

Nama : Annisa Nugraheni

NIM : L200180066

Kelas : C

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

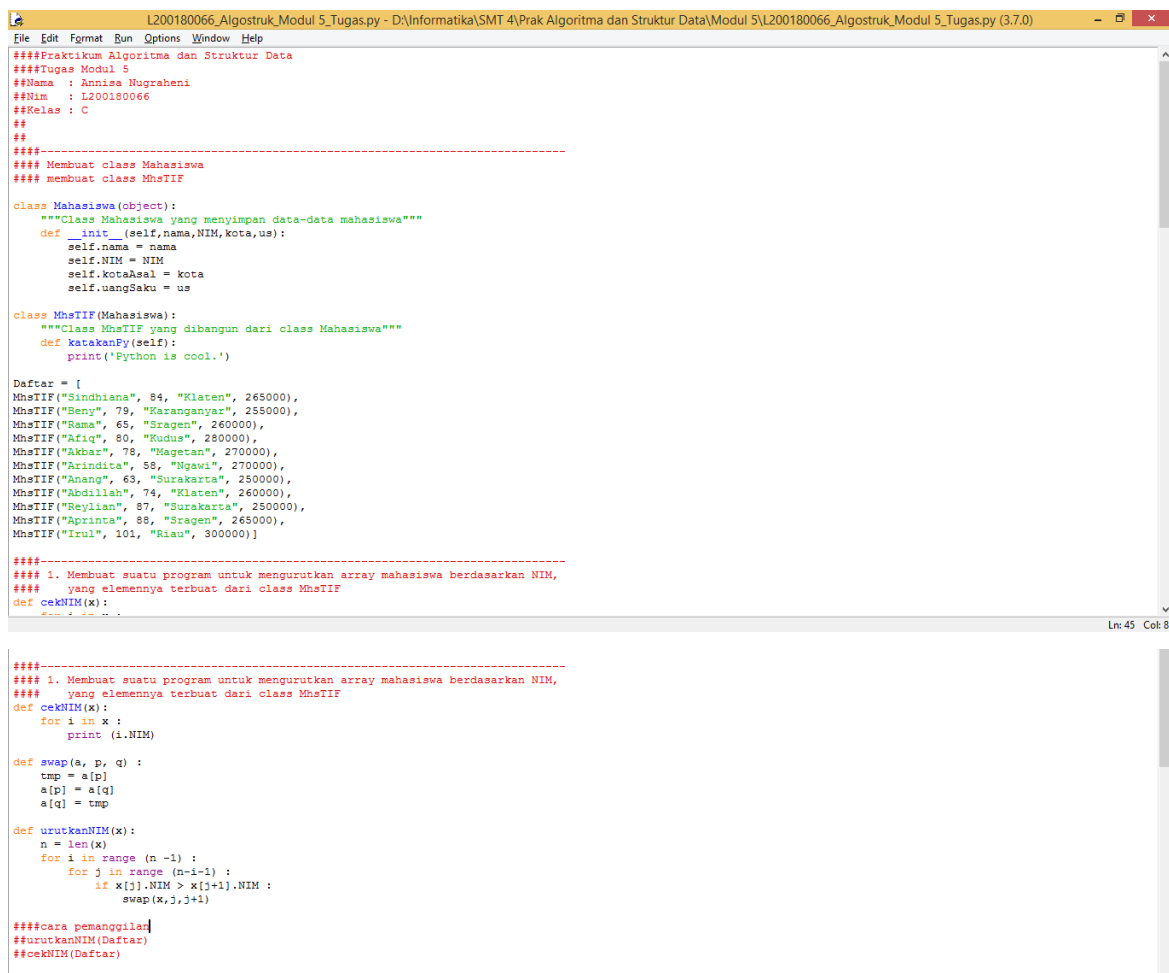
MODUL 5

PENGURUTAN

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

Jawab:

Berikut adalah screenshot dari program yang saya buat:



```
#####Praktikum Algoritma dan Struktur Data
#####Tugas Modul 5
##Nama : Annisa Nugraheni
##Nim : L200180066
##Kelas : C
##
##
#####-----
#### Membuat class Mahasiswa
#### membuat class MhsTIF

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("Sindhiana", 84, "Klaten", 265000),
    MhsTIF("Beny", 79, "Karanganyar", 255000),
    MhsTIF("Rama", 65, "Sragen", 260000),
    MhsTIF("Afiq", 80, "Kudus", 280000),
    MhsTIF("Akbar", 78, "Magetan", 270000),
    MhsTIF("Arindita", 58, "Ngawi", 270000),
    MhsTIF("Anang", 63, "Surakarta", 250000),
    MhsTIF("Abdillah", 74, "Klaten", 260000),
    MhsTIF("Reyllan", 87, "Surakarta", 250000),
    MhsTIF("Aprinta", 88, "Sragen", 265000),
    MhsTIF("Izul", 101, "Riau", 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)
```

Berikut adalah program yang saya buat:

Membuat class Mahasiswa

membuat class MhsTIF

```

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("Sindhiana", 84, "Klaten", 265000),
MhsTIF("Beny", 79, "Karanganyar", 255000),
MhsTIF("Rama", 65, "Sragen", 260000),
MhsTIF("Afiq", 80, "Kudus", 280000),
MhsTIF("Akbar", 78, "Magetan", 270000),
MhsTIF("Arindita", 58, "Ngawi", 270000),
MhsTIF("Anang", 63, "Surakarta", 250000),
MhsTIF("Abdillah", 74, "Klaten", 260000),
MhsTIF("Reylian", 87, "Surakarta", 250000),
MhsTIF("Aprinta", 88, "Sragen", 265000),
MhsTIF("Irul", 101, "Riau", 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 :
```

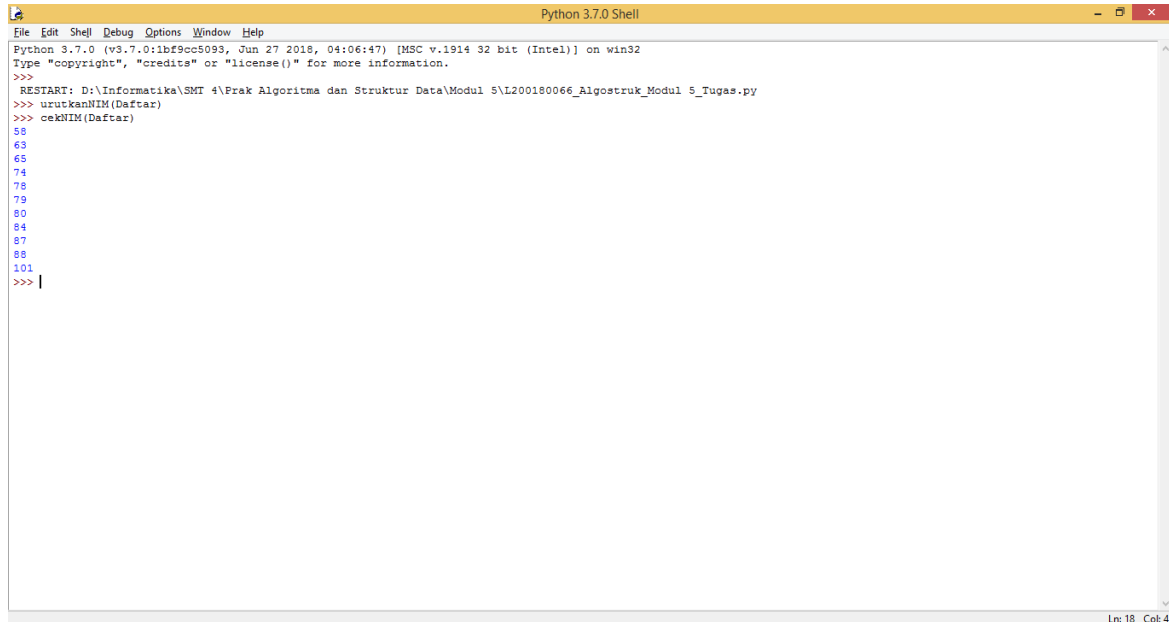
```
                swap(x,j,j+1)
```

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

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

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

Berikut adalah screenshot hasil ketika program dijalankan:



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/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:\Informatika\SMI 4\Prak Algoritma dan Struktur Data\Modul 5\L200180066_Algostruk_Modul 5_Tugas.py
>>> urutkanNIM(Daftar)
>>> cekNIM(Daftar)
58
63
65
74
78
79
80
84
87
88
101
>>> |
```

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:

Berikut adalah screenshot 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)

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

Berikut adalah 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)

#####cara pemanggilan

##urutkan(C)

```
##print(C)
```

Berikut adalah screenshot hasil ketika program dijalankan:

```
\\RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 5\L200180066_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:

Berikut adalah screenshot dari program yang saya buat:



```
L200180066_Algostruk_Modul 5_Tugas.py - D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 5\L200180066_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]
            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 program yang saya buat:

```
#### 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
```

```

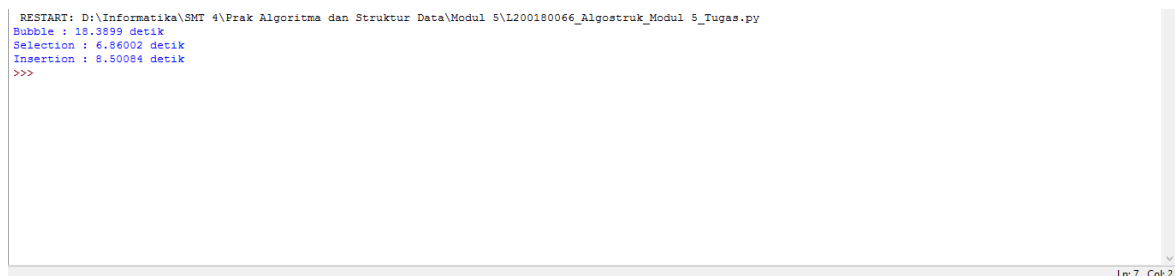
while pos > 0 and nilai < A[pos - 1]:
    A[pos] = A[pos - 1]
    pos = pos - 1
A[pos] = nilai

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));

```

Berikut adalah screenshot dari hasil ketika program dijalankan:



```

RESTART: D:\Informatika\SMT 4\Prak Algoritma dan Struktur Data\Modul 5\L200180066_Algostruk_Modul 5_Tugas.py
Bubble : 18.3899 detik
Selection : 6.86002 detik
Insertion : 8.50084 detik
>>>

```

Paling cepat adalah selection sort (6.8 detik), kemudian insertion sort (8.5 detik), dan terakhir bubble sort (18.3 detik).