

# Algorithm and Structure Data



Name : *Aji Tirto Prayogo/L200164008*

**Faculty Communication dan Informatics  
2018**



# ASD\_MODULKE5\_X\_L200164008

## Input

```
practice5.py - G:\menyangkut tugas\ALL TUGAS\SEMSTER 4\pratikum algoritma struktur data\Pratikum5\practice5.py (2.7.12)
File Edit Format Run Options Window Help

class mhsTIF():
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.us = us

c0 = mhsTIF('ika', 10, 'sukoharjo', 240000)
c1 = mhsTIF('budi', 51, 'seragen', 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', 245000)
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 = [c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]
|
A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 30, 31, 32]
B = [2, 3, 5, 10, 20, 22, 28, 29, 30, 33, 40]
C = [2, 1, 4, 10, 5, 4, 8, 9, 3, 7, 6]

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)
            else:
                continue
    return A
```

Ln: 21 Col: 0

```
practice5.py - G:\menyangkut tugas\ALL TUGAS\SEMSTER 4\pratikum algoritma struktur data\Pratikum5\practice5.py (2.7.12)
File Edit Format Run Options Window Help

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)
            else:
                continue
    return A

def insertionSort(A):
    n = len(A)
    for i in range(n-1):
        nilai = A[i]
        pos = i
        while pos > 0 and nilai < A[pos-1]:
            A[pos] = A[pos-1]
            pos = pos - 1
        A[pos] = nilai

##insertion(daftar)
##for i in daftar:
##    print i.nim

def selectionSort(A):
    n = len(A)
    for i in range(n-1):
        indexKecil = cariPosisiYangTerkecil(A, i, n)
        if indexKecil != i:
            swap(A, i, indexKecil)
    return A

def printArray(A):
    for i in A:
        print(i.nama + ' ' + str(i.nim) + ' ' + i.kota + ' ' + str(i.us))

def cariArrayYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1, sampaiSini):
        if A[i].NIM < A[posisiYangTerkecil].NIM:
            posisiYangTerkecil = i
    return posisiYangTerkecil
```

Ln: 28 Col: 15

# ASD\_MODULKE5\_X\_L200164008

```
*practice5.py - G:\menyangkut tugas\ALL TUGAS\SEMSTER 4\pratikum algoritma struktur data\Pratikum5\practice5.py (2.7.12)*
File Edit Format Run Options Window Help

def cariArrayYangTerkecil(A,dariSini,sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1,sampaiSini):
        if A[i].NIM < A[posisiYangTerkecil].NIM:
            posisiYangTerkecil = i
    return posisiYangTerkecil

def sortArray(A):
    n=len(A)
    for i in range(n-1):
        indexKecil = cariArrayYangTerkecil(A,i,n)
        if indexKecil != i :
            swap(A,i,indexKecil)
    return A

def ArrayJoiner(A,B):
    mergedList=A+B
    mergedList=selectionSort(mergedList)
    return mergedList

print(A);print(B)

from time import time
from random import shuffle

k = range(1,6001)
shuffle(k)
u_bub=k[:]
u_sel=k[:]
u_ins=k[:]

aw=time();bubbleSort(u_bub);ak=time();print("bubble: %g detik"%(ak-aw));
aw=time();selectionSort(u_sel);ak=time();print("selection: %g detik"%(ak-aw));
aw=time();insertionSort(u_ins);ak=time();print("insertion: %g detik"%(ak-aw));

Activate Windows
Go to Settings to activate Windows.

Ln: 100 Col: 0
```

## Output

```
>>> sortArray(daftar)
[<_main_.mhsTIF instance at 0x03094B98>, <_main_.mhsTIF instance at 0x03B74620>, <_main_.mhsTIF instance at 0x03B77C10>, <_main_.mhsTIF instance at 0x03B778F0>,
 <_main_.mhsTIF instance at 0x03B74530>, <_main_.mhsTIF instance at 0x03B77AF8>, <_main_.mhsTIF instance at 0x03B779B8>, <_main_.mhsTIF instance at 0x03B74238>,
 <_main_.mhsTIF instance at 0x03094C60>, <_main_.mhsTIF instance at 0x03B77B48>]
>>> printArray(daftar)
ahmad 2 surakarta 250000
eka 4 boyolali 240000
galuh 5 wonogiri 245000
deni 13 klaten 245000
chandra 18 surakarta 235000
janto 23 klaten 245000
khalid 29 purwodadi 265000
fandi 31 selatiga 250000
budi 51 sragen 230000
hasan 64 karanganyar 270000
>>>
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 30, 31, 32]
[2, 3, 5, 10, 20, 22, 28, 29, 30, 33, 40]
>>> ArrayJoiner(A,B)
[1, 2, 2, 3, 3, 4, 5, 5, 6, 7, 8, 9, 10, 10, 20, 22, 28, 29, 30, 30, 31, 32, 33, 40]
>>> |
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 30, 31, 32]
[2, 3, 5, 10, 20, 22, 28, 29, 30, 33, 40]
bubble: 7.744 detik
selection: 2.724 detik
insertion: 2.336 detik
>>> |
```