

Nama : Amron Akhsanul Arif
Nim : L200180105
Kelas : D

Modul 5

1). Mengurutkan daftar MhsTIF berdasarkan NIM

```
Modul_5.py - C:\Users\asus\AppData\Local\Programs\Python\Python37\Modul_5.py (3...
Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

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\Modul_5.py ==
>>> cekNIM(Daftar)
L200180105
L200180100
L200180097
L200180101
L200180079
L200180078
L200180069
L200180066
L200180088
L200180094
>>> urutNIM(Daftar)
>>> cekNIM(Daftar)
L200180066
L200180069
L200180078
L200180079
L200180084
L200180098
L200180097
L200180100
L200180101
L200180105
>>>

class MhsTIF(object):
    def __init__(self, nama, NIM, alamat, us):
        self.nama = nama
        self.NIM = NIM
        self.alamat = alamat
        self.us = us

    def __str__(self):
        s = self.nama + "NIM" + str(self.NIM)\
            + ". Tinggal di " + self.alamat\
            + ". Uang Saku Rp. " + str(self.us)\
            + " Tiap Bulannya."

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

Daftar = [MhsTIF('Amron', "L200180105", 'Surakarta', 900000),
           MhsTIF('Rayhan', "L200180100", 'Karanganyar', 1000000),
           MhsTIF('Dika', "L200180097", 'Bekasi', 800000),
           MhsTIF('Irul', "L200180101", 'Riau', 300000),
           MhsTIF('Beny', "L200180079", 'Karanganyar', 1200000),
           MhsTIF('Akbar', "L200180078", 'Madiun', 1130000),
           MhsTIF('Taufiq', "L200180069", 'Pacitan', 750000),
           MhsTIF('Annisa', "L200180066", 'Surakarta', 830000),
           MhsTIF('Aprinta', "L200180088", 'Sragen', 780000),
           MhsTIF('Sindhi', "L200180084", 'Klaten', 650000)]

def cekNIM(object):
    for i in object:
        print (i.NIM)

def urutNIM(object):
    n = len(object)
    for i in range(n-1):
        for j in range(n-i-1):
            if object[j].NIM > object[j+1].NIM:
                swap(object,j,j+1)
```

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

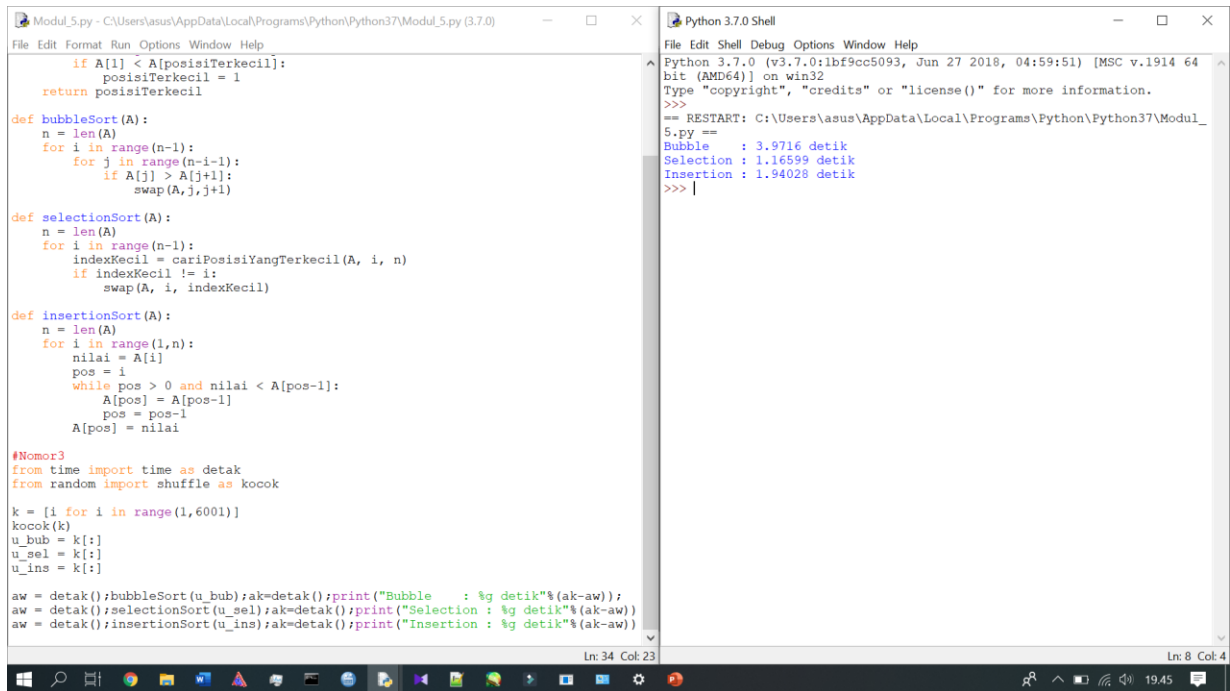
```
Modul_5.py - C:\Users\asus\AppData\Local\Programs\Python\Python37\Modul_5.py (3...
Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

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\Modul_5.py ==
>>> a = (1,3,11,15,31)
>>> b = (2,4,5,8,12)
>>> combine(a,b)
[1, 2, 3, 4, 5, 8, 11, 12, 15, 31]
>>>

def combine(A, B):
    la = len(A)
    lb = len(B)
    c = list()
    i = 0
    j = 0
    while i < la and j < lb:
        if A[i] < B[j]:
            c.append(A[i])
            i += 1
        else:
            c.append(B[j])
            j += 1
    while i < la:
        c.append(A[i])
        i += 1
    while j < lb:
        c.append(B[j])
        j += 1
    return c
```

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



The image shows a side-by-side comparison of two windows. The left window is a text editor titled 'Modul_5.py' containing Python code for three sorting algorithms: bubbleSort, selectionSort, and insertionSort. The right window is the 'Python 3.7.0 Shell' showing the execution results of these algorithms on a list of 6001 random integers. The results show that insertion sort is the fastest, followed by selection sort, and then bubble sort.

```
Modul_5.py - C:\Users\asus\AppData\Local\Programs\Python\Python37\Modul_5.py (3.7.0)
File Edit Format Run Options Window Help

    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

#Nomor3
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))

Ln: 34 Col: 23

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

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\Modul_
5.py ==
Bubble      : 3.9716 detik
Selection : 1.16599 detik
Insertion : 1.94028 detik
>>> |

Ln: 8 Col: 4
```