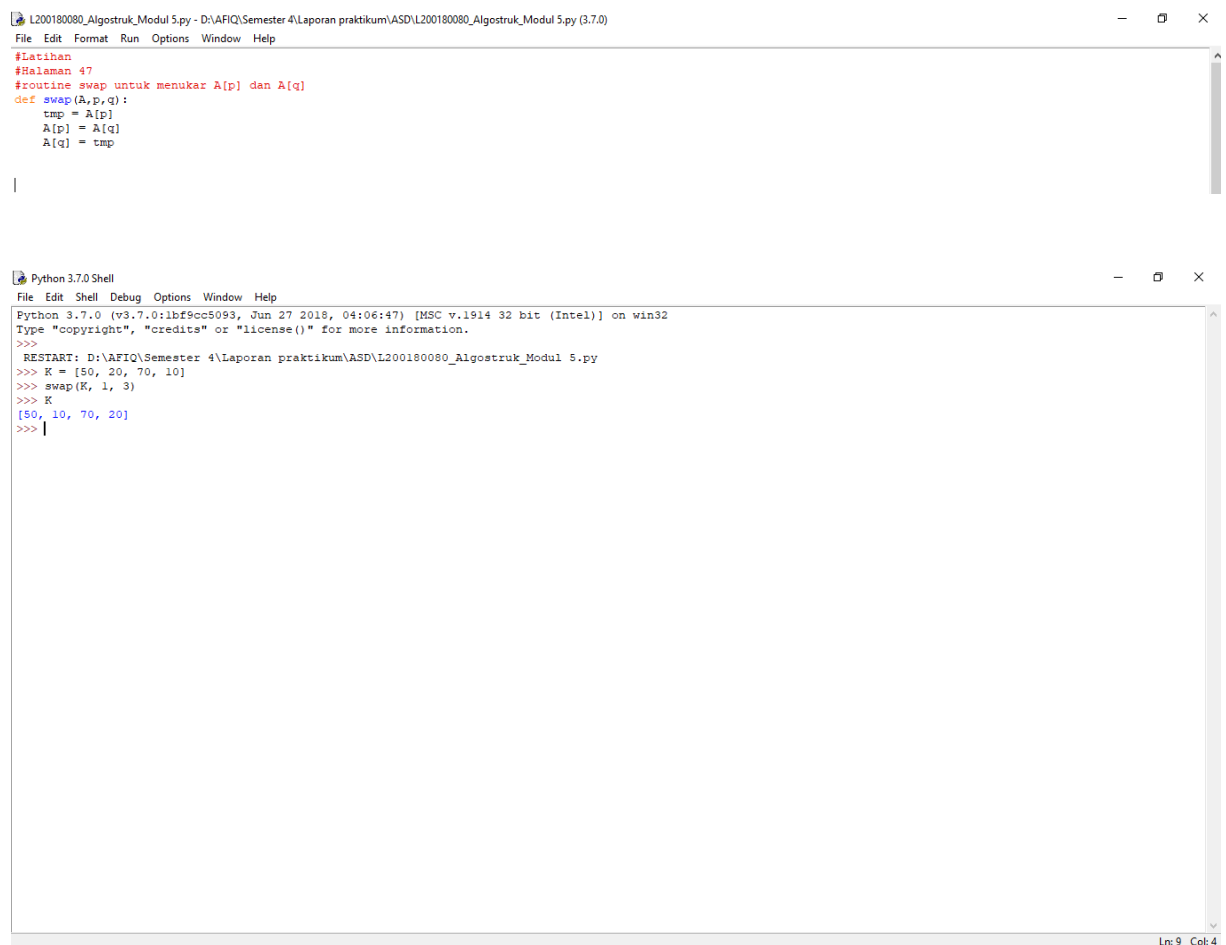
Kelas : C

## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

### MODUL 5

#### Latihan

#### Halaman 47



The image shows two windows from a computer screen. The top window is a text editor titled 'L200180080\_Algostruk\_Modul 5.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\L200180080\_Algostruk\_Modul 5.py (3.7.0)'. It contains a Python function definition for swapping two elements in a list. The bottom window is a 'Python 3.7.0 Shell' showing the execution of the script. It starts with a 'RESTART' message, then defines a list 'K' with values [50, 20, 70, 10]. The 'swap' function is called with arguments 'K, 1, 3'. Finally, the list 'K' is printed, showing the result [50, 10, 70, 20].

```
L200180080_Algostruk_Modul 5.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\L200180080_Algostruk_Modul 5.py (3.7.0)
File Edit Format Run Options Window Help
#Latihan
#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 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\L200180080_Algostruk_Modul 5.py
>>> K = [50, 20, 70, 10]
>>> swap(K, 1, 3)
>>> K
[50, 10, 70, 20]
>>> |
```

#### Halaman 48

```
#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
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\L200180080_Algostruk_Modul 5.py
>>> A = [18, 13, 44, 25, 66, 107, 78, 89]
>>> j = cariPosisiYangTerkecil(A, 2, len(A))
>>> j
3
>>> |
```

## Halaman 49

```
#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
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\L200180080_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 pembandingan 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 ]:  $O(n^2)$

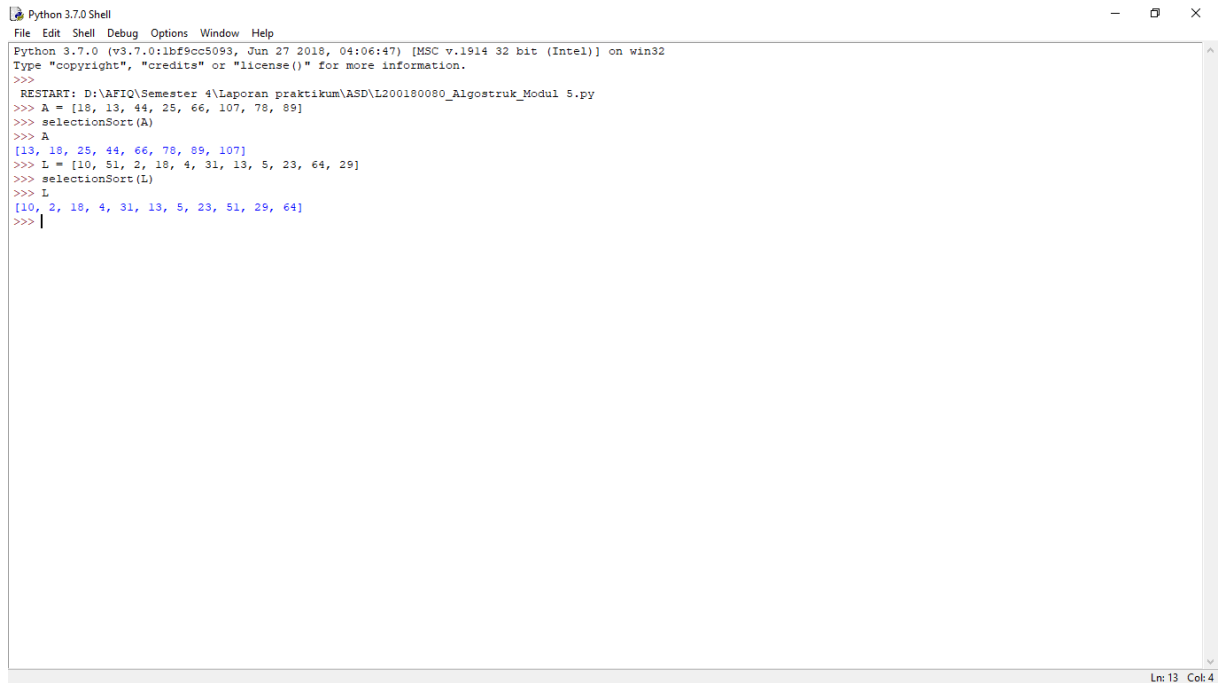
Best Case Time Complexity [Big-omega]:  $O(n)$

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

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

## Halaman 50

```
#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)
```



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\L200180080_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]
>>> |
```

## Halaman 52

```
#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]:
            A[pos] = A[pos-1]
            pos = pos - 1
        A[pos] = nilai
```

#-> cari posisi yang tepat  
# dan geser ke kanan terus  
# nilai - nilai yang lebih besar  
#-> pada posisi ini tempatkan nilai elemen ke i

Ln: 34 Col: 98

Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.

```
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\L200180080_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: 13 Col: 4