

Nama : Berlian Vidia Puspa
NIM : L200180107
Kelas : D

MODUL 5

PENGURUTAN

1. Buatlah suatu program untuk mengurutkan array mahasiswa berdasarkan NIM, yang elemennya terbuat dari class MhsTIF, yang telah kamu buat sebelumnya.

```
nomer1.py - C:\Users\ACER\Downloads\modul5\nomer1.py (3.7.4)
File Edit Format Run Options Window Help

import mahasiswa as mhs

h0 = mhs.mhsTIF("Berlian", 107, "Pati", 240000)
h1 = mhs.mhsTIF("Elsa", 108, "Mojolaban", 230000)
h2 = mhs.mhsTIF("Ayudhia", 95, "Surakarta", 250000)
h3 = mhs.mhsTIF("Wulan", 91, "Kartasura", 230000)
h4 = mhs.mhsTIF("Nayu", 99, "Pangkalan Bun", 240000)
h5 = mhs.mhsTIF("Tata", 61, "Pati", 250000)
h6 = mhs.mhsTIF("Irul", 101, "Riau", 245000)
h7 = mhs.mhsTIF("Caca", 97, "Banten", 245000)
h8 = mhs.mhsTIF("Diah", 106, "Sorong", 245000)
h9 = mhs.mhsTIF("Anggit", 111, "NTT", 230000)
h10 = mhs.mhsTIF("Amron", 105, "Kudus", 265000)

x = [h0,h1, h2, h3, h4, h5, h6, h7, h8, h9, h10]

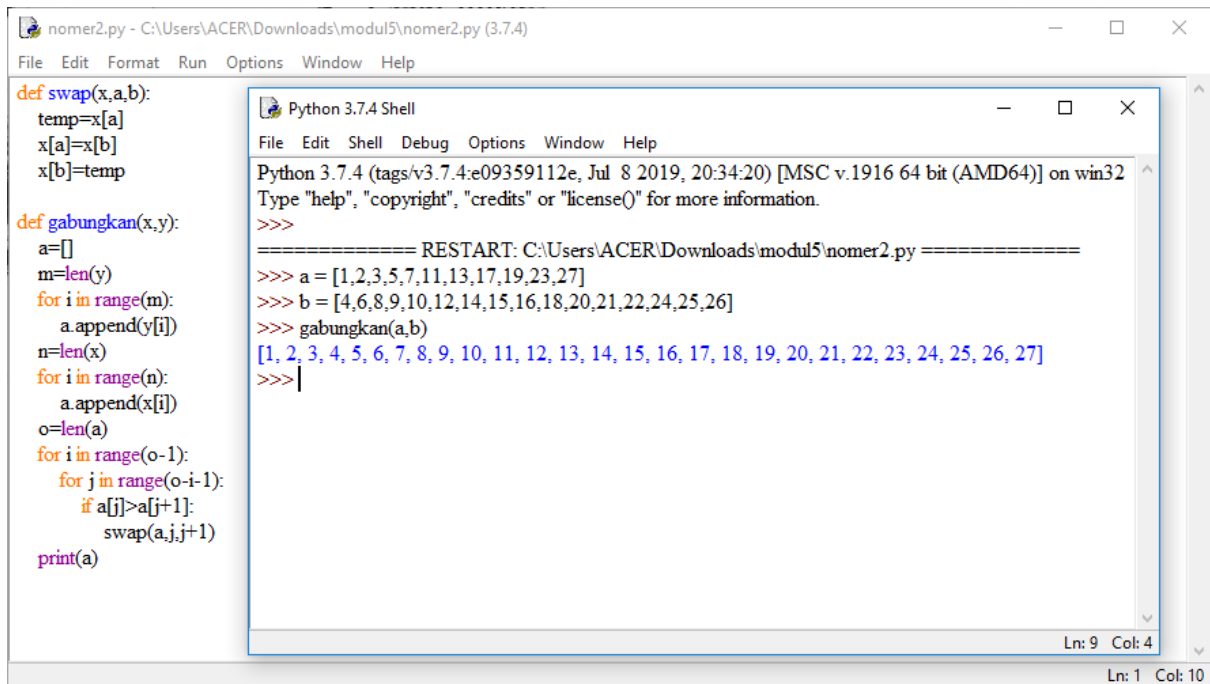
def swap(x,a,b):
    temp=x[a]
    x[a]=x[b]
    x[b]=temp

def sort(x):
    n=len(x)
    for i in range(n-1):
        for j in range(n-i-1):
            if x[j].nim>x[j+1].nim:
                swap(x,j,j+1)
    for i in x:
        print(i.nama)

sort(x)
```

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019) [AMD64]
Type "help", "copyright", "credits" or "license()"
>>>
===== RESTART: C:\Users\ACEF
Tata
Wulan
Ayudhia
Caca
Nayu
Irul
Amron
Diah
Berlian
Elsa
Anggit
>>>|
```

2. Misal terdapat dua buah array yang sudah urut A dan B. Buatlah suatu program untuk menggabungkan, secara efisien, kedua array itu menjadi suatu array C yang urut.



The image shows a Python IDE window titled 'nomer2.py - C:\Users\ACER\Downloads\modul5\nomer2.py (3.7.4)'. The code defines two functions: 'swap(x,a,b)' which swaps elements at indices a and b in array x, and 'gabungkan(x,y)' which merges two sorted arrays x and y into a new array a. The 'gabungkan' function appends elements from y to a, then appends elements from x to a, and finally sorts the array 'a' using a bubble sort algorithm. A second window titled 'Python 3.7.4 Shell' shows the execution of the script. It displays the arrays 'a = [1,2,3,5,7,11,13,17,19,23,27]' and 'b = [4,6,8,9,10,12,14,15,16,18,20,21,22,24,25,26]', followed by the call 'gabungkan(a,b)' and the resulting sorted array '[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]'. The status bar at the bottom indicates 'Ln: 9 Col: 4' and 'Ln: 1 Col: 10'.

```
def swap(x,a,b):
    temp=x[a]
    x[a]=x[b]
    x[b]=temp

def gabungkan(x,y):
    a=[]
    m=len(y)
    for i in range(m):
        a.append(y[i])
    n=len(x)
    for i in range(n):
        a.append(x[i])
    o=len(a)
    for i in range(o-1):
        for j in range(o-i-1):
            if a[j]>a[j+1]:
                swap(a,j,j+1)
    print(a)
```

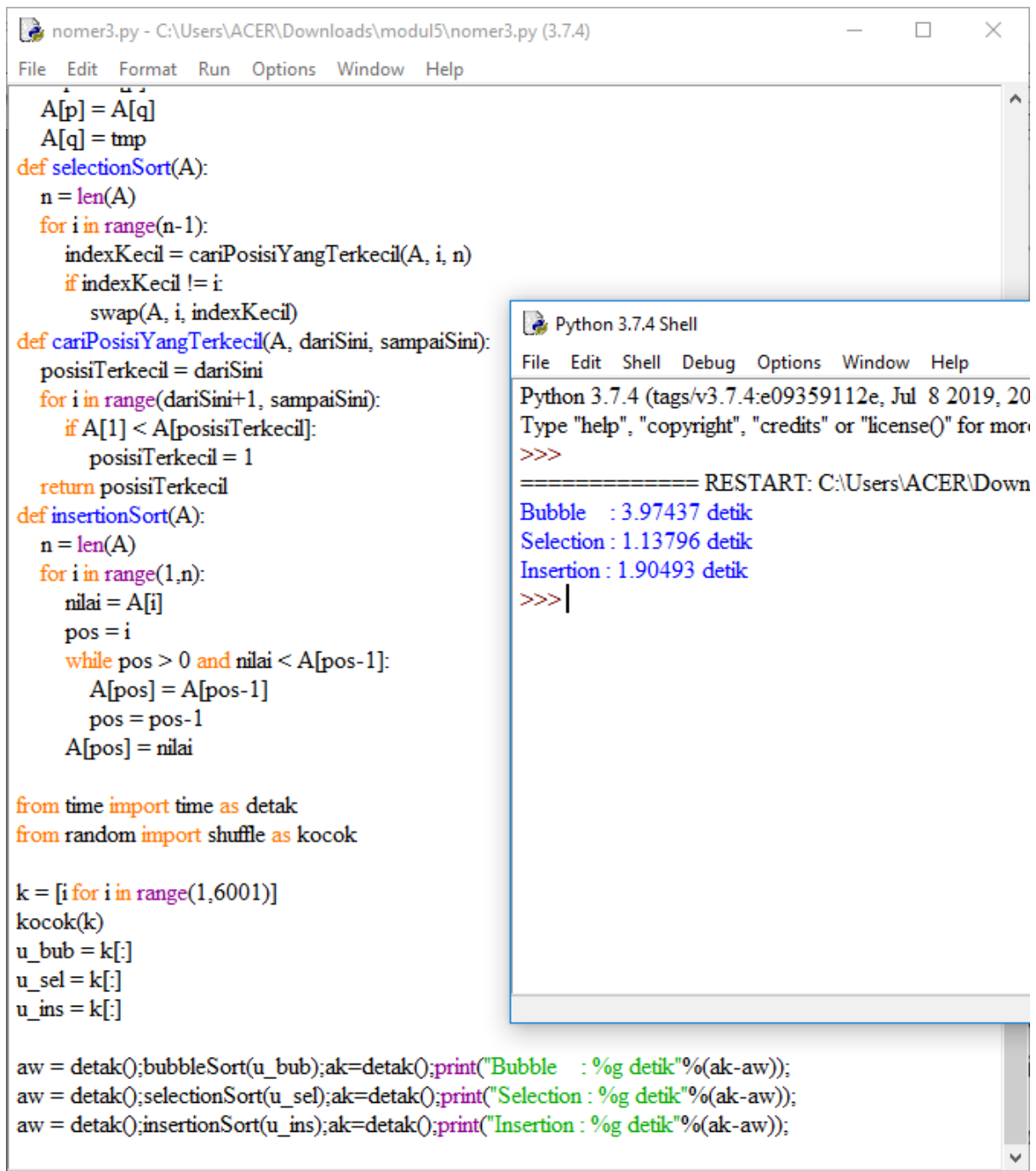
```
Python 3.7.4 Shell
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ACER\Downloads\modul5\nomer2.py =====
>>> a = [1,2,3,5,7,11,13,17,19,23,27]
>>> b = [4,6,8,9,10,12,14,15,16,18,20,21,22,24,25,26]
>>> gabungkan(a,b)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]
>>> |
```

Ln: 9 Col: 4

Ln: 1 Col: 10

3. Kamu mungkin sudah menduga, bubble sort lebih lambat dari selection sort dan juga insertion sort. Tapi manakah yang lebih cepat antara selection sort dan insertion sort?

Untuk mulai menyelidikinya, kamu bisa membandingkan waktu yang diperlukan untuk mengurutkan sebuah array yang besar, misal sepanjang 6000(enam ribu) elemen.



```
nomer3.py - C:\Users\ACER\Downloads\modul5\nomer3.py (3.7.4)
File Edit Format Run Options Window Help

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

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20
Type "help", "copyright", "credits" or "license()" for mor
>>>
===== RESTART: C:\Users\ACER\Down
Bubble : 3.97437 detik
Selection : 1.13796 detik
Insertion : 1.90493 detik
>>> |
```

Yang lebih cepat adalah selection, kemudian insertion, yang terakhir bubble.