

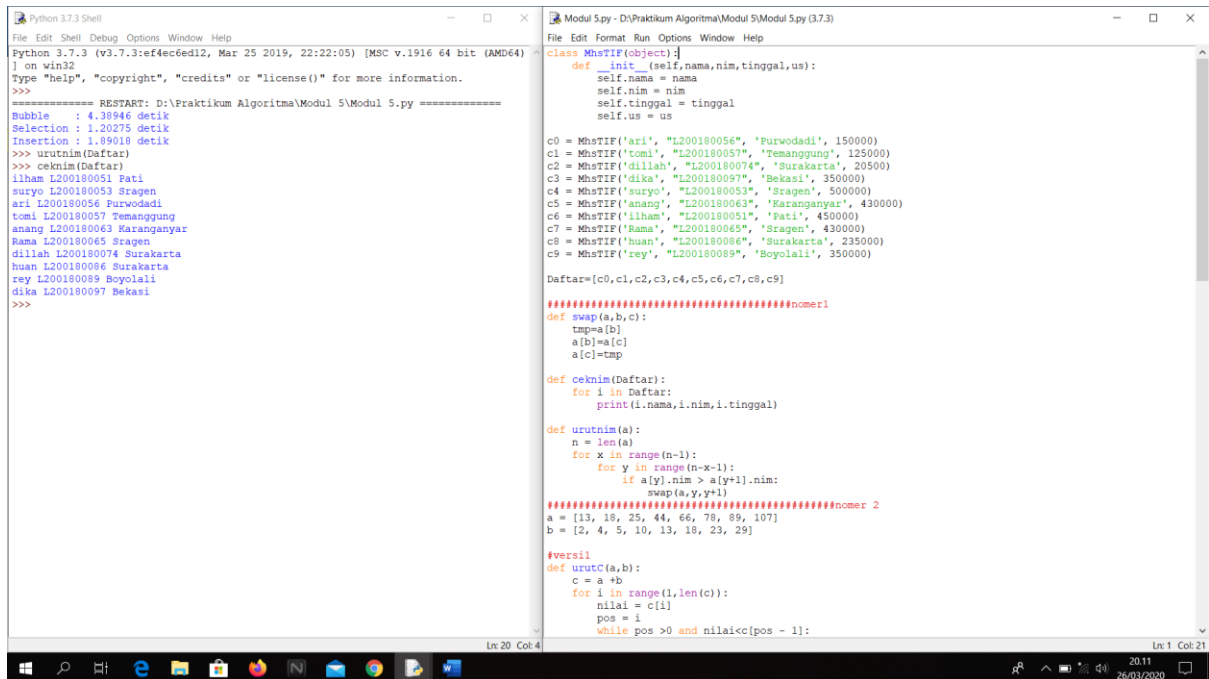
Nama : Afrizal Eka Putra

NIM : L200180075

Kelas : C

## MODUL 5

### 1. Mengurutkan daftar MhsTIF berdasarkan NIM



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Praktikum Algoritma\Modul 5\Modul 5.py =====
Bubble : 4.38946 detik
Selection : 1.20275 detik
Insertion : 1.89018 detik
>>> urutnim(Daftar)
>>> ceknim(Daftar)
ilham L200180051 Pati
surjo L200180053 Sragen
ari L200180056 Purwodadi
tomi L200180057 Temanggung
anang L200180063 Karanganyar
Rana L200180065 Sragen
dillah L200180074 Surakarta
huan L200180086 Surakarta
rey L200180089 Boyolali
dika L200180097 Bekasi
>>>
```

```
Modul 5.py - D:\Praktikum Algoritma\Modul 5\Modul 5.py (3.7.3)
File Edit Format Run Options Window Help

class MhsTIF(object):
    def __init__(self, nama, nim, tinggal, us):
        self.nama = nama
        self.nim = nim
        self.tinggal = tinggal
        self.us = us

c0 = MhsTIF('ari', "L200180056", 'Purwodadi', 150000)
c1 = MhsTIF('tomi', "L200180057", 'Temanggung', 125000)
c2 = MhsTIF('dillah', "L200180074", 'Surakarta', 20500)
c3 = MhsTIF('dika', "L200180097", 'Bekasi', 350000)
c4 = MhsTIF('surjo', "L200180053", 'Sragen', 500000)
c5 = MhsTIF('anang', "L200180063", 'Karanganyar', 430000)
c6 = MhsTIF('ilham', "L200180051", 'Pati', 450000)
c7 = MhsTIF('Rana', "L200180065", 'Sragen', 430000)
c8 = MhsTIF('huan', "L200180086", 'Surakarta', 235000)
c9 = MhsTIF('rey', "L200180089", 'Boyolali', 350000)

Daftar=[c0,c1,c2,c3,c4,c5,c6,c7,c8,c9]

#####nomer1
def swap(a,b,c):
    tmp=a[b]
    a[b]=a[c]
    a[c]=tmp

def ceknim(Daftar):
    for i in Daftar:
        print(i.nama,i.nim,i.tinggal)

def urutnim(a):
    n = len(a)
    for x in range(n-1):
        for y in range(n-x-1):
            if a[y].nim > a[y+1].nim:
                swap(a,y,y+1)

#####nomer 2
a = [13, 18, 25, 44, 66, 78, 89, 107]
b = [2, 4, 5, 10, 13, 18, 23, 29]

#versil
def urutC(a,b):
    c = a + b
    for i in range(1,len(c)):
        nilai = c[i]
        pos = i
        while pos > 0 and nilai<c[pos - 1]:
```

### 2. Penggabungan array a dan b menjadi array c dan urutkan.

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6d12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Praktikum Algoritma\Modul 5\Modul 5.py =====
Bubble : 3.92081 detik
Selection : 1.23404 detik
Insertion : 1.87456 detik
>>> urutC(a,b)
[2, 4, 5, 10, 13, 13, 18, 18, 23, 25, 29, 44, 66, 78, 89, 107]
>>> urutC(a,b)
[2, 4, 5, 10, 13, 13, 18, 18, 23, 25, 29, 44, 66, 78, 89, 107]
>>>

Modul 5.py - D:\Praktikum Algoritma\Modul 5\Modul 5.py (3.7.3)
File Edit Format Run Options Window Help
#####nomer 2
a = [13, 18, 25, 44, 66, 78, 89, 107]
b = [2, 4, 5, 10, 13, 18, 23, 29]

#versi1
def urutC(a,b):
    c = a + b
    for i in range(1,len(c)):
        nilai = c[i]
        pos = i
        while pos > 0 and nilai < c[pos - 1]:
            c[pos] = c[pos-1]
            pos = pos - 1
        c[pos] = nilai
    print(c)

#versi2
def urutC(a,b):
    pan1 = len(a)
    pan2 = len(b)
    x = 0
    y = 0
    c = []
    while x < pan1 and y < pan2:
        if a[x] < b[y]:
            c.append(a[x])
            x = x + 1
        else:
            c.append(b[y])
            y = y + 1
    while x < pan1:
        c.append(a[x])
        x = x + 1
    while y < pan2:
        c.append(b[y])
        y = y + 1
    return c

#####nomer 3
def swap(A,p,q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp

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

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)

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

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

from time import time as detik
from random import shuffle as kocok

k = [i for i in range(1,6001)]
kocok(k)
u_bub = k[:]
u_sel = k[:]
u_ins = k[:]

aw = detik();bubbleSort(u_bub);ak=detak();print("Bubble : %g detik"%(ak-aw));
```

### 3. Membandingkan lebih cepat mana bubble,selection atau insertion.

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6d12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Praktikum Algoritma\Modul 5\Modul 5.py =====
Bubble : 3.88958 detik
Selection : 1.20287 detik
Insertion : 1.7026 detik
>>> ##lebih cepat selection , menurut saya lebih cepat selection karena dia menggunakan
2 define sekaligus sehingga mempercepat perulangan daripada yang atas bubble maupun yang
bawah insertion

Modul 5.py - D:\Praktikum Algoritma\Modul 5\Modul 5.py (3.7.3)
File Edit Format Run Options Window Help
#####nomer 3
def swap(A,p,q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp

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

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)

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

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

from time import time as detik
from random import shuffle as kocok

k = [i for i in range(1,6001)]
kocok(k)
u_bub = k[:]
u_sel = k[:]
u_ins = k[:]

aw = detik();bubbleSort(u_bub);ak=detak();print("Bubble : %g detik"%(ak-aw));
```