LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
MODUL 6
PENGURUTAN LANJUTAN

Disusun oleh:

Bima Triadmaja

L200210137

E

TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2022/2023

6.4 Soal-soal untuk Mahasiswa

Nomor 1 :

```
#NO.1
class Mahasiswa(object):
    def __init__(self, nama,NIM,kota,us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us

c0 = Mahasiswa('Bima','L200210137','Sukoharjo', 240000)
c1 = Mahasiswa('Triadmaja','L200210351','Sragen', 230000)
c2 = Mahasiswa('Risma','L200210302','Surakarta', 250000)
c3 = Mahasiswa('Nanda','L200210318','Surakarta', 235000)
c4 = Mahasiswa('Fatika','L200210304','Boyolali', 240000)
c5 = Mahasiswa('Sari','L200210331','Salatiga', 250000)
c6 = Mahasiswa('Dimas','L200210313','Klaten', 245000)
c7 = Mahasiswa('Cahyo','L200210305','Wonogiri', 245000)
c8 = Mahasiswa('Vikki','L200210323','Klaten', 245000)
c9 = Mahasiswa('Gilang','L200210364','Karanganyar', 270000)
c10 = Mahasiswa('Eko','L200210329','Purwodadi', 265000)

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

def mergeSort(A):
    if len(A) > 1:
        mid = len(A) // 2
        separuhKiri = A[:mid]
        separuhKanan = A[mid:]

        mergeSort(separuhKiri)
        mergeSort(separuhKanan)

        i=0 ; j=0 ; k=0
        while i < len(separuhKiri) and j < len(separuhKanan):
            if separuhKiri[i] < separuhKanan[j]:
                A[k] = separuhKiri[i]
                i = i + 1
            else:
                A[k] = separuhKanan[j]
                j = j + 1
            k = k+1
```

```
        while i < len(separuhKiri):
            A[k] = separuhKiri[i]
            i = i + 1
            k = k + 1
        while j < len(separuhKanan):
            A[k] = separuhKanan[j]
            j = j + 1
            k = k + 1

def convert(arr, obj):
    hasil = []
    for x in range (len(arr)):
        for i in range (len(arr)):
            if arr[x] == obj[i].NIM:
                hasil.append(obj[i])
    return hasil

A = []
for x in Daftar:
    A.append(x.NIM)

print("---------- Merge Sort ----------")
mergeSort(A)
for i in convert(A, Daftar):
    print(i.nama, i.NIM, i.kotaTinggal, i.uangSaku)
print()

def partisi(A, awal, akhir):
    nilaiPivot = A[awal]

    penandaKiri = awal + 1
    penandaKanan = akhir

    selesai = False
    while not selesai:

        while penandaKiri <= penandaKanan and \
              A[penandaKiri] <= nilaiPivot:
            penandaKiri = penandaKiri + 1

        while A[penandaKanan] >= nilaiPivot and \
              penandaKanan >= penandaKiri:
            penandaKanan = penandaKanan - 1
```

```
        if penandaKanan < penandaKiri:
            selesai = True
        else:
            temp = A[penandaKiri]
            A[penandaKiri] = A[penandaKanan]
            A[penandaKanan] = temp

    temp = A[awal]
    A[awal] = A[penandaKanan]
    A[penandaKanan] = temp

    return penandaKanan

def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah = partisi(A, awal, akhir)
        quickSortBantu(A, awal, titikBelah - 1)
        quickSortBantu(A, titikBelah + 1, akhir)

def quickSort(A):
    quickSortBantu(A, 0, len(A) - 1)

def convert(arr, obj):
    hasil = []
    for x in range (len(arr)):
        for i in range (len(arr)):
            if arr[x] == obj[i].NIM:
                hasil.append(obj[i])
    return hasil

A = []
for x in Daftar:
    A.append(x.NIM)

print("---------- Quick Sort ----------")
quickSort(A)
for i in convert(A, Daftar):
    print(i.nama, i.NIM, i.kotaTinggal, i.uangSaku)

print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```

IDLE Shell 3.10.1

File Edit Shell Debug Options Window Help

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 6\nomor1.py
---------- Merge Sort ----------
Bima L200210137 Sukoharjo 240000
Risma L200210302 Surakarta 250000
Fatika L200210304 Boyolali 240000
Cahyo L200210305 Wonogiri 245000
Dimas L200210313 Klaten 245000
Nanda L200210318 Surakarta 235000
Vikki L200210323 Klaten 245000
Eko L200210329 Purwodadi 265000
Sari L200210331 Salatiga 250000
Triadmaja L200210351 Sragen 230000
Gilang L200210364 Karanganyar 270000

---------- Quick Sort ----------
Bima L200210137 Sukoharjo 240000
Risma L200210302 Surakarta 250000
Fatika L200210304 Boyolali 240000
Cahyo L200210305 Wonogiri 245000
Dimas L200210313 Klaten 245000
Nanda L200210318 Surakarta 235000
Vikki L200210323 Klaten 245000
Eko L200210329 Purwodadi 265000
Sari L200210331 Salatiga 250000
Triadmaja L200210351 Sragen 230000
Gilang L200210364 Karanganyar 270000

Sudah selesai.

--- Oleh L200210137 ---
>>>
```

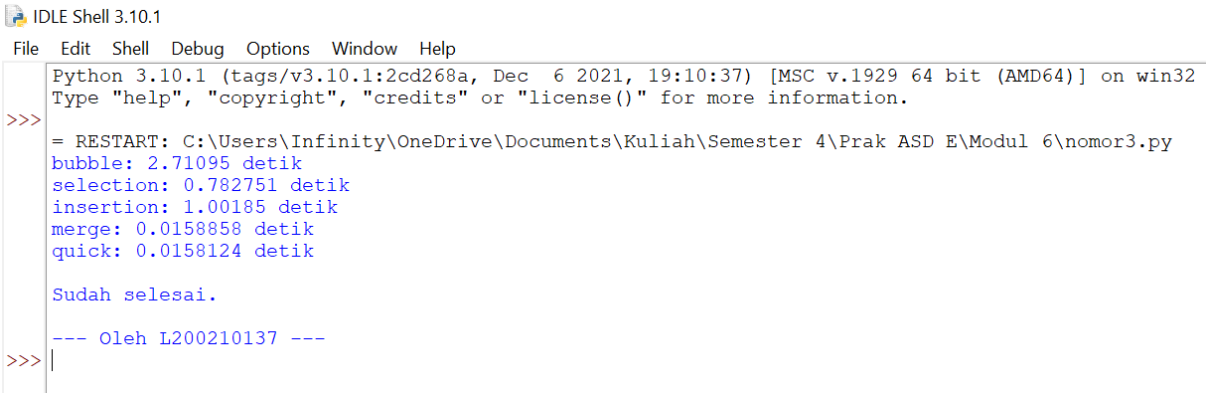
Nomor 3 :

```
#NO.3
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[1] < A[posisiTerkecil]:
            posisiTerkecil = 1
    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

def mergeSort(A):

    if len(A) > 1:
        mid = len(A) // 2
        separuhKiri = A[:mid]
        separuhKanan = A[mid:]
```

```
        mergeSort(separuhKiri)
        mergeSort(separuhKanan)


        i=0 ; j=0 ; k=0
        while i < len(separuhKiri) and j < len(separuhKanan):
            if separuhKiri[i] < separuhKanan[j]:
                A[k] = separuhKiri[i]
                i = i + 1
            else:
                A[k] = separuhKanan[j]
                j = j + 1
            k=k+1

        while i < len(separuhKiri):
            A[k] = separuhKiri[i]
            i = i + 1
            k = k + 1

        while j < len(separuhKanan):
            A[k] = separuhKanan[j]
            j = j + 1
            k = k + 1



def quickSort(A):
    quickSortBantu(A, 0, len(A) - 1 )
def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah = partisi(A, awal, akhir)
        quickSortBantu(A, awal, titikBelah - 1)
        quickSortBantu(A, titikBelah + 1, akhir)

def partisi(A, awal, akhir):
    nilaiPivot = A[awal]

    penandaKiri = awal + 1
    penandaKanan = akhir

    selesai = False
    while not selesai:
```

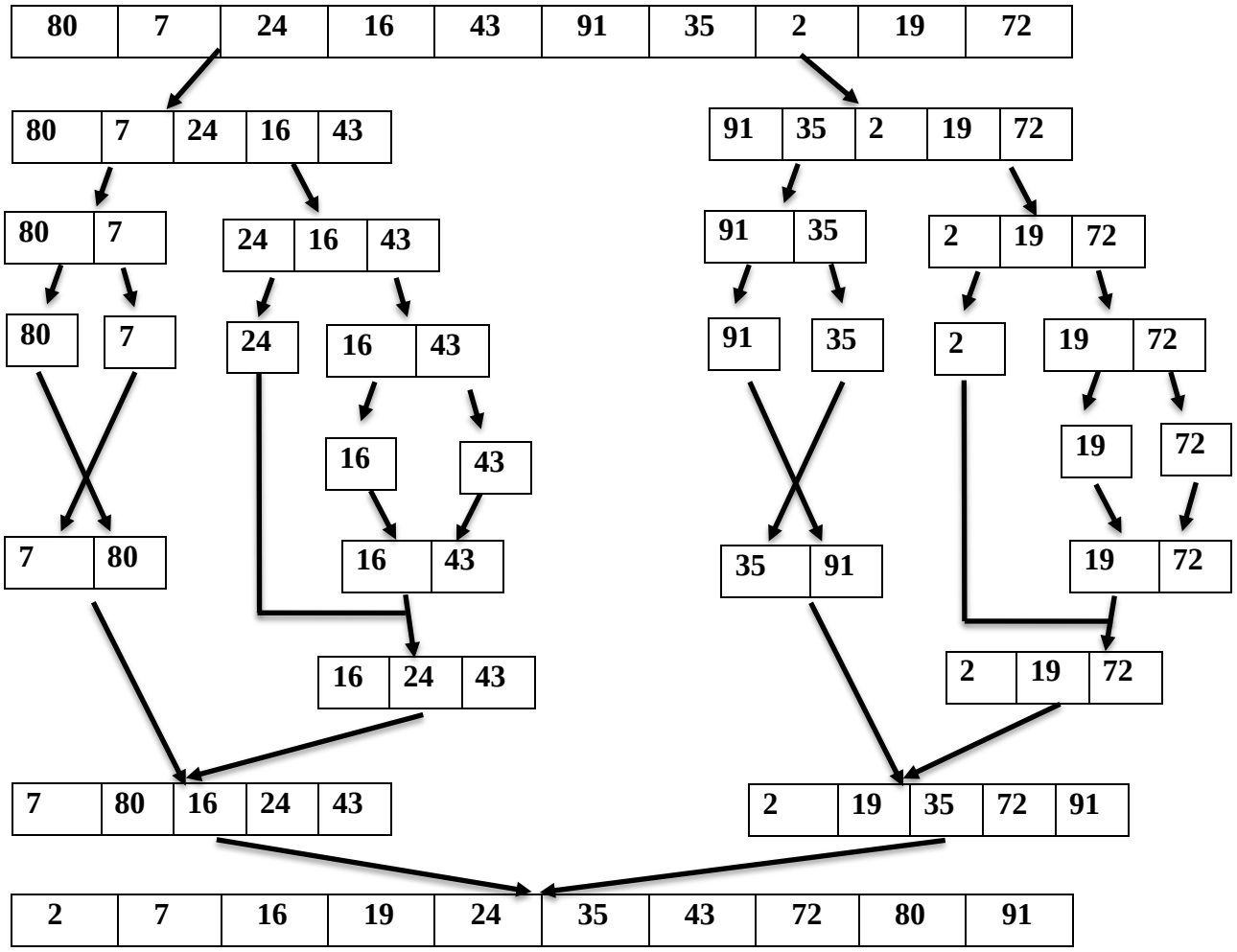
```
        while penandaKiri <= penandaKanan and A[penandaKiri]
<= nilaiPivot:

            penandaKiri = penandaKiri + 1

        while A[penandaKanan] >= nilaiPivot and penandaKanan
>= penandaKiri:
            penandaKanan = penandaKanan - 1

        if penandaKanan < penandaKiri:
            selesai = True
        else:
            temp = A[penandaKiri]
            A[penandaKiri] = A[penandaKanan]
            A[penandaKanan] = temp

    temp = A[awal]
    A[awal] = A[penandaKanan]
    A[penandaKanan] = temp

    return penandaKanan

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

k = [i for i in range(1,6001)]
kocok(k)
u_bub = k[:]
u_sel = k[:]
u_ins = k[:]
u_mrg = k[:]
u_qck = k[:]

aw=detak();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) );
aw=detak();mergeSort(u_mrg);ak=detak();print('merge: %g detik'
%(ak-aw) );
aw=detak();quickSort(u_qck);ak=detak();print('quick: %g detik'
%(ak-aw) );
```

```
print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```

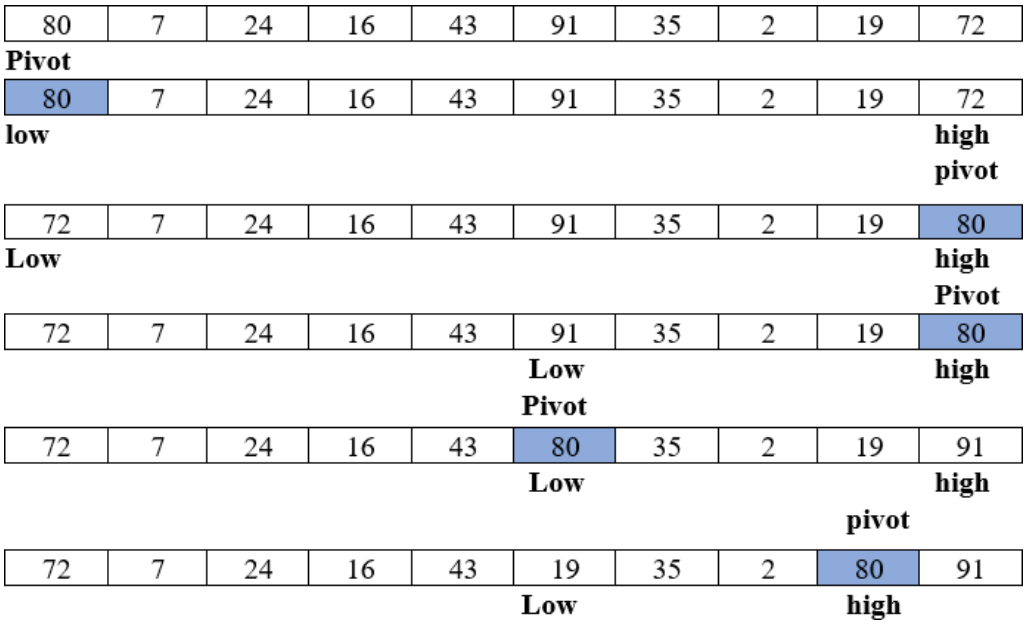


Nomor 4 :

a) Merge Sort



b) Quick Sort



Nomor 5 :

```
#NO.5
class Mahasiswa:

    def __init__(self,nama,NIM,kota,us):

        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.UangSaku = us

    def __str__(self):

        return ("Nama {}, NIM {}, Kota {}, Uang Saku {}".

format(self.nama,self.NIM,self.kotaTinggal,self.UangSaku))
    def ambilNama(self):
        return self.nama
    def ambilNim(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.UangSaku

c0 = Mahasiswa('Bima','L200210137','Sukoharjo', 240000)
c1 = Mahasiswa('Triadmaja','L200210351','Sragen', 230000)
c2 = Mahasiswa('Risma','L200210302','Surakarta', 250000)
c3 = Mahasiswa('Nanda','L200210318','Surakarta', 235000)
c4 = Mahasiswa('Fatika','L200210304','Boyolali', 240000)
c5 = Mahasiswa('Sari','L200210331','Salatiga', 250000)
c6 = Mahasiswa('Dimas','L200210313','Klaten', 245000)
c7 = Mahasiswa('Cahyo','L200210305','Wonogiri', 245000)
c8 = Mahasiswa('Vikki','L200210323','Klaten', 245000)
c9 = Mahasiswa('Gilang','L200210364','Karanganyar', 270000)
c10 = Mahasiswa('Eko','L200210329','Purwodadi', 265000)


c0.next = c1
c1.next = c2
c2.next = c3
c3.next = c4
c4.next = c5
c5.next = c6
c6.next = c7
c7.next = c8
```

```
c8.next = c9
c9.next = c10


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

def cetak(A):
    for i in A:
        print (i)

def mergeSort2(A, awal, akhir):
    mid = (awal+akhir)//2
    if awal < akhir:
        mergeSort2(A, awal, mid)
        mergeSort2(A, mid+1, akhir)

    a, y, l = 0, awal, mid+1
    tmp = [None] * (akhir - awal + 1)
    while y <= mid and l <= akhir:
        if A[y].ambilNim() < A[l].ambilNim():
            tmp[a] = A[y]
            y += 1
        else:
            tmp[a] = A[l]
            l += 1
        a += 1

    if y <= mid:
        tmp[a:] = A[y:mid+1]

    if l <= akhir:
        tmp[a:] = A[l:akhir+1]

    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

def mergeSort(A):
    mergeSort2(A, 0, len(A)-1)

print("---------------------- Sebelum diurutkan --------------
--------")
```

```
cetak(Daftar)
mergeSort(Daftar)
print("\n---------------------- Setelah diurutkan ------------
----------")
cetak(Daftar)

print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```

IDLE Shell 3.10.1

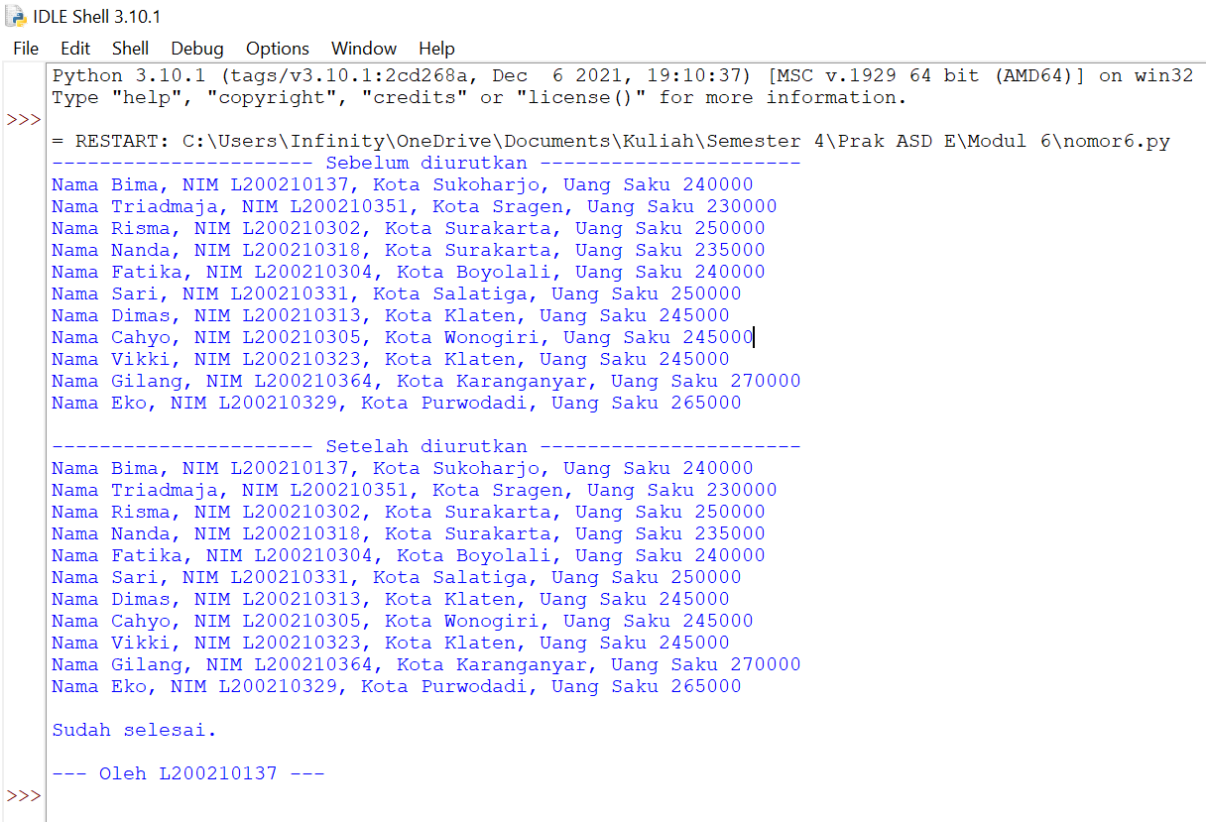
File Edit Shell Debug Options Window Help

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 6\nomor5.py
---------------------- Sebelum diurutkan ----------------------
Nama Bima, NIM L200210137, Kota Sukoharjo, Uang Saku 240000
Nama Triadmaja, NIM L200210351, Kota Sragen, Uang Saku 230000
Nama Risma, NIM L200210302, Kota Surakarta, Uang Saku 250000
Nama Nanda, NIM L200210318, Kota Surakarta, Uang Saku 235000
Nama Fatika, NIM L200210304, Kota Boyolali, Uang Saku 240000
Nama Sari, NIM L200210331, Kota Salatiga, Uang Saku 250000
Nama Dimas, NIM L200210313, Kota Klaten, Uang Saku 245000
Nama Cahyo, NIM L200210305, Kota Wonogiri, Uang Saku 245000
Nama Vikki, NIM L200210323, Kota Klaten, Uang Saku 245000
Nama Gilang, NIM L200210364, Kota Karanganyar, Uang Saku 270000
Nama Eko, NIM L200210329, Kota Purwodadi, Uang Saku 265000

---------------------- Setelah diurutkan ----------------------
Nama Bima, NIM L200210137, Kota Sukoharjo, Uang Saku 240000
Nama Risma, NIM L200210302, Kota Surakarta, Uang Saku 250000
Nama Fatika, NIM L200210304, Kota Boyolali, Uang Saku 240000
Nama Cahyo, NIM L200210305, Kota Wonogiri, Uang Saku 245000
Nama Dimas, NIM L200210313, Kota Klaten, Uang Saku 245000
Nama Nanda, NIM L200210318, Kota Surakarta, Uang Saku 235000
Nama Vikki, NIM L200210323, Kota Klaten, Uang Saku 245000
Nama Eko, NIM L200210329, Kota Purwodadi, Uang Saku 265000
Nama Sari, NIM L200210331, Kota Salatiga, Uang Saku 250000
Nama Triadmaja, NIM L200210351, Kota Sragen, Uang Saku 230000
Nama Gilang, NIM L200210364, Kota Karanganyar, Uang Saku 270000

Sudah selesai.

--- Oleh L200210137 ---
>>>
```

Nomor 6 :

```
#NO.6
class Mahasiswa:
    """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

    def __str__(self):

        return ("Nama {}, NIM {}, Kota {}, Uang Saku {}"

.format(self.nama,self.NIM,self.kotaTinggal,self.UangSaku))
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.UangSaku

c0 = Mahasiswa('Bima','L200210137','Sukoharjo', 240000)
c1 = Mahasiswa('Triadmaja','L200210351','Sragen', 230000)
c2 = Mahasiswa('Risma','L200210302','Surakarta', 250000)
c3 = Mahasiswa('Nanda','L200210318','Surakarta', 235000)
c4 = Mahasiswa('Fatika','L200210304','Boyolali', 240000)
c5 = Mahasiswa('Sari','L200210331','Salatiga', 250000)
c6 = Mahasiswa('Dimas','L200210313','Klaten', 245000)
c7 = Mahasiswa('Cahyo','L200210305','Wonogiri', 245000)
c8 = Mahasiswa('Vikki','L200210323','Klaten', 245000)
c9 = Mahasiswa('Gilang','L200210364','Karanganyar', 270000)
c10 = Mahasiswa('Eko','L200210329','Purwodadi', 265000)


c0.next = c1
c1.next = c2
c2.next = c3
c3.next = c4
c4.next = c5
c5.next = c6
c6.next = c7
```

```
c7.next = c8
c8.next = c9
c9.next = c10


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


def cetak(A):
    for i in A:
        print(i)

def quickSort(arr):
    kurang = []
    pivotList = []
    lebih = []
    if len(arr) <= 1:
        return arr
    else:
        pivot = arr[0]
        for i in arr:
            if i.ambilNIM() < pivot.ambilNIM():
                kurang.append(i)
            elif i.ambilNIM() > pivot.ambilNIM():
                lebih.append(i)
            else:
                pivotList.append(i)
        kurang = quickSort(kurang)
        lebih = quickSort(lebih)
        return kurang + pivotList + lebih

print("---------------------- Sebelum diurutkan --------------
--------")
cetak(Daftar)
print("\n---------------------- Setelah diurutkan ------------
----------")
quickSort(Daftar)
cetak(Daftar)

print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```


IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 6\nomor6.py
---------- Sebelum diurutkan ----------
Nama Bima, NIM L200210137, Kota Sukoharjo, Uang Saku 240000
Nama Triadmaja, NIM L200210351, Kota Sragen, Uang Saku 230000
Nama Risma, NIM L200210302, Kota Surakarta, Uang Saku 250000
Nama Nanda, NIM L200210318, Kota Surakarta, Uang Saku 235000
Nama Fatika, NIM L200210304, Kota Boyolali, Uang Saku 240000
Nama Sari, NIM L200210331, Kota Salatiga, Uang Saku 250000
Nama Dimas, NIM L200210313, Kota Klaten, Uang Saku 245000
Nama Cahyo, NIM L200210305, Kota Wonogiri, Uang Saku 245000
Nama Vikki, NIM L200210323, Kota Klaten, Uang Saku 245000
Nama Gilang, NIM L200210364, Kota Karanganyar, Uang Saku 270000
Nama Eko, NIM L200210329, Kota Purwodadi, Uang Saku 265000
---------- Setelah diurutkan ----------
Nama Bima, NIM L200210137, Kota Sukoharjo, Uang Saku 240000
Nama Triadmaja, NIM L200210351, Kota Sragen, Uang Saku 230000
Nama Risma, NIM L200210302, Kota Surakarta, Uang Saku 250000
Nama Nanda, NIM L200210318, Kota Surakarta, Uang Saku 235000
Nama Fatika, NIM L200210304, Kota Boyolali, Uang Saku 240000
Nama Sari, NIM L200210331, Kota Salatiga, Uang Saku 250000
Nama Dimas, NIM L200210313, Kota Klaten, Uang Saku 245000
Nama Cahyo, NIM L200210305, Kota Wonogiri, Uang Saku 245000
Nama Vikki, NIM L200210323, Kota Klaten, Uang Saku 245000
Nama Gilang, NIM L200210364, Kota Karanganyar, Uang Saku 270000
Nama Eko, NIM L200210329, Kota Purwodadi, Uang Saku 265000
Sudah selesai.
--- Oleh L200210137 ---
>>>

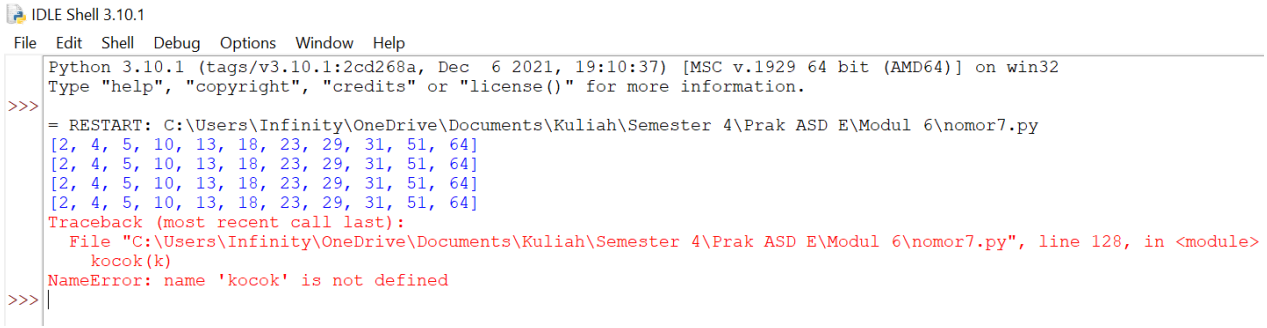
Nomor 7 :

```
#NO.7
def mergeSort(A):
    if len(A) > 1:
        mid = len(A) // 2
        separuhkiri = A[:mid]
        separuhkanan = A[mid:]

        mergeSort(separuhkiri)
        mergeSort(separuhkanan)

        i = 0;j=0;k=0
        while i < len(separuhkiri) and j < len(separuhkanan):
            if separuhkiri[i] < separuhkanan[j]:
                A[k] = separuhkiri[i]
                i = i + 1
            else:
                A[k] = separuhkanan[j]
                j = j + 1
            k=k+1

        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i = i + 1
            k=k+1

        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j = j + 1
            k=k+1

def partisi(A, awal, akhir):
    nilaipivot = A[awal]

    penandakiri = awal + 1
    penandakanan = akhir

    selesai = False
    while not selesai:

        while penandakiri <= penandakanan and A[penandakiri]
<= nilaipivot:
            penandakiri = penandakiri + 1
```

```
        while penandakanan >= penandakiri and A[penandakanan]
>= nilaipivot:
            penandakanan = penandakanan - 1

        if penandakanan < penandakiri:
            selesai = True
        else:
            temp = A[penandakiri]
            A[penandakiri] = A[penandakanan]
            A[penandakanan] = temp

    temp = A[awal]
    A[awal] = A[penandakanan]
    A[penandakanan] = temp

    return penandakanan

def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah = partisi(A, awal, akhir)
        quickSortBantu(A, awal, titikBelah-1)
        quickSortBantu(A, titikBelah+1, akhir)

def quickSort(A):
    quickSortBantu (A, 0, len(A)-1)

def mergeSort2(A, awal, akhir):
    mid = (awal+akhir)//2
    if awal < akhir:
        mergeSort2(A, awal, mid)
        mergeSort2(A, mid+1, akhir)

    a, f, l = 0, awal, mid+1
    tmp = [None] * (akhir - awal + 1)
    while f <= mid and l <= akhir:
        if A[f] < A[l]:
            tmp[a] = A[f]
            f += 1
        else:
            tmp[a] = A[l]
            l += 1
        a += 1
    if f <= mid:
        tmp[a:] = A[f:mid+1]
```

```
    if l <= akhir:
        tmp[a:] = A[l:akhir+1]
    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

def mergeSortNew(A):
    mergeSort2(A, 0, len(A)-1)

def quickSortNew(arr):
    kurang = []
    pivotList = []
    lebih = []
    if len(arr) <= 1:
        return arr
    else:
        pivot = arr[0]
        for i in arr:
            if i < pivot:
                kurang.append(i)
            elif i > pivot:
                lebih.append(i)
            else:
                pivotList.append(i)
        kurang = quickSortNew(kurang)
        lebih = quickSortNew(lebih)
        return kurang + pivotList + lebih

daftar = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]

mergeSort(daftar)
print (daftar)
quickSort(daftar)
print (daftar)
mergeSortNew(daftar)
print (daftar)
quickSortNew(daftar)
print (daftar)

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

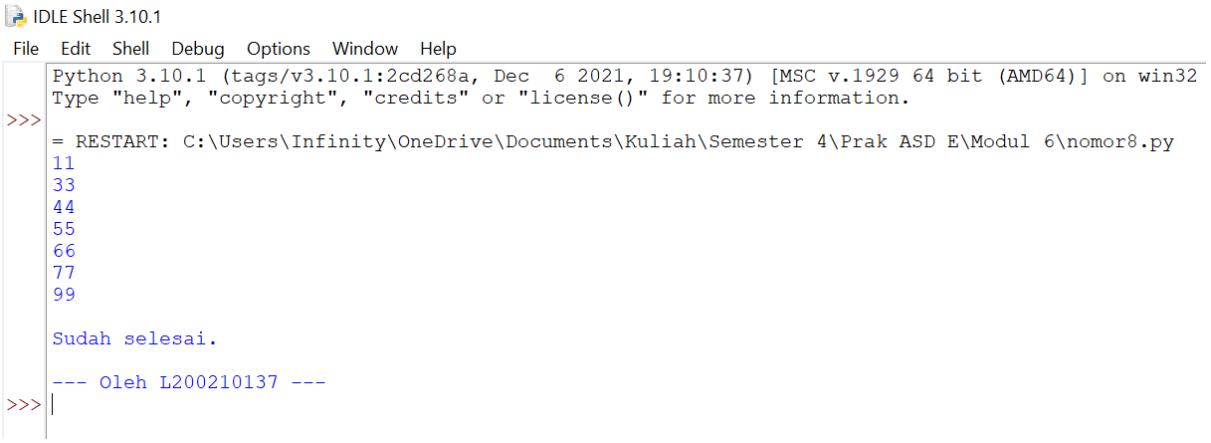
```
u_qck = k[:]
u_mrgNew = k[:]
u_qckNew = k[:]

aw=detak();mergeSort(u_mrg);ak=detak();print("mergeSort : %g
detik" %(ak-aw));
aw=detak();quickSort(u_qck);ak=detak();print("quickSort: %g
detik" %(ak-aw));
aw=detak();mergeSortNew(u_mrgNew);ak=detak();print("mergeSort
baru: %g detik" %(ak-aw));
aw=detak();quickSortNew(u_qckNew);ak=detak();print("quickSort
baru: %g detik" %(ak-aw));

print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```



Nomor 8 :

```
#NO.8
class Node():
    def __init__(self, data, tautan=None):
        self.data = data
        self.taut = tautan

def cetak(head):
    curr = head
    while curr is not None:
        try:
            print (curr.data)
            curr = curr.taut
        except:
            pass

a = Node(11)
b = Node(33)
c = Node(55)
d = Node(66)
e = Node(44)
f = Node(99)
g = Node(77)

a.taut = b
b.taut = c
c.taut = d
d.taut = e
e.taut = f
f.taut = g

def mergeSortLL(A):
    linked = A
    try:
        daftar = []
        curr = A
        while curr:
            daftar.append(curr.data)
            curr = curr.taut
        A = daftar
    except:
        A = A

    if len(A) > 1:
```

```
        mid = len(A) // 2
        separuhkiri = A[:mid]
        separuhkanan = A[mid:]

        mergeSortLL(separuhkiri)
        mergeSortLL(separuhkanan)

        i = 0;j=0;k=0
        while i < len(separuhkiri) and j < len(separuhkanan):
            if separuhkiri[i] < separuhkanan[j]:
                A[k] = separuhkiri[i]
                i = i + 1
            else:
                A[k] = separuhkanan[j]
                j = j + 1
            k=k+1
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i = i + 1
            k=k+1
        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j = j + 1
            k=k+1
    for x in A:
        try:
            linked.data = x
            linked = linked.taut
        except:
            pass

mergeSortLL(a)
cetak(a)

print('')
print('Sudah selesai.')

print('\n--- Oleh L200210137 ---')
```


IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Infinity\OneDrive\Documents\Kuliah\Semester 4\Prak ASD E\Modul 6\nomor8.py
11
33
44
55
66
77
99
Sudah selesai.
--- Oleh L200210137 ---
>>>