

Nama : Chandika Aulia
NIM : L200180097
Kelas : D

Modul 5

1).Mengurutkan daftar MhsTIF berdasarkan NIM

```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\smt 4\prak algo\modul 5\modul 5.py =====
>>> urutnim(Daftar)
>>> ceknim(Daftar)
Rama L200180065 Sragen
wulan L200180091 Surakarta
Bayu L200180096 Sragen
dika L200180097 Bekasi
nayu L200180099 Lampung
rohmad L200180101 Riau
dimas L200180102 Palembang
amron L200180105 Klaten
fandit L200180118 Sragen
dhim L200180148 Sragen
>>>

modul_5.py - D:\smt 4\prak algo\modul 5\modul 5.py (3.7.6)
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('rohmad', "L200180101", 'Riau', 150000)
c1 = MhsTIF('amron', "L200180105", 'Klaten', 125000)
c2 = MhsTIF('dimas', "L200180102", 'Palembang', 20500)
c3 = MhsTIF('dika', "L200180097", 'Bekasi', 350000)
c4 = MhsTIF('Bayu', "L200180096", 'Sragen', 500000)
c5 = MhsTIF('nayu', "L200180099", 'Lampung', 430000)
c6 = MhsTIF('wulan', "L200180091", 'Surakarta', 450000)
c7 = MhsTIF('Rama', "L200180065", 'Sragen', 430000)
c8 = MhsTIF('fandit', "L200180118", 'Sragen', 235000)
c9 = MhsTIF('dhim', "L200180148", 'Sragen', 350000)

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

#1
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]

def urutC(a,b):
```

2). penggabungan array a dan b menjadi array c dan urutkan

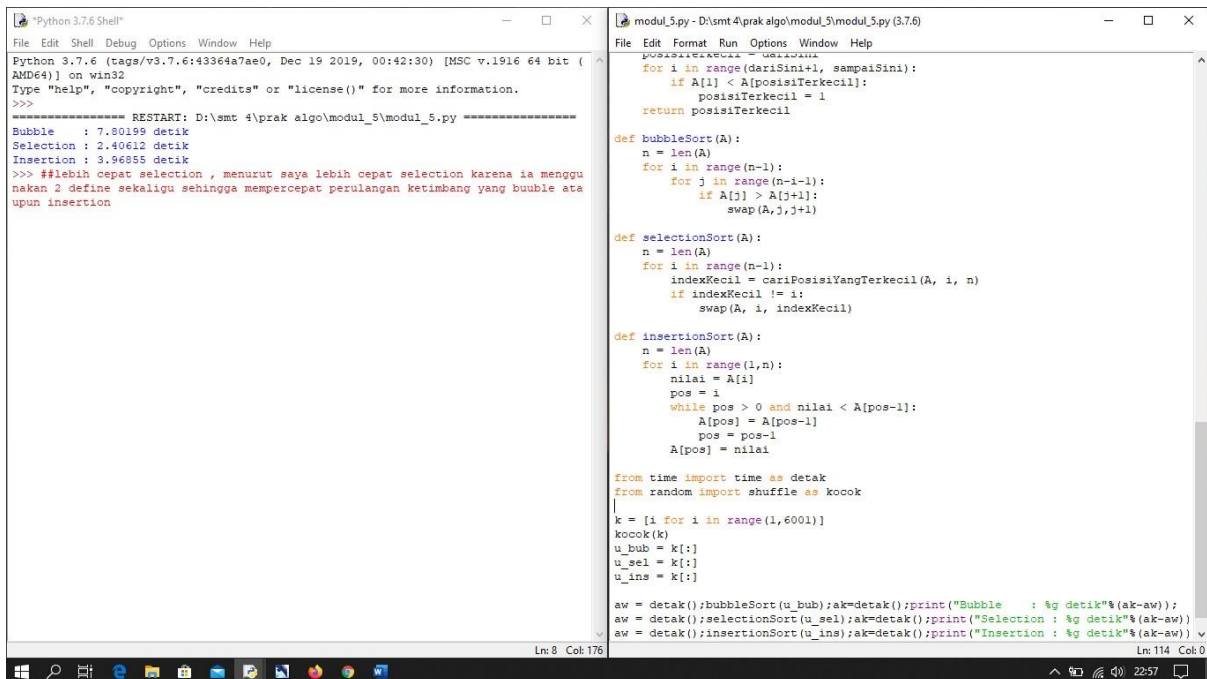
```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\smt 4\prak algo\modul 5\modul 5.py =====
>>> ##versi 1
>>> urutC(a,b)
[2, 4, 5, 10, 13, 18, 18, 23, 25, 29, 44, 66, 78, 89, 107]
>>> ##versi 2
>>> urut(a,b)
[2, 4, 5, 10, 13, 18, 18, 23, 25, 29, 44, 66, 78, 89, 107]
>>>

modul_5.py - D:\smt 4\prak algo\modul 5\modul 5.py (3.7.6)
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 -= 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+=1
        else:
            c.append(b[y])
            y+=1
    while x<pan1:
        c.append(a[x])
        x+=1
    while y<pan2:
        c.append(b[y])
        y+=1
    return c
```

3).membandingkan lebih cepat mana bubble,selection atau insertion



The image shows two side-by-side windows of a Python 3.7.6 Shell. The left window displays the output of a script that compares the execution time of three sorting algorithms: Bubble Sort, Selection Sort, and Insertion Sort. The right window shows the source code of the script, which defines functions for each algorithm and uses the 'time' module to measure their performance on a random list of 6001 integers.

```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\smt 4\prak algo\modul 5\modul 5.py =====
Bubble      : 7.80199 detik
Selection   : 2.40612 detik
Insertion    : 3.96855 detik
>>> ##lebih cepat selection , menurut saya lebih cepat selection karena ia mengu
nekan 2 define sekaligus sehingga mempercepat perulangan ketimbang yang bubble ata
upun insertion

modul_5.py - D:\smt 4\prak algo\modul 5\modul 5.py (3.7.6)
File Edit Format Run Options Window Help
posisiTerkecil = cariPosisiYangTerkecil(A, 1, n)
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));
aw = detik(); selectionSort(u_sel); ak=detak(); print("Selection : %g detik"%(ak-aw));
aw = detik(); insertionSort(u_ins); ak=detak(); print("Insertion : %g detik"%(ak-aw));
```