Nama : Zulfa Fajrul Falah
Nim : L200170149
Kelas : D

LAPORAN PRAKTIKUM ALGORITMA STRUKTUR DATA
MODUL 5

Nomor 1

```
#####NOMOR 1

class Mahasiswa(object):
    """Class Mahasiswa yang dibangun dari class Manusia."""
    def __init__(self, nama, NIM, kota, us):
        """Metode inisiasi ini menutupi metode inisiasi di class Manusia"""
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us

c0 = Mahasiswa('Ika',10,'Sukoharjo',240000)
c1 = Mahasiswa('Budi',51,'Sragen',230000)
c2 = Mahasiswa('Ahmad',2,'Surakarta',250000)
c3 = Mahasiswa('Chandra',18,'Surakarta',235000)
c4 = Mahasiswa('Eka',4,'Boyolali',240000)
c5 = Mahasiswa('Fandi',31,'Salatiga',250000)
c6 = Mahasiswa('Deni',13,'Klaten',245000)
c7 = Mahasiswa('Galuh',5,'Wonogiri',245000)
c8 = Mahasiswa('Janto',23,'Klaten',245000)
c9 = Mahasiswa('Hasan',64,'Karanganyar',270000)
c10 = Mahasiswa('Khalid',29,'Purwodadi',230000)

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

def urutkanNim(A):
    baru = {}
    for i in range(len(A)):
        baru[A[i].nama] = A[i].NIM
    listofTuples = sorted(baru.items(), key=lambda x: x[1])
    for elem in listofTuples :
        print(elem[0] , ":" , elem[1] )
urutkanNim(Daftar)
```

Nomor 2

```

####NOMOR 2

def bubblesort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
def gabung(a,b):
    c = []
    c = a+b
    n = len(c)
    for i in range(n):
        for j in range(0, n-i-1):
            if c[j] > c[j+1] :
                c[j], c[j+1] = c[j+1], c[j]
    return c
a = [8,3,6,13,14,6,13,2]
b = [12,30,53,15,46]
a, b = bubblesort(a), bubblesort(b)

print(a,"\n", b)
print()
print(gabung(a,b))

```

Nomor 3

pengurutan.py | laporan modul 5.py | modul4.py

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

k = [i for i in range(1, 6001)]
kocok(k)


def bubb(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n):

        # Last i elements are already in place
        for j in range(0, n - i - 1):

            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]


def sele(A):
    for i in range(len(A)):

        # Find the minimum element in remaining
        # unsorted array
        min_idx = i
        for j in range(i + 1, len(A)):
            if A[min_idx] > A[j]:
                min_idx = j

                # Swap the found minimum element with
        # the first element
        A[i], A[min_idx] = A[min_idx], A[i]


def inse(arr):
    # Traverse through 1 to len(arr)
    for i in range(1, len(arr)):

        key = arr[i]

        # Move elements of arr[0..i-1], that are
        # greater than key, to one position ahead
        # of their current position
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key
```

```
55
56      bub = k[:]
57      sel = k[:]
58      ins = k[:]
59
60      aw = detak();
61      bubb(bub);
62      ak = detak();
63      print('bubble : %g detik' % (ak - aw));
64      aw = detak();
65      sele(sel);
66      ak = detak();
67      print('selection : %g detik' % (ak - aw));
68      aw = detak();
69      inse(ins);
70      ak = detak();
71      print('insertion : %g detik' % (ak - aw));
72
```