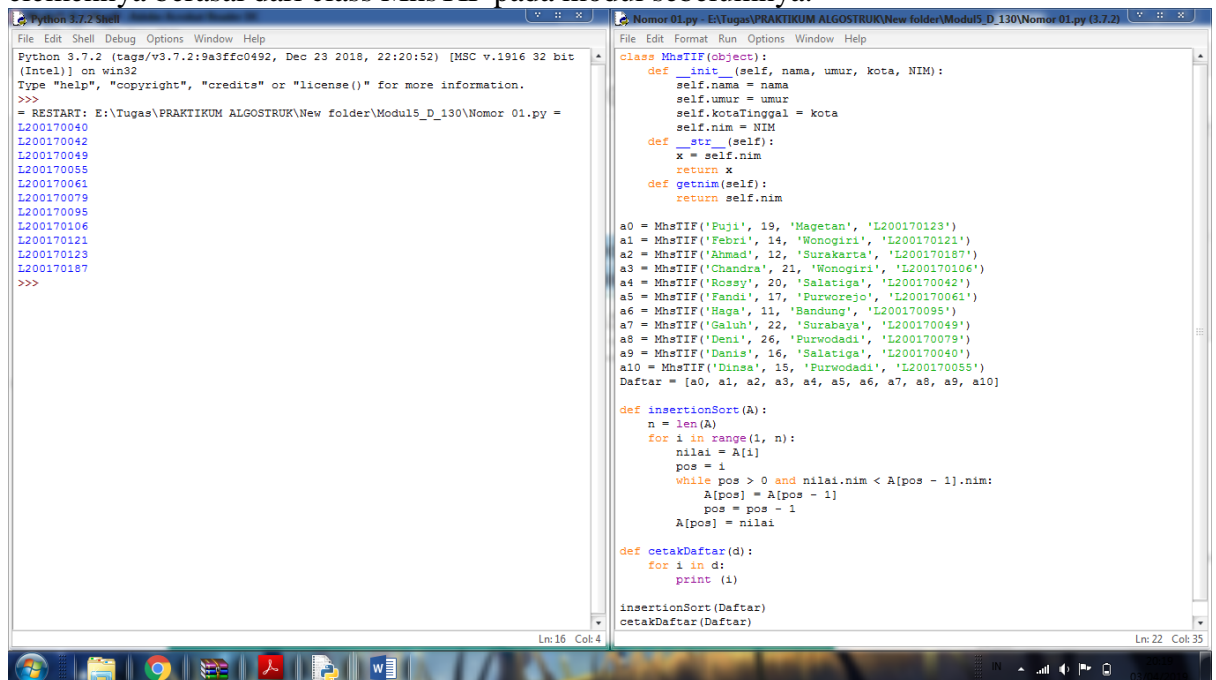


NAMA : PUJI NUGROHO  
NIM : L200170123  
KELAS : D  
MODUL : 5

### Jawab Soal Untuk Mahasiswa

1. Program yang mengurutkan suatu Array Mahasiswa berdasarkan NIM, yang elemennya berasal dari class MhsTIF pada modul sebelumnya.



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Tugas\PRAKTIKUM ALGOSTRUK\New folder\Modul5_D_130\Nomor 01.py =
L200170040
L200170042
L200170049
L200170055
L200170061
L200170079
L200170095
L200170106
L200170121
L200170123
L200170187
>>>
```

```
Nomor 01.py - E:\Tugas\PRAKTIKUM ALGOSTRUK\New folder\Modul5_D_130\Nomor 01.py (3.7.2)
File Edit Format Run Options Window Help
class MhsTIF(object):
    def __init__(self, nama, umur, kota, NIM):
        self.nama = nama
        self.umur = umur
        self.kotaTinggal = kota
        self.nim = NIM
    def __str__(self):
        x = self.nim
        return x
    def getnim(self):
        return self.nim

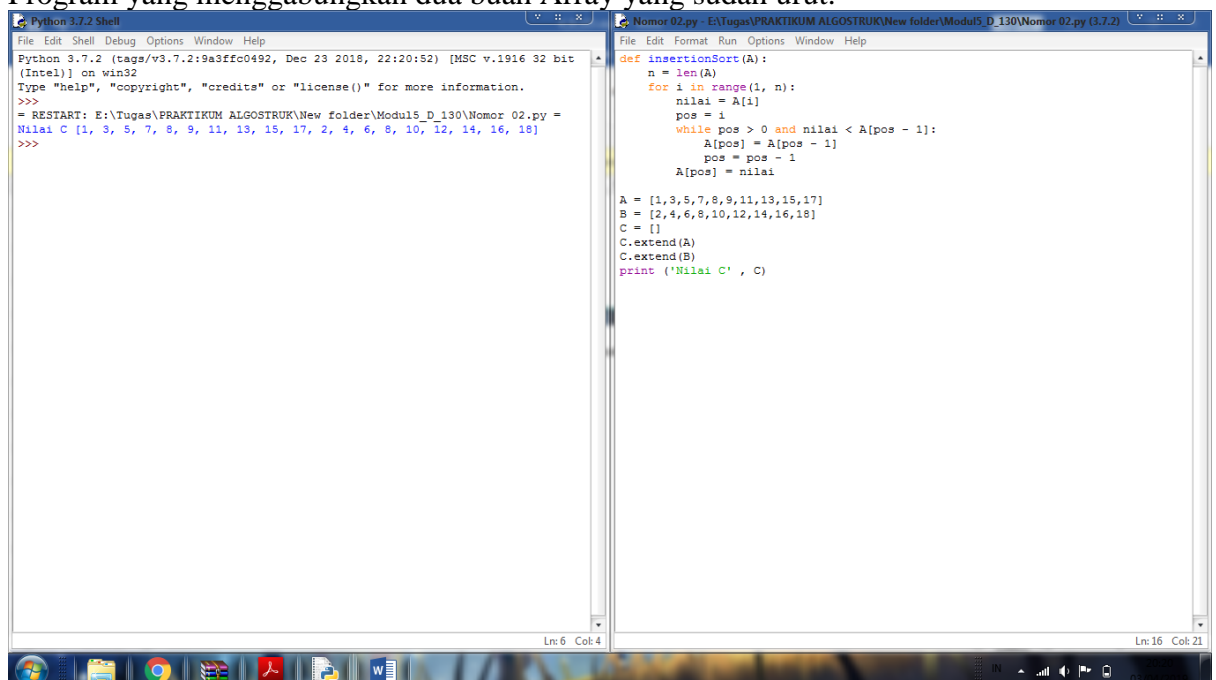
a0 = MhsTIF('Puji', 19, 'Magetan', 'L200170123')
a1 = MhsTIF('Febri', 14, 'Wonogiri', 'L200170121')
a2 = MhsTIF('Ahmad', 12, 'Surakarta', 'L200170187')
a3 = MhsTIF('Chandra', 21, 'Wonogiri', 'L200170106')
a4 = MhsTIF('Rosy', 20, 'Salatiga', 'L200170042')
a5 = MhsTIF('Fandi', 17, 'Purworejo', 'L200170061')
a6 = MhsTIF('Haga', 11, 'Bandung', 'L200170095')
a7 = MhsTIF('Galuh', 22, 'Surabaya', 'L200170049')
a8 = MhsTIF('Deni', 26, 'Purwodadi', 'L200170079')
a9 = MhsTIF('Danis', 16, 'Salatiga', 'L200170040')
a10 = MhsTIF('Dinsa', 15, 'Purwodadi', 'L200170055')
Daftar = [a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10]

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)

insertionSort(Daftar)
cetakDaftar(Daftar)
```

2. Program yang menggabungkan dua buah Array yang sudah urut.

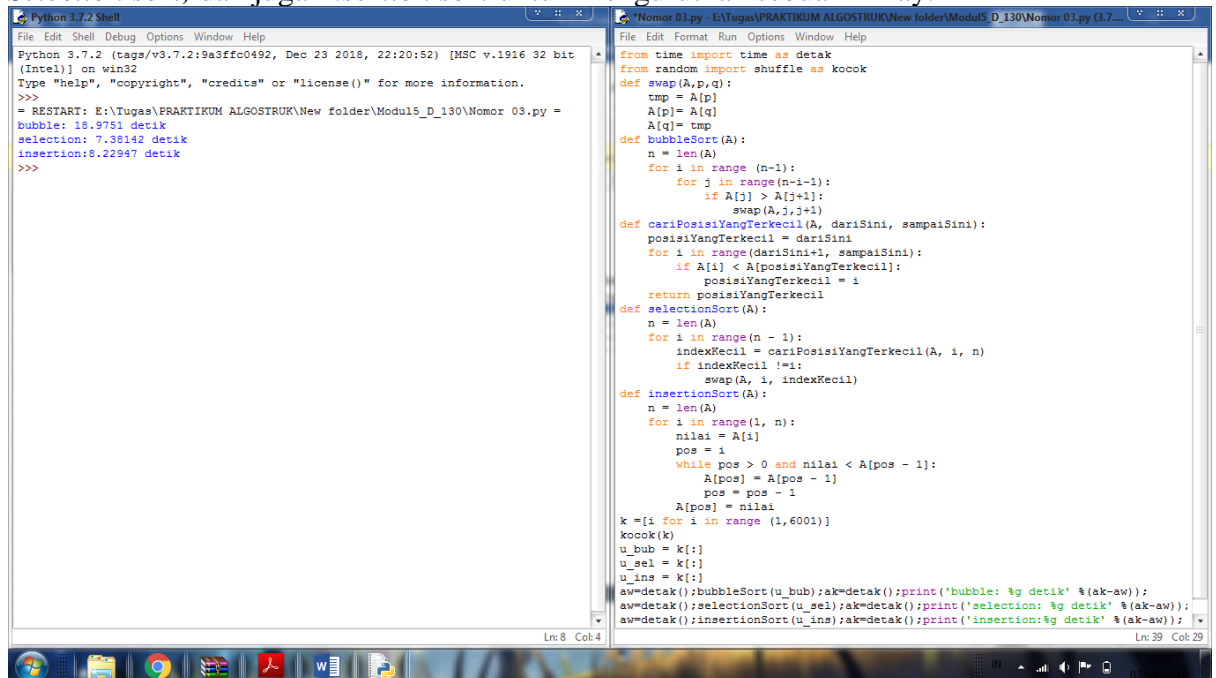


```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Tugas\PRAKTIKUM ALGOSTRUK\New folder\Modul5_D_130\Nomor 02.py =
Nilai C [1, 3, 5, 7, 8, 9, 11, 13, 15, 17, 2, 4, 6, 8, 10, 12, 14, 16, 18]
>>>
```

```
Nomor 02.py - E:\Tugas\PRAKTIKUM ALGOSTRUK\New folder\Modul5_D_130\Nomor 02.py (3.7.2)
File Edit Format Run Options Window Help
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

A = [1,3,5,7,8,9,11,13,15,17]
B = [2,4,6,8,10,12,14,16,18]
C = []
C.extend(A)
C.extend(B)
print ('Nilai C' , C)
```

3. Program yang digunakan untuk membandingkan waktu yang diperlukan *Bubble sort*, *Selection sort*, dan juga *Insertion sort* untuk mengurutkan sebuah Array.



The image shows two side-by-side windows of the Python 3.7.2 Shell. The left window displays the output of a script that compares the execution time of three sorting algorithms: Bubble sort, Selection sort, and Insertion sort. The right window shows the source code of the script.

**Left Window Output:**

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
- RESTART: E:\Tugas\PRAKTIKUM ALGOSTRUK\New folder\Modul5_D_130\Nomor 03.py =
bubble: 18.9751 detik
selection: 7.38142 detik
insertion: 8.22947 detik
>>>
```

**Right Window Code:**

```
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 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

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=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));
```