

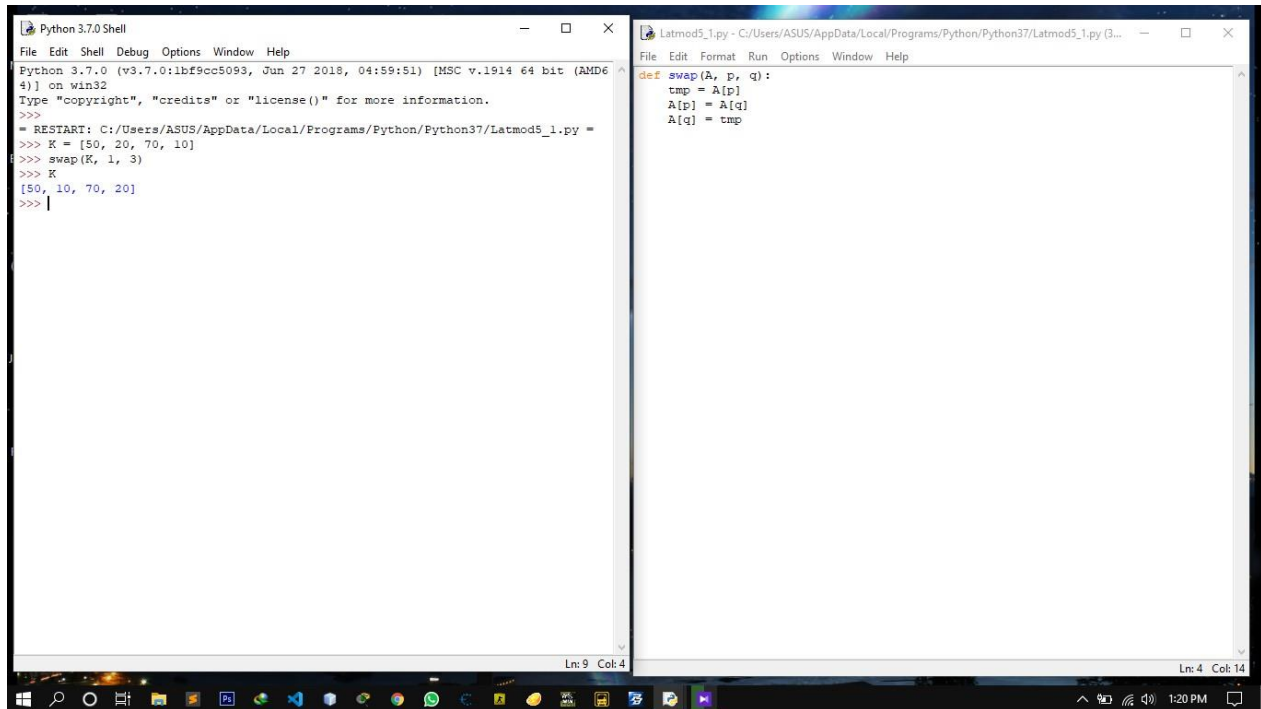
Nama : Bayu Prayitno Aji

NIM : L200180054

Kelas : C

Kegiatan Praktikum Modul 5

5.0

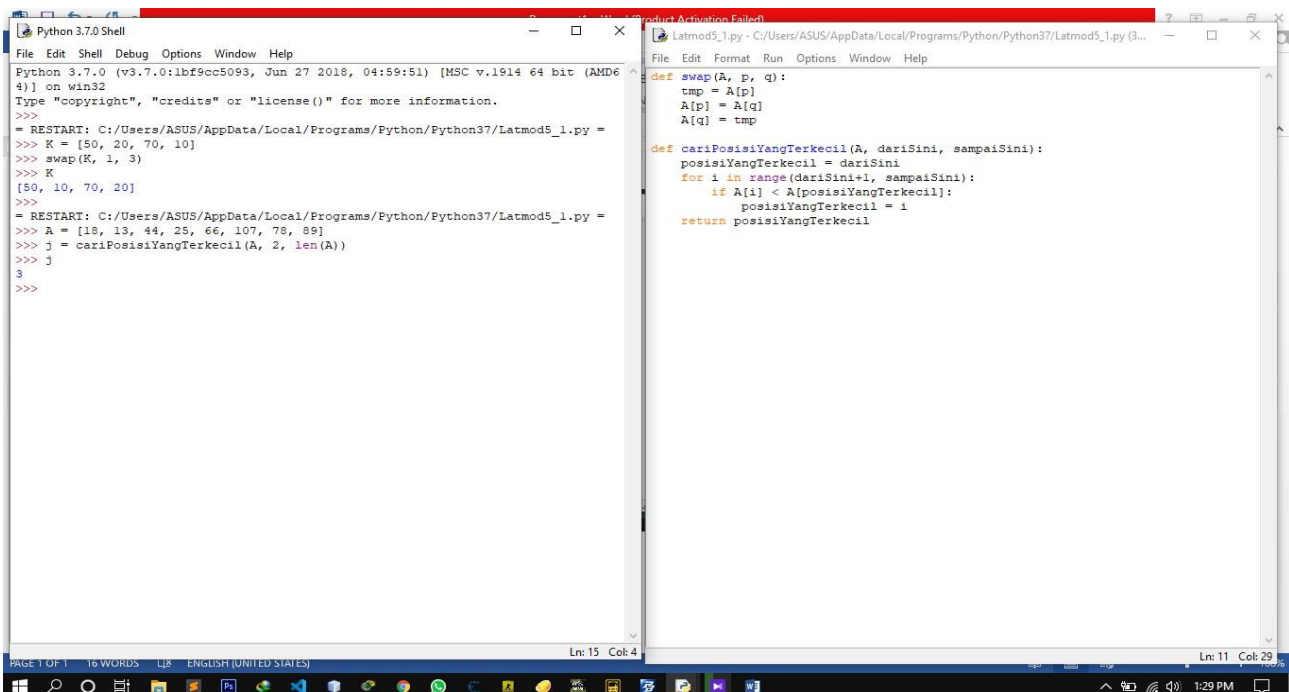


The screenshot shows two windows. The left window is a Python 3.7.0 Shell with the following code:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python37/Latmod5_1.py =
>>> K = [50, 20, 70, 10]
>>> swap(K, 1, 3)
>>> K
[50, 10, 70, 20]
>>>
```

The right window is a text editor showing the definition of the swap function:

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



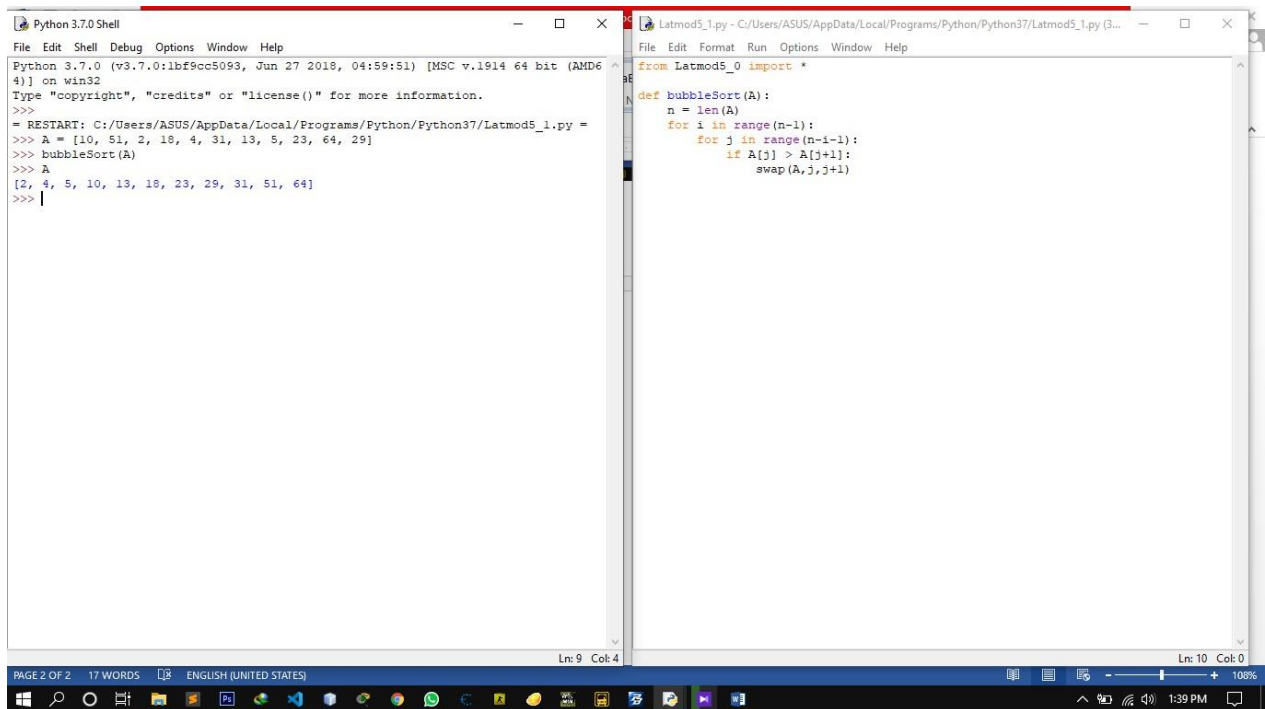
The screenshot shows two windows. The left window is a Python 3.7.0 Shell with the following code:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python37/Latmod5_1.py =
>>> K = [50, 20, 70, 10]
>>> swap(K, 1, 3)
>>> K
[50, 10, 70, 20]
>>>
= RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python37/Latmod5_1.py =
>>> A = [18, 13, 44, 25, 66, 107, 78, 89]
>>> j = cariPosisiYangTerkecil(A, 2, len(A))
>>> j
3
>>>
```

The right window is a text editor showing the definition of the cariPosisiYangTerkecil function:

```
def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1, sampaiSini):
        if A[i] < A[posisiYangTerkecil]:
            posisiYangTerkecil = i
    return posisiYangTerkecil
```

5.1 Bubble sort



The image shows two windows from a Windows operating system. The left window is titled 'Python 3.7.0 Shell' and displays the following text:

```
Python 3.7.0 (tags/v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python37/Latmod5_1.py =
>>> A = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
>>> bubbleSort(A)
>>> A
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>>
```

The right window is titled 'Latmod5_1.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python37/Latmod5_1.py (3...)' and shows the following Python code:

```
from Latmod5_0 import *

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)
```

The taskbar at the bottom shows the Windows Start button, search icon, and several application icons. The system tray on the right shows the date and time as 1:39 PM.

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 :

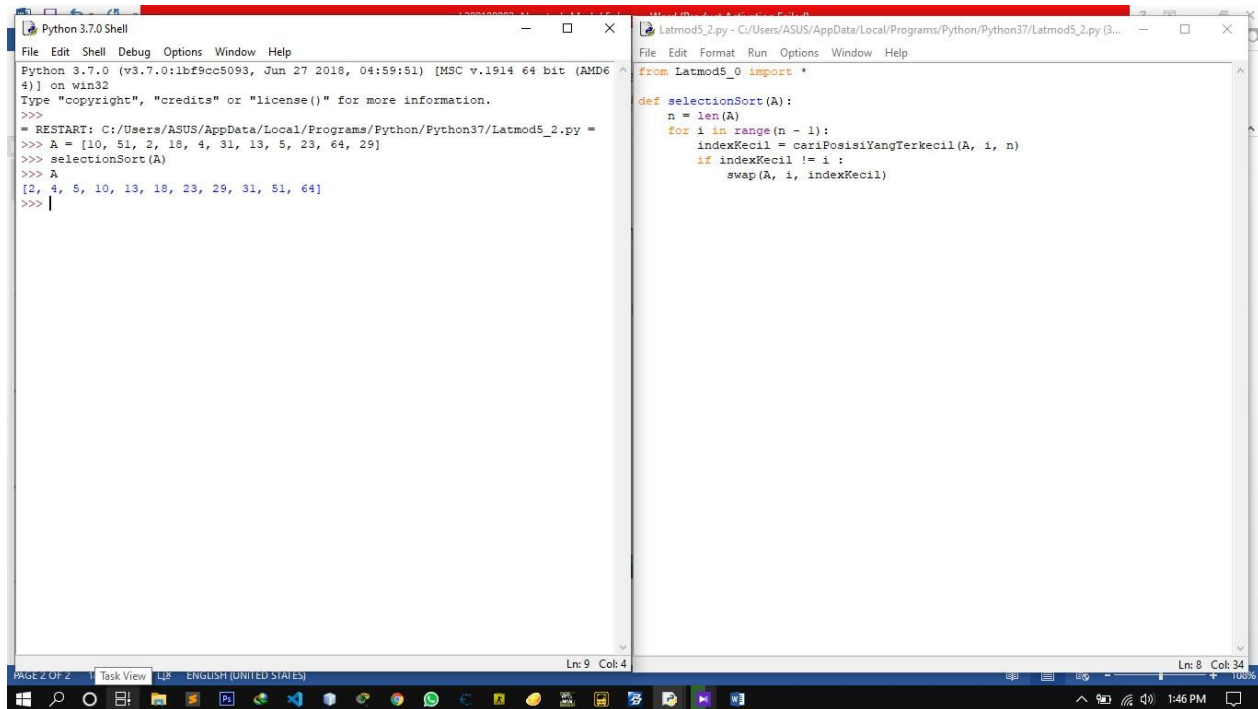
Worst Case Time Complexity [Big-O]: $O(n^2)$

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

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

Berdasarkan perhitungan dengan rumus tersebut, ada 107 perbandingan dan pertukaran.

5.2 Selection sort



The image shows a screenshot of a Windows desktop with two open applications. On the left is a 'Python 3.7.0 Shell' window, and on the right is a text editor window titled 'Latmod5_2.py'.

The Python Shell window displays the following code and output:

```
Python 3.7.0 (tags/v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python37/Latmod5_2.py =
>>> A = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
>>> selectionSort(A)
>>> A
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>>
```

The text editor window shows the implementation of the selection sort algorithm:

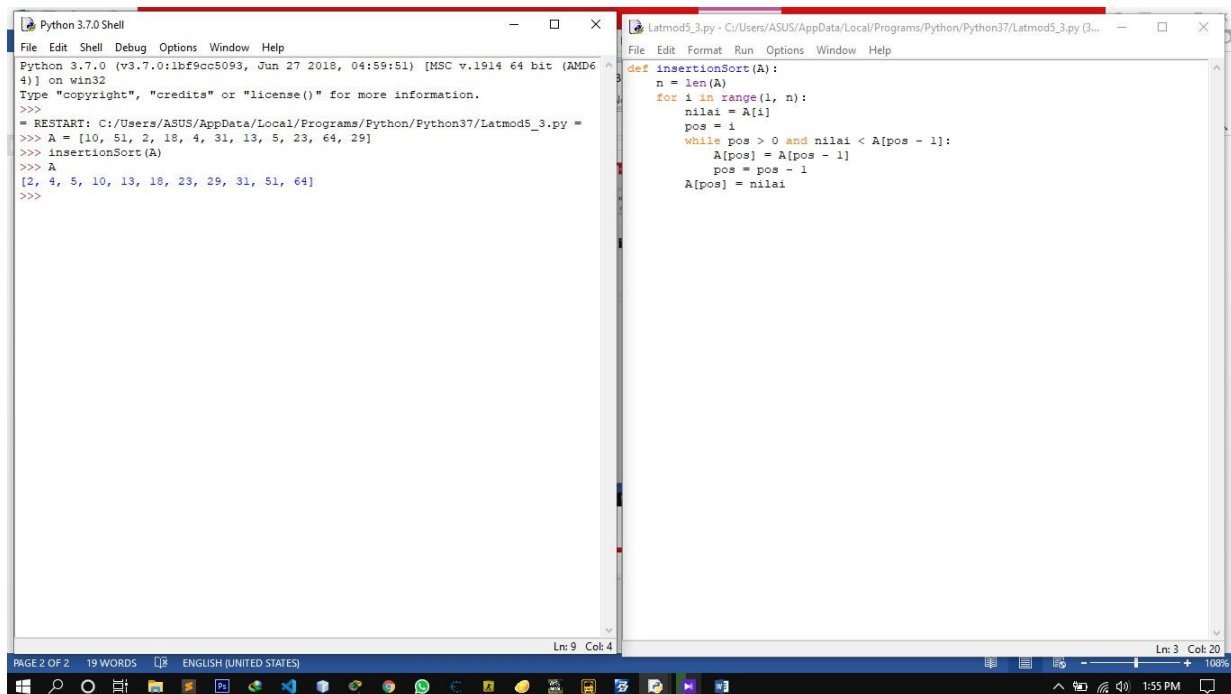
```
from Latmod5_0 import *

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

Pertanyaan : dengan elemen sebanyak n , berapa banyakkah operasi perbandingan dan pertukaran yang dilakukan oleh algoritma selection sort ini? Selidiki nilainya untuk *worst-case*, *average-case*, dan *best-case scenario*

Jawab :

5.3 Insertion sort



The image shows a Windows desktop with two application windows. The left window is titled 'Python 3.7.0 Shell' and displays the following text:

```
Python 3.7.0 (tags/v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python37/Latmod5_3.py =
>>> A = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
>>> insertionSort(A)
>>> A
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>>
```

The right window is a text editor titled 'Latmod5_3.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python37/Latmod5_3.py (3...'. It contains the following Python code for the insertion sort algorithm:

```
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
```

The taskbar at the bottom shows the Windows Start button, search icon, and several open applications including File Explorer, Photoshop, and a web browser. The system clock indicates 1:55 PM.