

Nama : Anang Prasetyo

NIM : L200180063

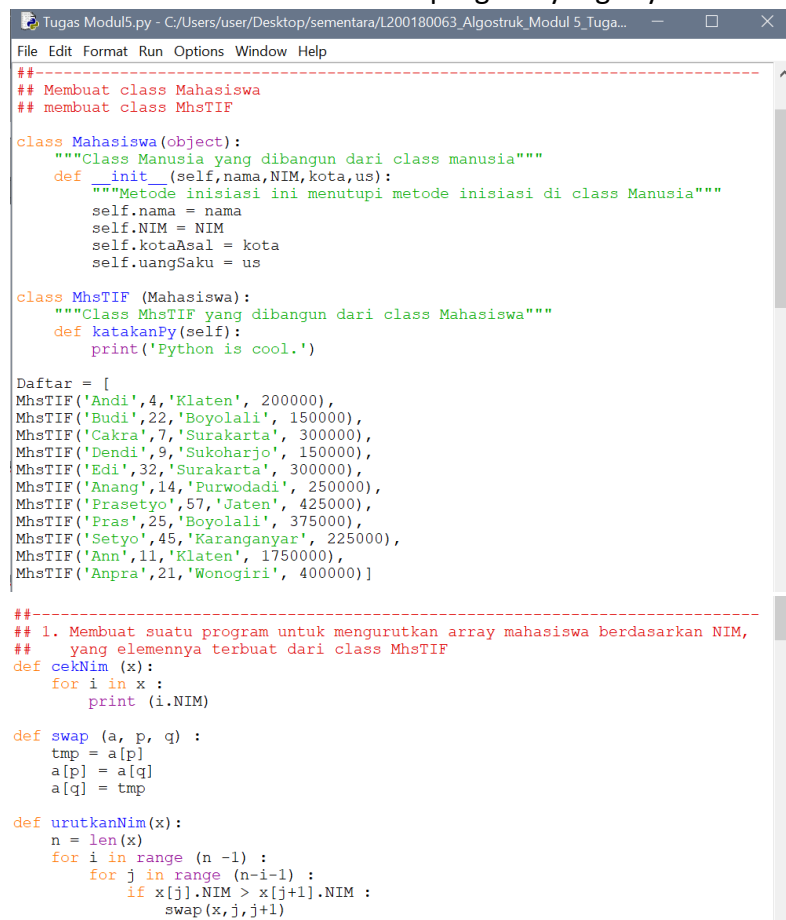
Kelas : C

Modul 5 - Pengurutan

Tugas

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

Berikut adalah screenshot dari program yang saya buat:



```
##-----  
## Membuat class Mahasiswa  
## membuat class MhsTIF  
  
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('Andi',4,'Klaten', 200000),
MhsTIF('Budi',22,'Boyolali', 150000),
MhsTIF('Cakra',7,'Surakarta', 300000),
MhsTIF('Dendi',9,'Sukoharjo', 150000),
MhsTIF('Edi',32,'Surakarta', 300000),
MhsTIF('Anang',14,'Purwodadi', 250000),
MhsTIF('Prasetyo',57,'Jaten', 425000),
MhsTIF('Pras',25,'Boyolali', 375000),
MhsTIF('Setyo',45,'Karanganyar', 225000),
MhsTIF('Ann',11,'Klaten', 1750000),
MhsTIF('Anpra',21,'Wonogiri', 400000)]

```
##-----  
## 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)
```

Berikut adalah screenshot saat program dijalankan:

```
...
= RESTART: C:/Users/user/Desktop/semntara/L200180063
ul 5_Tugas/Tugas Modul5.py
>>> urutkanNim(Daftar)
>>> cekNim(Daftar)
4
7
9
11
14
21
22
25
32
45
57
>>>
```

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:

```
#####
## 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)

urutkan(C)
print(C)
```

Berikut adalah screenshot saat program dijalankan:

```
= RESTART: C:/Users/user/Desktop/semntara/L200180063_Algostruk_Mod
ul 5_Tugas/Tugas Modul5.py
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
>>> |
```

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

Berikut adalah screenshot dari program yang saya buat:

```
Tugas Modul5.py - C:/Users/user/Desktop/semntara/L200180063_Algostruk_Modul 5_Tuga...
File Edit Format Run Options Window Help
##
## 3. Membandingkan waktu yang diperlukan untuk mengurutkan sebuah array yang
##     besar, missal 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 = 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));
```

Berikut adalah screenshoot saat program dijalankan:

```
= RESTART: C:/Users/user/Desktop/semntara/L200180063_Algostruk_Mod
ul 5_Tugas/Tugas Modul5.py
Bubble : 3.53169 detik
Selection : 1.00009 detik
Insertion : 1.70342 detik
>>> |
```

Paling cepat adalah selection sort(2.1 detik), disusul insertion sort (2.7 detik) dan terakhir bubble sort (6.0 detik)