

Nama : Nur Taufiq Hidayat  
NIM : L200180069  
Kelas : B

## TUGAS MODUL 5

### PENGURUTAN

#### 5.4 Soal-soal untuk Mahasiswa

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

```
class MhsTIF(object):  
    def __init__(self, nama, NIM, kota, us):  
        self.nama = nama  
        self.NIM = NIM  
        self.kotaTinggal = kota  
        self.uangSaku = us  
    def ambilNama(self):  
        return self.nama  
    def ambilNIM(self):  
        return self.NIM  
    def ambilKota(self):  
        return self.kota  
    def ambilUangSaku(self):  
        return self.uangSaku  
  
c0 = MhsTIF('Ika', 10, 'Sukoharjo', 240000)  
c1 = MhsTIF('Budi', 51, 'Sragen', 230000)  
c2 = MhsTIF('Ahmad', 2, 'Surakarta', 250000)  
c3 = MhsTIF('Chandra', 18, 'Surakarta', 235000)  
c4 = MhsTIF('Eka', 4, 'Boyolali', 240000)  
c5 = MhsTIF('Fandi', 31, 'Salatiga', 250000)  
c6 = MhsTIF('Deni', 13, 'Klaten', 240000)  
c7 = MhsTIF('Galuh', 5, 'Wonogiri', 245000)  
c8 = MhsTIF('Janto', 23, 'Klaten', 245000)  
c9 = MhsTIF('Hasan', 64, 'Karanganyar', 270000)  
c10 = MhsTIF('Khalid', 29, 'Purwodadi', 265000)  
  
Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]
```

Dengan program sebagai berikut :

```
def swap(A, p, q):  
    tmp = A[p]  
    A[p] = A[q]  
    A[q] = tmp  
  
def nim(daftar):  
    for i in daftar:  
        print(i.NIM)  
  
def urutNim(daftar):  
    n = len(daftar)  
    for i in range (n-1):  
        for j in range(n-i-1):  
            if daftar[j].NIM > daftar[j+1].NIM:  
                swap(daftar,j,j+1)
```

Saat dijalankan di python shell, hasilnya :

```
>>> urutNim(Daftar)  
>>> print(nim(Daftar))  
2  
4  
5  
10  
13  
18  
23  
29  
31  
51  
64  
None
```

2. Membuat suatu program untuk menggabungkan dua buah array yang sudah urut A dan B, menjadi suatu array C yang urut.

```
#Nomer2
A = [1,3,5,7,8,10,13,15]
B = [2,4,6,9,11,12,14]

def gabung(A, B):
    la = len(A)
    lb = len(B)
    c = list()
    i = 0
    j = 0
    while i < la and j < lb:
        if A[i] < B[j]:
            c.append(A[i])
            i += 1
        else:
            c.append(B[j])
            j += 1
    while i < la:
        c.append(A[i])
        i += 1
    while j < lb:
        c.append(B[j])
        j += 1
    return c
```

Saat dijalankan di python shell hasilnya :

```
>>> C = gabung(A, B)
>>> print(C)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

3. Membandingkan waktu yang diperlukan untuk mengurutkan array besar (misal 6000 elemen) antara bubble sort, selection sort, dan insertion sort.

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

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

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

```

Dengan memakai program berikut :

```

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));

```

Hasil yang didapat adalah :

```

Bubble      : 6.31834 detik
Selection  : 2.23714 detik
Insertion  : 3.92466 detik

```