

Nama : Afiq Tri Nugraha

NIM : L200180080

Kelas : C

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

MODUL 5

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

Jawab:

```
L200180080_Algostruk_Modul_5_Tugas.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\modul 5\L200180080_Algostruk_Modul_5_Tugas.py (3.7.0)
File Edit Format Run Options Window Help

class Mahasiswa(object):
    """Class Mahasiswa yang menyimpan data-data mahasiswa"""
    def __init__(self,nama,NIM,kota,us):
        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("Abel", 23, "Purwosari", 265000),
    MhsTIF("Novi", 15, "Loram", 255000),
    MhsTIF("Intan", 26, "Mlati", 260000),
    MhsTIF("Afiq", 7, "Kudus", 280000),
    MhsTIF("Riyo", 69, "Mayong", 270000),
    MhsTIF("Naufal", 56, "Getas", 270000),
    MhsTIF("Ilham", 33, "Prambatan", 250000),
    MhsTIF("Aini", 3, "Jati", 260000),
    MhsTIF("Fara", 76, "Gebog", 250000),
    MhsTIF("Ulfa", 88, "Jati", 265000),
    MhsTIF("Riko", 100, "Demak", 300000)]

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

###cara pemanggilan
#urutkanNIM(Daftar)
#cekNIM(Daftar)
```

class Mahasiswa(object):

"""Class Mahasiswa yang menyimpan data-data mahasiswa"""

def __init__(self,nama,NIM,kota,us):

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("Abel", 23, "Purwosari", 265000),
    MhsTIF("Novi", 15, "Loram", 255000),
    MhsTIF("Intan", 26, "Mlati", 260000),
    MhsTIF("Afiq", 7, "Kudus", 280000),
    MhsTIF("Riyo", 69, "Mayong", 270000),
    MhsTIF("Naufal", 56, "Getas", 270000),
    MhsTIF("Ilham", 33, "Prambatan", 250000),
    MhsTIF("Aini", 3, "Jati", 260000),
    MhsTIF("Fara", 76, "Gebog", 250000),
    MhsTIF("Ulfa", 88, "Jati", 265000),
    MhsTIF("Riko", 100, "Demak", 300000)]

```

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

```

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)
```

```
####cara pemanggilan
```

```
##urutkanNIM(Daftar)
```

```
##cekNIM(Daftar)
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\modul 5\L200180080_Algostruk_Modul 5_Tugas.py
>>> urutkanNIM(Daftar)
>>> cekNIM(Daftar)
3
7
15
23
26
33
56
69
76
88
100
>>>
```

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.

Jawab:

```
"L200180080_Algostruk_Modul 5_Tugas.py" - D:\AFIQ\Semester 4\Laporan praktikum\ASD\modul 5\L200180080_Algostruk_Modul 5_Tugas.py (3.7.0)*
File Edit Format Run Options Window Help
## 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 = [2,4,6,8,10,12]
B = [1,3,5,7,9,11]
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)
```

```
## 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 = [2,4,6,8,10,12]
```

```
B = [1,3,5,7,9,11]
```

```
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)
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\modul 5\L200180080_Algostruk_Modul 5_Tugas.py
>>> urutkan(C)
>>> print(C)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
>>>
```

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

Jawab:

L200180080_Algostruk_Modul 5_Tugas.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\modul 5\L200180080_Algostruk_Modul 5_Tugas.py (3.7.0)

File Edit Format Run Options Window Help

```
## 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 as kocok
```

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

```
def cariPosisiYangTerkecil(A,darisini, sampaisini):
    posisiYangTerkecil = darisini
    for i in range (darisini+1, sampaisini):
        if A[i] < 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 = []
```

```
k = []
for i in range(1, 6001):
    k.append(i)
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));
```

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 as kocok

def swap(A,p,q):

tmp = A[p]

A[p]= A[q]

A[q]= tmp

```

def cariPosisiYangTerkecil(A,darisini, sampaisini):
    posisiYangTerkecil = darisini
    for i in range (darisini+1, sampaisini):
        if A[i] < 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)

```

```
kocok(k)
```

```
u_bub = k[:]
```

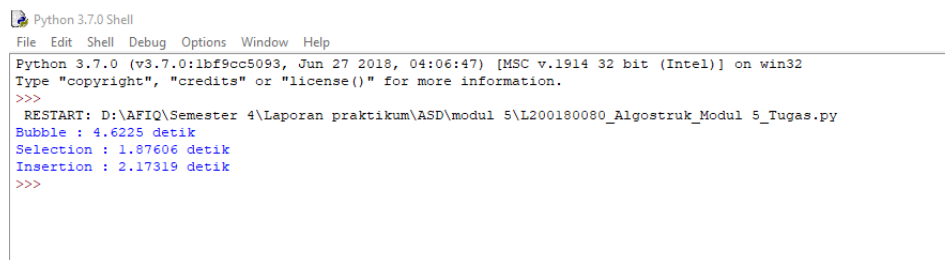
```
u_sel = k[:]
```

```
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));
```



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\modul 5\L200180080_Algostruk_Modul 5_Tugas.py
Bubble : 4.6225 detik
Selection : 1.87606 detik
Insertion : 2.17319 detik
>>>
```

Paling cepat adalah selection sort (1.9 detik), kemudian insertion sort (2.2 detik), dan terakhir bubble sort (4.6 detik).