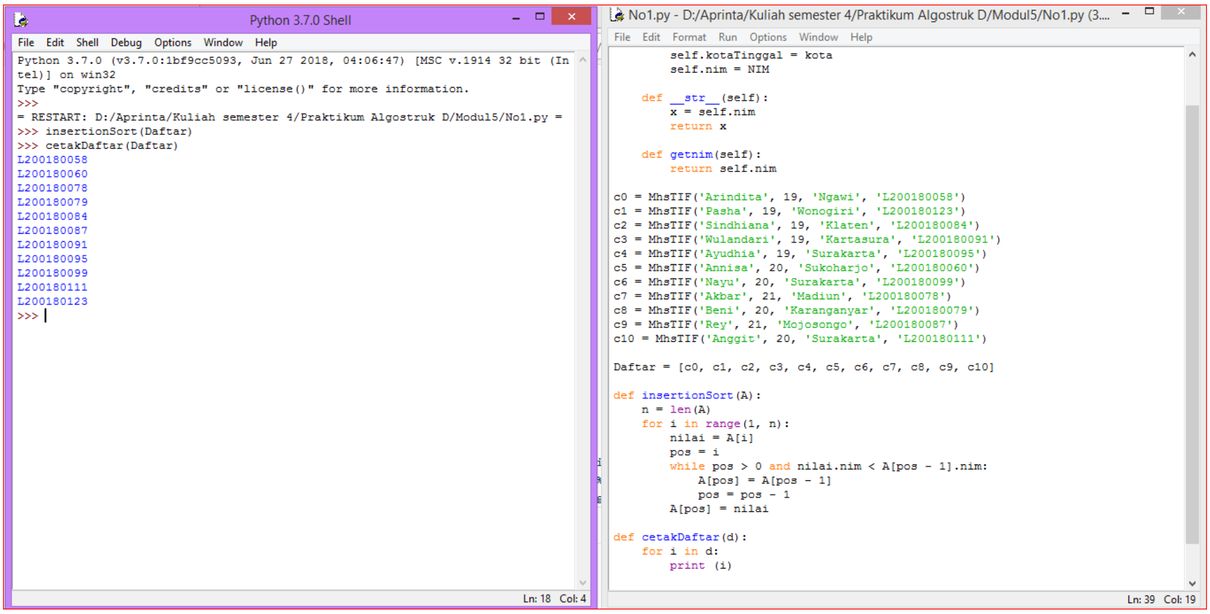
Nama : Aprinta Sewelastami

NIM : L200180088

Kelas : D

MODUL 5
PENGURUTAN

1. Program untuk mengurutkan array mahasiswa berdasarkan NIM

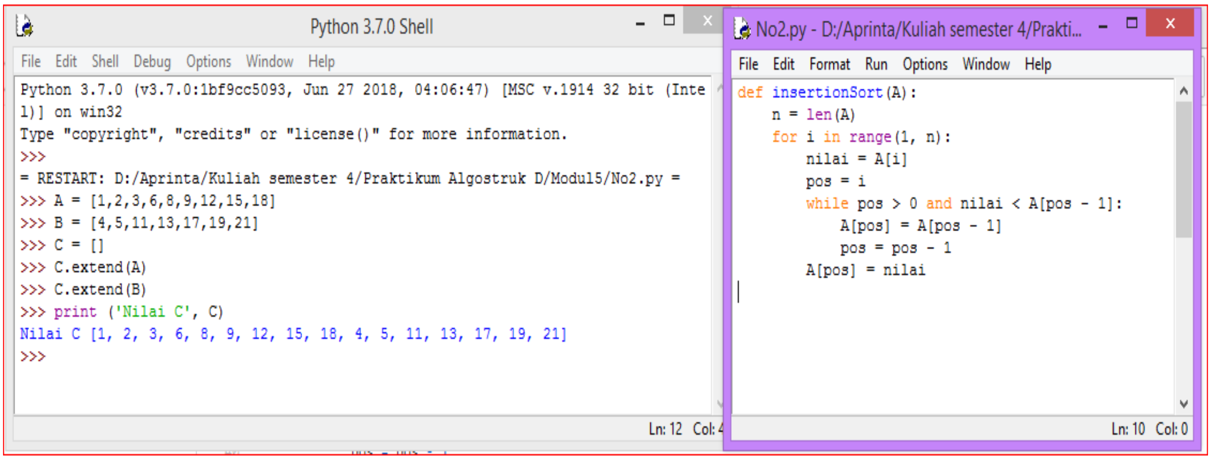


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Ln: 18 Col: 4
No1.py - D:/Aprinta/Kuliah semester 4/Praktikum Algostruk D/Modul5/No1.py (3....
File Edit Format Run Options Window Help
Ln: 39 Col: 19

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Aprinta/Kuliah semester 4/Praktikum Algostruk D/Modul5/No1.py =
>>> insertionSort(Daftar)
>>> cetakDaftar(Daftar)
L200180058
L200180060
L200180078
L200180079
L200180084
L200180087
L200180091
L200180095
L200180099
L200180111
L200180123
>>> 
```

```
        self.kotaTinggal = kota
        self.nim = NIM

    def __str__(self):
        x = self.nim
        return x

    def getnim(self):
        return self.nim

c0 = MhsTIF('Arindita', 19, 'Ngawi', 'L200180058')
c1 = MhsTIF('Pasha', 19, 'Wonogiri', 'L200180123')
c2 = MhsTIF('Sindhiana', 19, 'Klaten', 'L200180084')
c3 = MhsTIF('Wulandari', 19, 'Kartasura', 'L200180091')
c4 = MhsTIF('Ayudhia', 19, 'Surakarta', 'L200180095')
c5 = MhsTIF('Annisa', 20, 'Sukoharjo', 'L200180060')
c6 = MhsTIF('Nayu', 20, 'Surakarta', 'L200180099')
c7 = MhsTIF('Akbar', 21, 'Madiun', 'L200180078')
c8 = MhsTIF('Beni', 20, 'Karanganyar', 'L200180079')
c9 = MhsTIF('Rey', 21, 'Mojosongo', 'L200180087')
c10 = MhsTIF('Anggit', 20, 'Surakarta', 'L200180111')

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]

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

def cetakDaftar(d):
    for i in d:
        print (i)
```

2. Menggabungkan array yag sudah urut

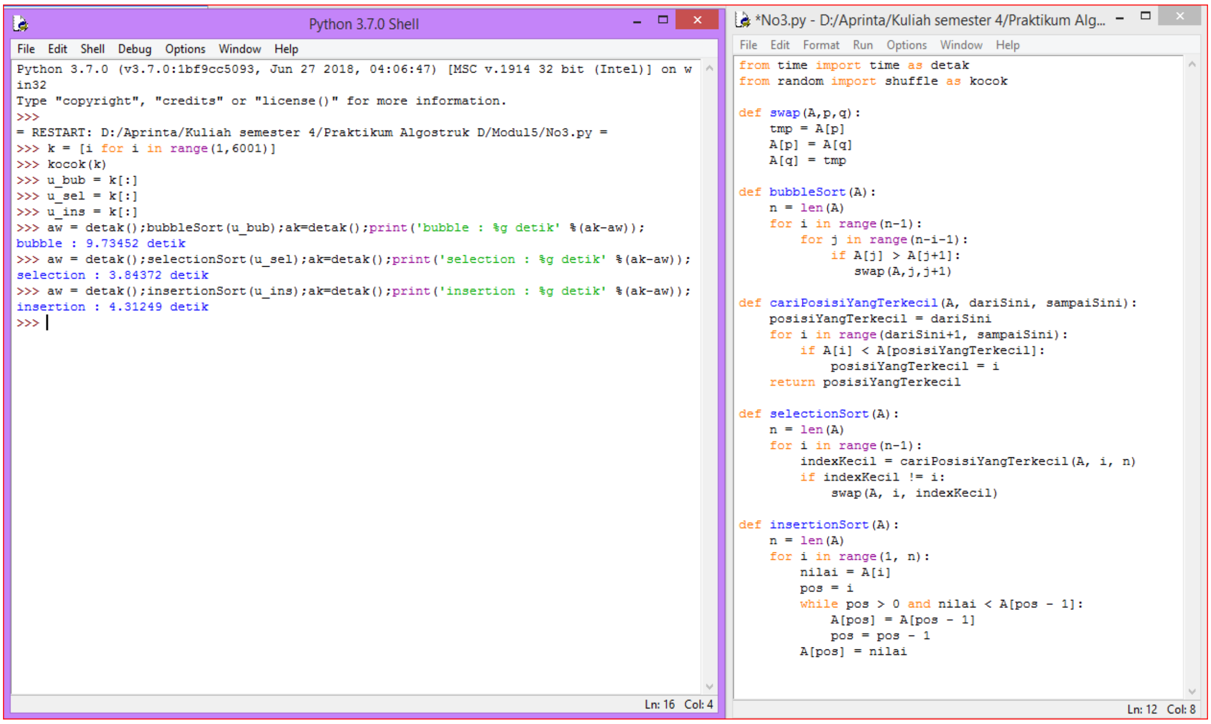


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Ln: 12 Col: 4
No2.py - D:/Aprinta/Kuliah semester 4/Prakti...
File Edit Format Run Options Window Help
Ln: 10 Col: 0

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Aprinta/Kuliah semester 4/Praktikum Algostruk D/Modul5/No2.py =
>>> A = [1,2,3,6,8,9,12,15,18]
>>> B = [4,5,11,13,17,19,21]
>>> C = []
>>> C.extend(A)
>>> C.extend(B)
>>> print ('Nilai C', C)
Nilai C [1, 2, 3, 6, 8, 9, 12, 15, 18, 4, 5, 11, 13, 17, 19, 21]
>>>
```

```
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
```

3. Membandingkan antara bubbleSort, selectionSort dan insertionSort lebih cepat yang mana


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Ln: 16 Col: 4
*No3.py - D:/Aprinta/Kuliah semester 4/Praktikum Alg...
File Edit Format Run Options Window Help
Ln: 12 Col: 8

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on w
in32
Type "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: D:/Aprinta/Kuliah semester 4/Praktikum Algostruk D/Modul5/No3.py =
>>> k = [i for i in range(1,6001)]
>>> kocok(k)
>>> u_bub = k[:]
>>> u_sel = k[:]
>>> u_ins = k[:]
>>> aw = detak();bubbleSort(u_bub);ak=detak();print('bubble : %g detik' %(ak-aw));
bubble : 9.73452 detik
>>> aw = detak();selectionSort(u_sel);ak=detak();print('selection : %g detik' %(ak-aw));
selection : 3.84372 detik
>>> aw = detak();insertionSort(u_ins);ak=detak();print('insertion : %g detik' %(ak-aw));
insertion : 4.31249 detik
>>> |
```

```
from time import time as detak
from random import shuffle as kocok

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

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

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
```

Menurut saya lebih cepat selection, karena selection menggunakan 2 define sehingga mempercepat perulangan