

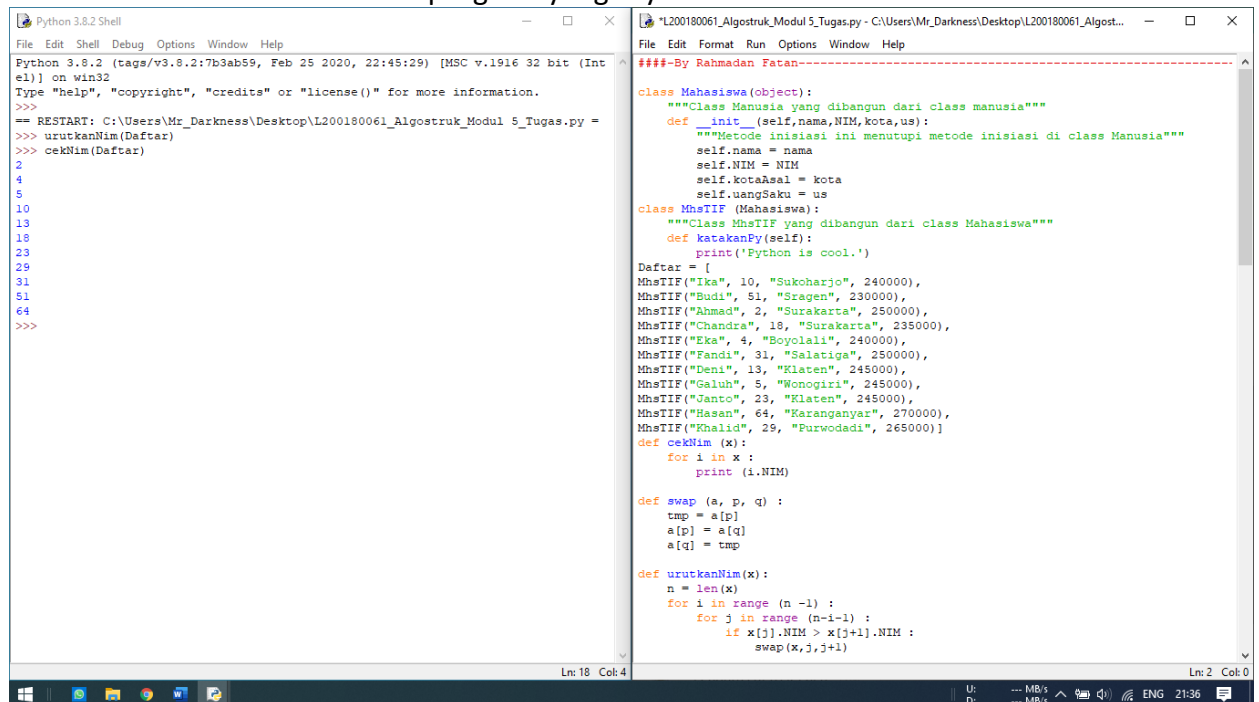
Nama : MuchFatan Rahmadan  
NIM : L200180061  
Kelas : C

## Modul 5 Pengurutan

### Tugas

1. Membuat suatu program untuk mengurutkan array mahasiswa berdasarkan NIM, yang elemennya terbuat dari class MhsTIF

Berikut adalah screenshoot dari program yang saya buat:



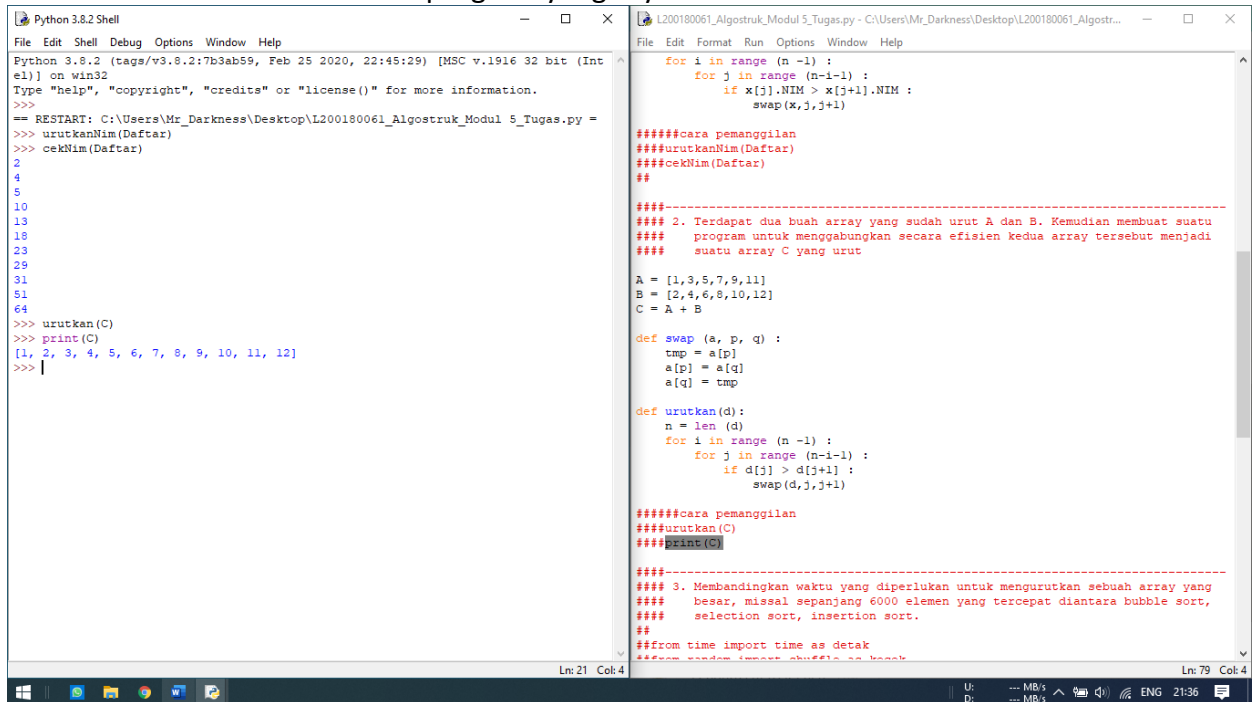
```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 5_Tugas.py =
>>> urutkanNim(Daftar)
>>> cekNim(Daftar)
2
4
5
10
13
18
23
29
31
51
64
>>>

L200180061_Algostruk_Modul 5_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 5_Tugas.py
File Edit Format Run Options Window Help
####-By Rahmadan Fatan-----
class Mahasiswa(object):
    """Class Manusia yang dibangun dari class manusia"""
    def __init__(self, nama, NIM, kota, us):
        """Metode inisiasi ini menutupi metode inisiasi di class Manusia"""
        self.nama = nama
        self.NIM = NIM
        self.kotaAsal = kota
        self.uangSaku = us
class MhsTIF (Mahasiswa):
    """Class MhsTIF yang dibangun dari class Mahasiswa"""
    def katakanPy(self):
        print('Python is cool.')
Daftar = [
    MhsTIF("Ika", 10, "Sukoharjo", 240000),
    MhsTIF("Budi", 51, "Sragen", 230000),
    MhsTIF("Ahmad", 2, "Surakarta", 250000),
    MhsTIF("Chandra", 18, "Surakarta", 235000),
    MhsTIF("Eka", 4, "Boyolali", 240000),
    MhsTIF("Fandi", 31, "Salatiga", 250000),
    MhsTIF("Deni", 13, "Klaten", 245000),
    MhsTIF("Galuh", 5, "Wonogiri", 245000),
    MhsTIF("Janto", 23, "Klaten", 245000),
    MhsTIF("Hasan", 64, "Karanganyar", 270000),
    MhsTIF("Khalid", 29, "Purwodadi", 265000)]
def cekNim (x):
    for i in x:
        print (i.NIM)
def swap (a, p, q):
    tmp = a[p]
    a[p] = a[q]
    a[q] = tmp
def urutkanNim(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)

Ln: 18 Col: 4
Ln: 2 Col: 0
```

2. Terdapat dua buah array yang sudah urut A dan B. Kemudian memmbuat suatu program untuk menggabungkan secara efisien kedua array tersebut menjadi suatu array C yang urut

Berikut adalah screenshot dari program yang saya buat:



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Int
el)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 5_Tugas.py =
>>> urutkanNim(Daftar)
>>> cekNim(Daftar)
2
4
5
10
13
18
23
29
31
51
64
>>> urutkan(C)
>>> print(C)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
>>> |

L200180061_Algostruk_Modul 5_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostr...
File Edit Format Run Options Window Help
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)

#####cara pemanggilan
#####urutkanNim(Daftar)
#####cekNim(Daftar)
##

#####-----
### 2. Terdapat dua buah array yang sudah urut A dan B. Kemudian membuat suatu
### program untuk menggabungkan secara efisien kedua array tersebut menjadi
### suatu array C yang urut

A = [1,3,5,7,9,11]
B = [2,4,6,8,10,12]
C = A + B

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

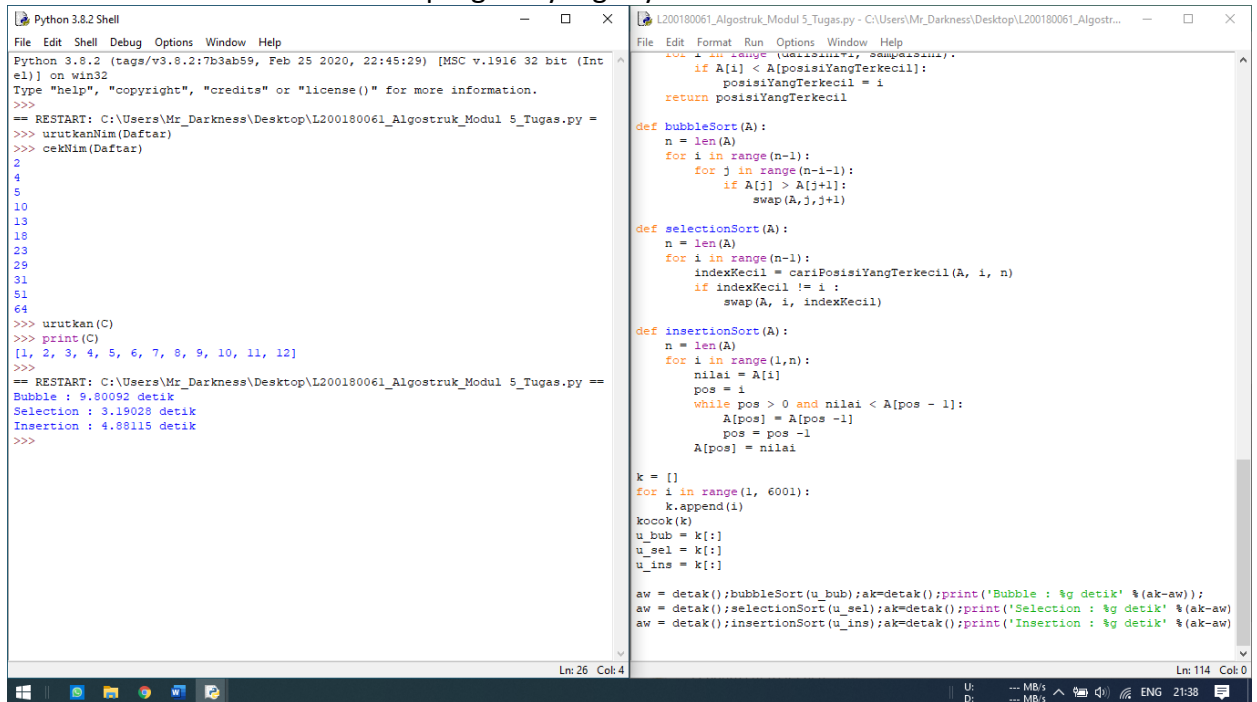
def urutkan(d):
    n = len (d)
    for i in range (n -1) :
        for j in range (n-i-1) :
            if d[j] > d[j+1] :
                swap(d,j,j+1)

#####cara pemanggilan
#####urutkan(C)
#####print(C)

#####-----
### 3. Membandingkan waktu yang diperlukan untuk mengurutkan sebuah array yang
### besar, misal sepanjang 6000 elemen yang tercepat diantara bubble sort,
### selection sort, insertion sort.
##
##from time import time as detik
##from random import shuffle asacak
```

3. Membandingkan waktu yang diperlukan untuk mengurutkan sebuah array yang besar, misal sepanjang 6000 elemen yang tercepat diantara bubble sort, selection sort, insertion sort.

Berikut adalah screenshot dari program yang saya buat:



The screenshot shows two side-by-side windows of a Python 3.8.2 Shell. The left window displays the execution of a script named 'L200180061\_Algostruk\_Modul 5\_Tugas.py'. The script defines three sorting functions: bubbleSort, selectionSort, and insertionSort. It then tests each function with a list of numbers [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] and prints the time taken for each. The results are: Bubble : 9.80092 detik, Selection : 3.19028 detik, and Insertion : 4.88115 detik. The right window shows the source code of the same script, including the definitions of the three sorting functions and the test code.

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Int
el)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 5_Tugas.py ==
>>> urutkanNim(Daftar)
>>> cekNim(Daftar)
2
4
5
10
13
18
23
29
31
51
64
>>> urutkan(C)
>>> print(C)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
>>>
== RESTART: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 5_Tugas.py ==
Bubble : 9.80092 detik
Selection : 3.19028 detik
Insertion : 4.88115 detik
>>>
```

```
L200180061_Algostruk_Modul 5_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostr...
File Edit Format Run Options Window Help
def cariPosisiYangTerkecil(A, n):
    if A[1] < A[posisiYangTerkecil]:
        posisiYangTerkecil = i
    return posisiYangTerkecil

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

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

aw = detak();bubbleSort(u_bub);ak=detak();print('Bubble : %g detik' %(ak-aw));
aw = detak();selectionSort(u_sel);ak=detak();print('Selection : %g detik' %(ak-aw));
aw = detak();insertionSort(u_ins);ak=detak();print('Insertion : %g detik' %(ak-aw))
```

Paling cepat adalah selection sort(3.1 detik), disusul insertion sort (4.8 detik) dan terakhir bubble sort (9.8 detik)