# LAPORAN

# PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

# ( MODUL 6 )

## “ PENGURUTAN LANJUTAN”


**Disusun oleh :**

**NAMA : CINDI DILA APRILIANA**

**NIM : L200200106**

**KELAS : E**

**INFORMATIKA**

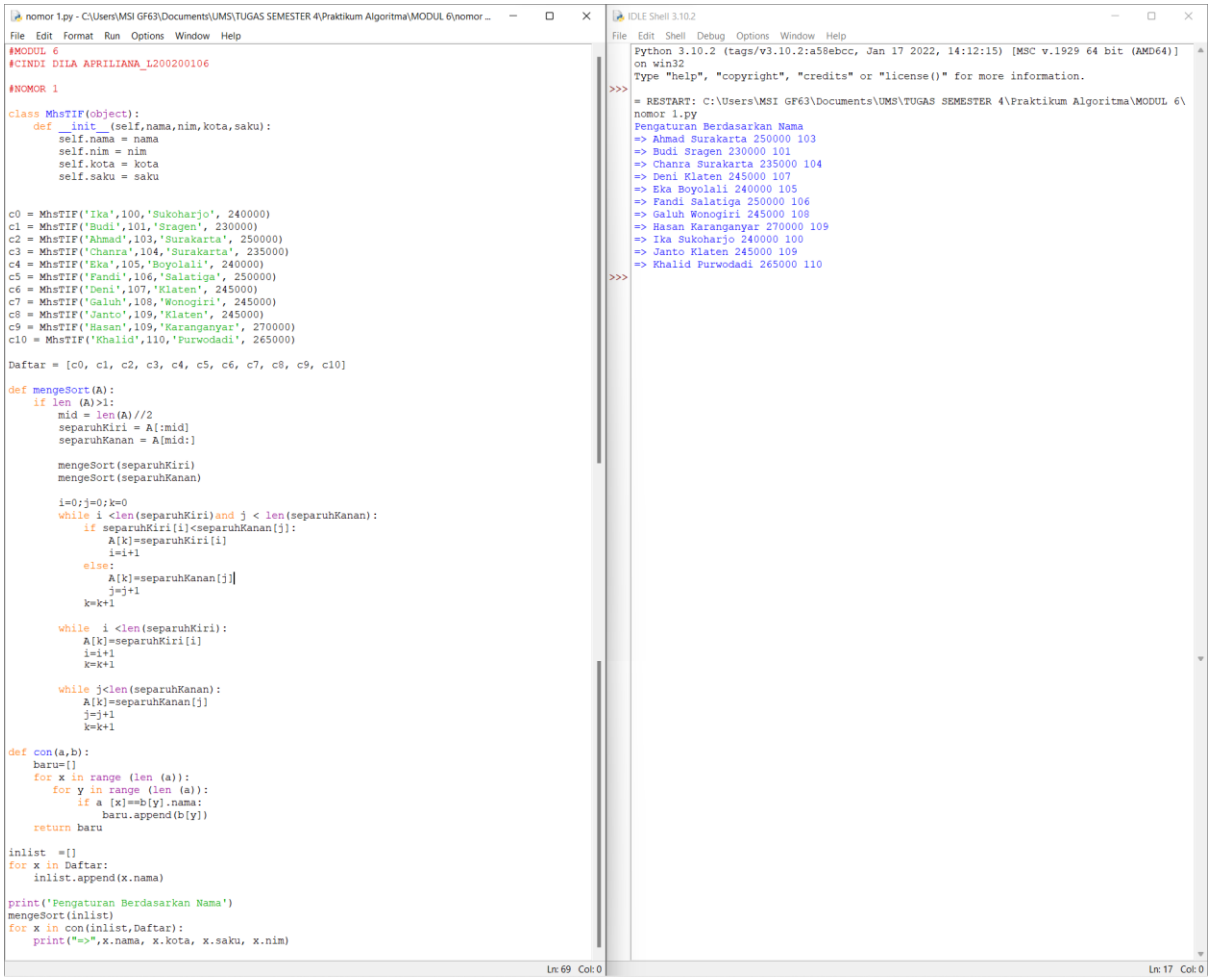
**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**TAHUN 2021/2022**

1. Ubahlah kode mergeSort dan quickSort di atas agar bisa mengurutkan list yang berisi object-object mhsTIF yang sudah kamu buat di Modul 2. Uji programmu secukupnya.

Soal 1



```
#MODUL 6
#CINDI DILA APRILIANA_L200200106

#NOMOR 1

class MhsTIF(object):
    def __init__(self,nama,nim,kota,saku):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.saku = saku


c0 = MhsTIF('Ika',100,'Sukoharjo', 240000)
c1 = MhsTIF('Budi',101,'Sragen', 230000)
c2 = MhsTIF('Ahmad',103,'Surakarta', 250000)
c3 = MhsTIF('Chanra',104,'Surakarta', 235000)
c4 = MhsTIF('Eka',105,'Boyolali', 240000)
c5 = MhsTIF('Fandi',106,'Salatiga', 250000)
c6 = MhsTIF('Deni',107,'Klaten', 245000)
c7 = MhsTIF('Galuh',108,'Wonogiri', 245000)
c8 = MhsTIF('Janto',109,'Klaten', 245000)
c9 = MhsTIF('Hasan',109,'Karanganyar', 270000)
c10 = MhsTIF('Khalid',110,'Purwodadi', 265000)

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

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

        mengeSort(separuhKiri)
        mengeSort(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 con(a,b):
    baru=[]
    for x in range (len (a)):
       for y in range (len (a)):
           if a [x]==b[y].nama:
               baru.append(b[y])
    return baru

inlist  =[]
for x in Daftar:
    inlist.append(x.nama)

print('Pengaturan Berdasarkan Nama')
mengeSort(inlist)
for x in con(inlist,Daftar):
    print("=>",x.nama, x.kota, x.saku, x.nim)
```

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\MSI GF63\Documents\UMS\TUGAS SEMESTER 4\Praktikum Algoritma\MODUL 6\
nomor 1.py
Pengaturan Berdasarkan Nama
=> Ahmad Surakarta 250000 103
=> Budi Sragen 230000 101
=> Chanra Surakarta 235000 104
=> Deni Klaten 245000 107
=> Eka Boyolali 240000 105
=> Fandi Salatiga 250000 106
=> Galuh Wonogiri 245000 108
=> Hasan Karanganyar 270000 109
=> Ika Sukoharjo 240000 100
=> Janto Klaten 245000 109
=> Khalid Purwodadi 265000 110
>>>
```

- Soal 1.2

nomor 2.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/nomor ...

File Edit Format Run Options Window Help

```
#MODUL 6
#CINDI DILA APRILIANA_L200200106



class MhsTIF(object):
    def __init__(self,nama,nim,kota,saku):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.saku = saku


c0 = MhsTIF('Ika',100,'Sukoharjo', 240000)
c1 = MhsTIF('Budi',101,'Sragen', 230000)
c2 = MhsTIF('Ahmad',103,'Surakarta', 250000)
c3 = MhsTIF('Chanra',104,'Surakarta', 235000)
c4 = MhsTIF('Eka',105,'Boyolali', 240000)
c5 = MhsTIF('Fandi',106,'Salatiga', 250000)
c6 = MhsTIF('Deni',107,'Klaten', 245000)
c7 = MhsTIF('Galuh',108,'Wonogiri', 245000)
c8 = MhsTIF('Janto',109,'Klaten', 245000)
c9 = MhsTIF('Hasan',109,'Karanganyar', 270000)
c10 = MhsTIF('Khalid',110,'Purwodadi', 265000)

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


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

def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah = partisi(A, awal, akhir)
        quickSortBantu(A, titikBelah + 1, akhir)
        quickSortBantu(A, awal, titikBelah - 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 con(a, b):
    baru = []
    for x in range(len(a)):
        for y in range(len(a)):
            if a[x] == b[y].nama:
                baru.append(b[y])
    return baru

inlist = []
for x in Daftar:
    inlist.append(x.nama)

print('Pengurutan berdasarkan nama')
quickSort(inlist)
for x in con(ls, Daftar):
    print("=>", x.nama, x.kota, x.saku, x.nim)
```

Ln: 38 Col: 0

IDLE Shell 3.10.2

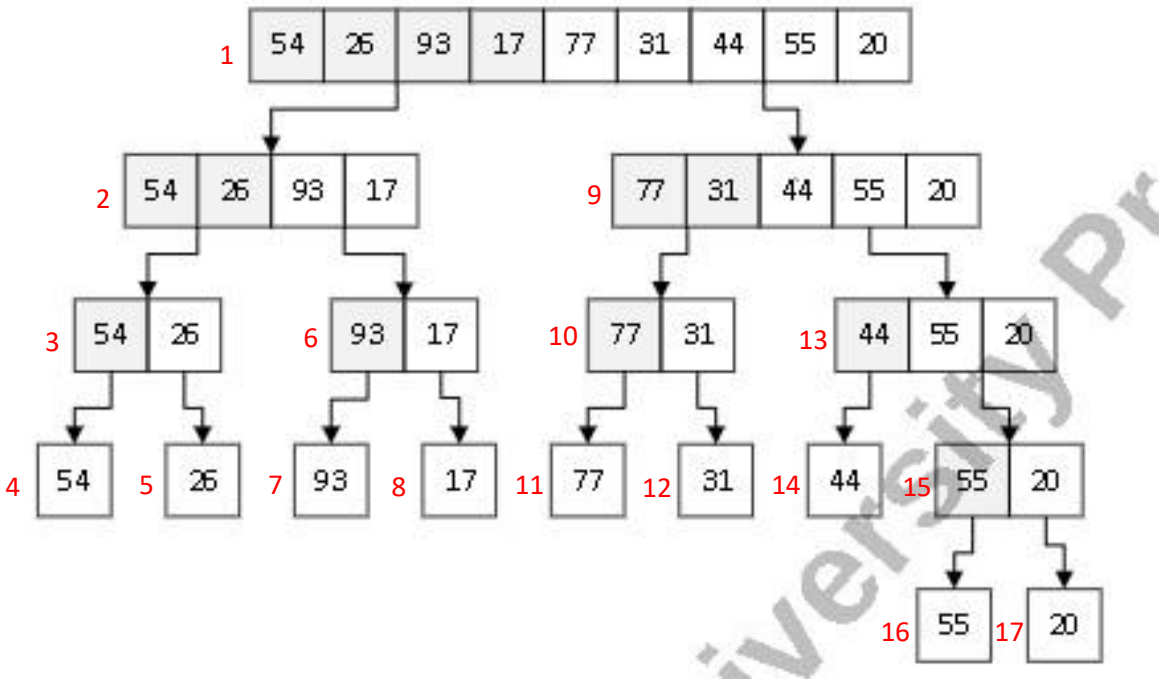
File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/
nomor 2.py
Pengurutan berdasarkan nama
=> Ahmad Surakarta 250000 103
=> Budi Sragen 230000 101
=> Chanra Surakarta 235000 104
=> Deni Klaten 245000 107
=> Eka Boyolali 240000 105
=> Fandi Salatiga 250000 106
=> Galuh Wonogiri 245000 108
=> Hasan Karanganyar 270000 109
=> Ika Sukoharjo 240000 100
=> Janto Klaten 245000 109
=> Khalid Purwodadi 265000 110
>>>
```

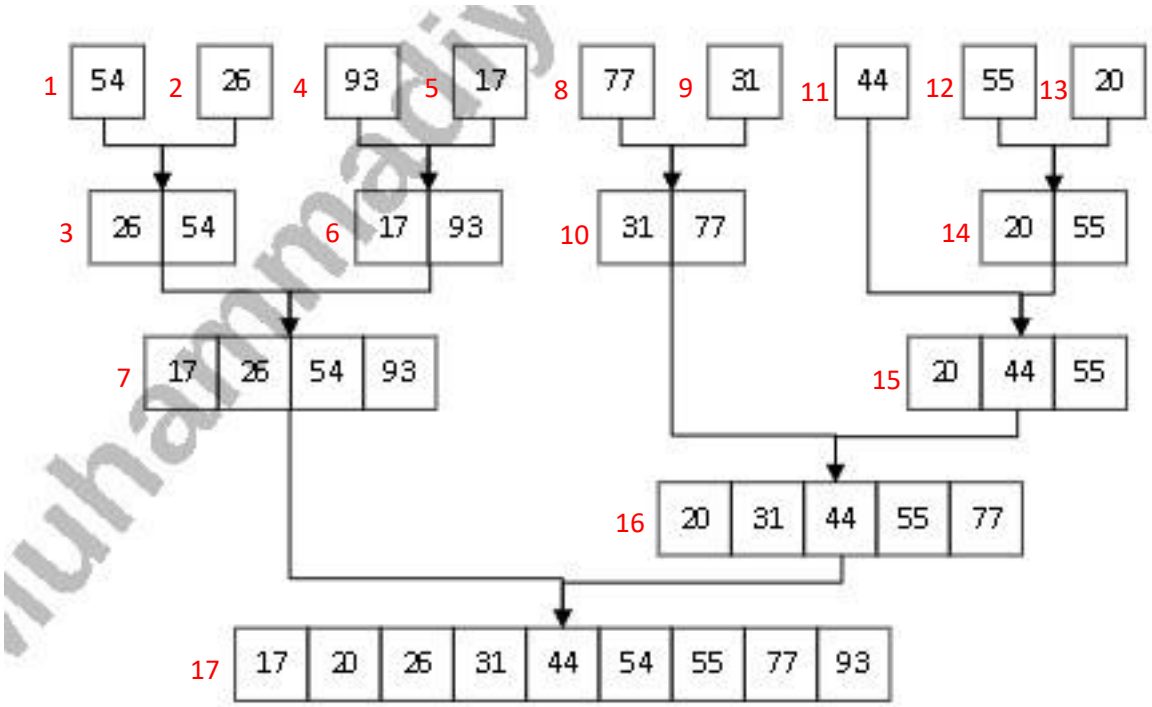
Ln: 3 Col: 0

2. Memakai bolpen merah atau biru, tandai dan beri nomer urut eksekusi proses pada Gambar 6.1 dan 6.2, dengan mengacu pada output di halaman 59.

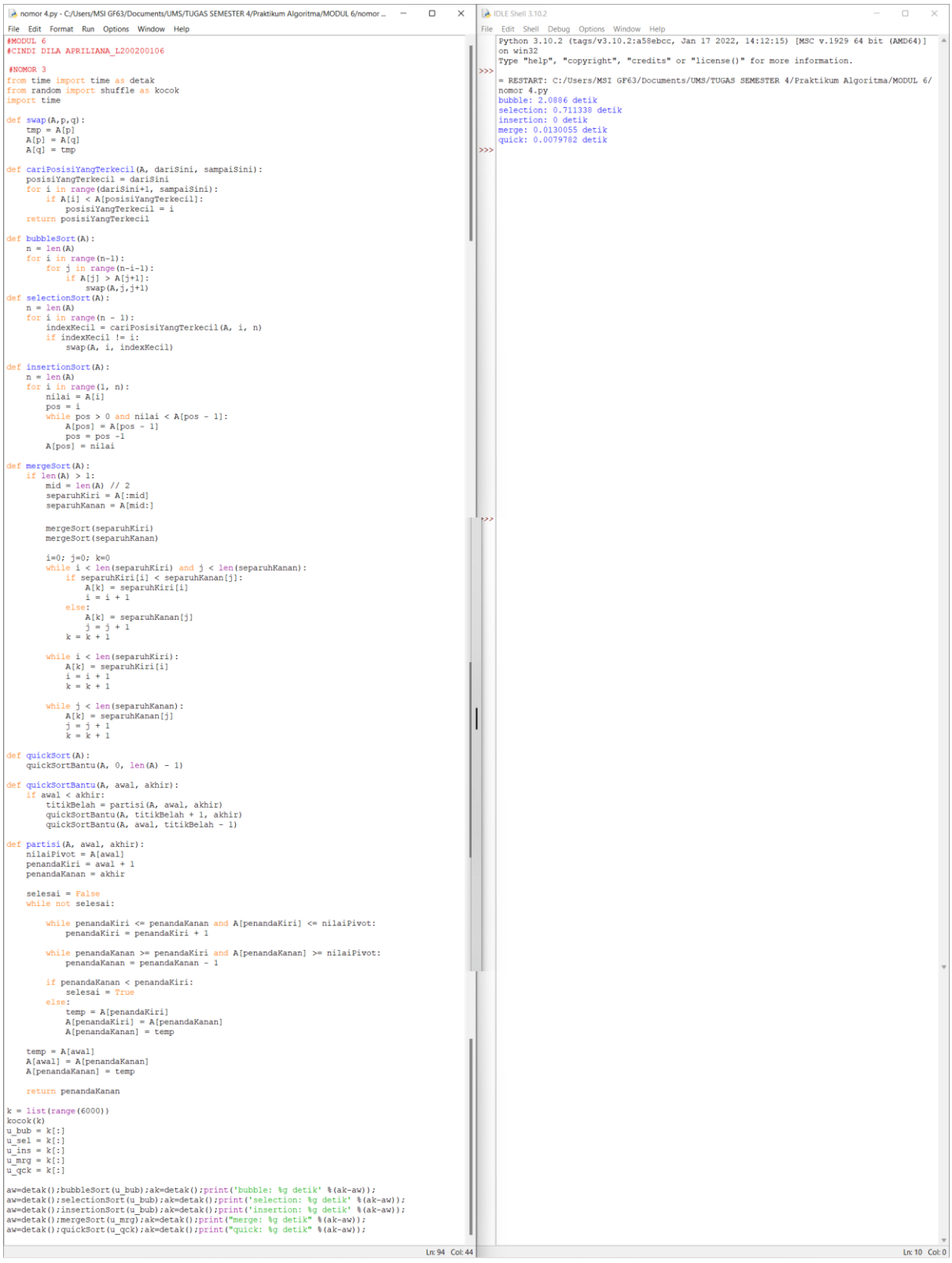
- Merge Sort




- Quick Sort



3. Uji kecepatan. Ujilah mergeSort dan quickSort di atas (bersama metode sort yang kamu pelajari sebelumnya) dengan kode di bawah ini.



nomor 4.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/nomor ...

File Edit Format Run Options Window Help

```
#MODUL 6
#CINDI DILA APRILIANA_L200200106

#NOMOR 3
from time import time as detak
from random import shuffle as kocok
import time

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)
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, titikBelah + 1, akhir)
        quickSortBantu(A, awal, titikBelah - 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

k = list(range(6000))
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_bub);ak=detak();print('selection: %g detik' %(ak-aw));
aw=detak();insertionSort(u_bub);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));
```

Ln: 94 Col: 44

IDLE Shell 3.10.2

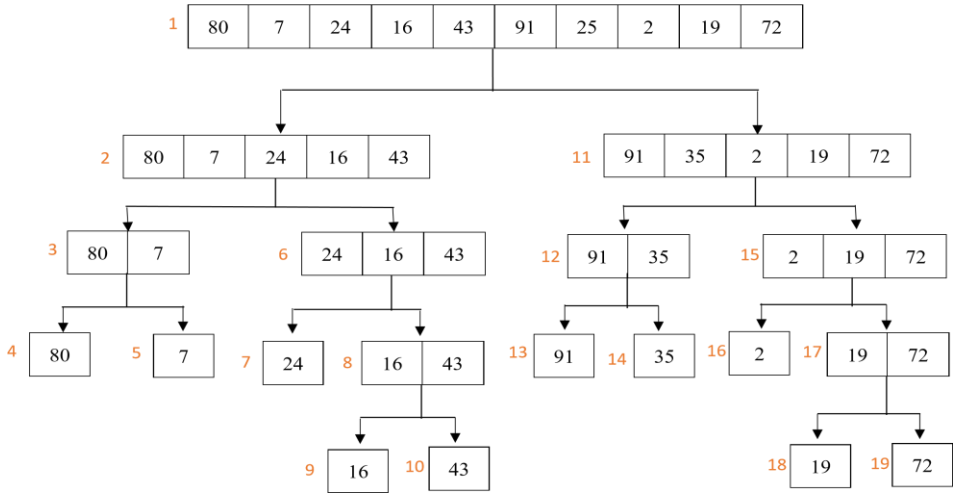
File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/
nomor 4.py
bubble: 2.0886 detik
selection: 0.711338 detik
insertion: 0 detik
merge: 0.0130055 detik
quick: 0.0079782 detik
>>>
```

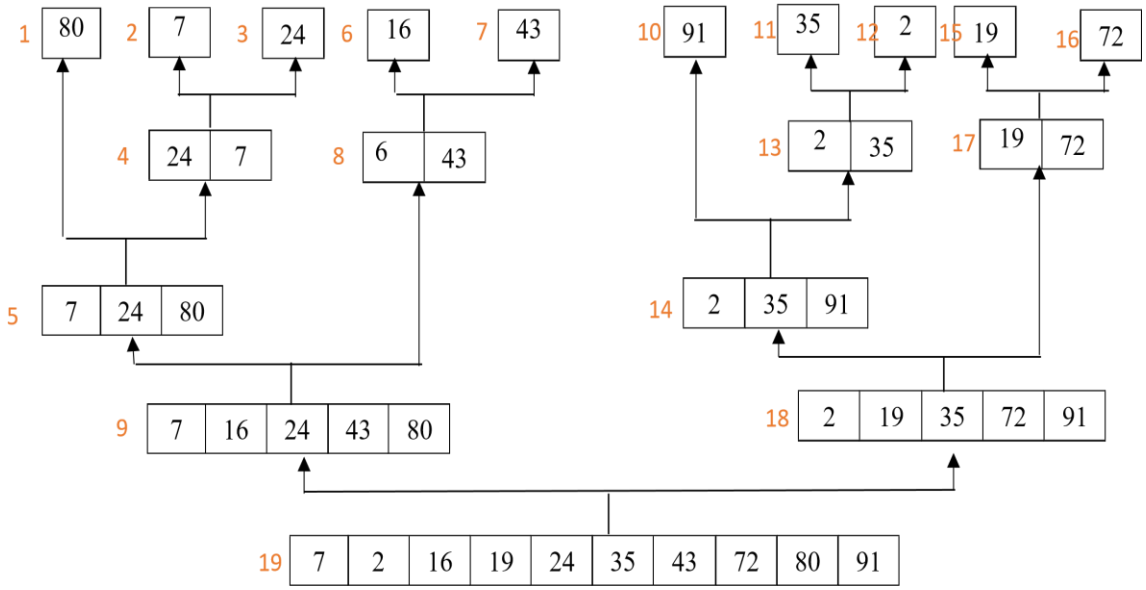
Ln: 10 Col: 0

4. Diberikan list L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72], gambarlah trace peng urutan10 untuk algoritma

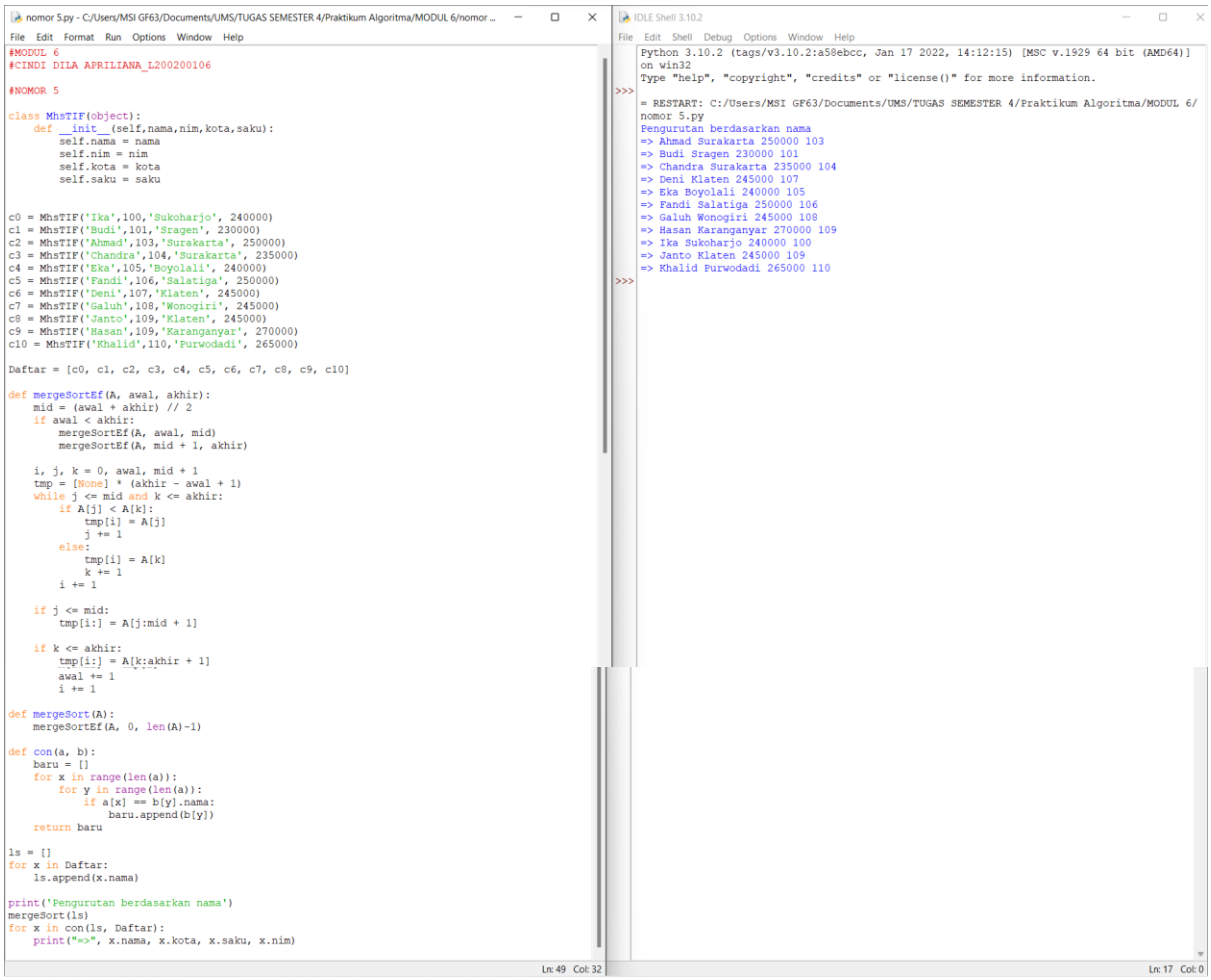
a) merge sort




b) quick sort



5. Tingkatkan efisiensi program mergeSort dengan tidak memakai operator slice (seperti A[:mid] dan A[mid:]), dan lalu mem-pass index awal dan index akhir bersama listnya saat kita memanggil mergeSort secara rekursif. Kamu akan perlu memisah fungsi mergeSort itu menjadi beberapa fungsi, mirip halnya dengan apa yang dilakukan algoritma quick sorth



nomor 5.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/nomor ...

File Edit Format Run Options Window Help

```
#MODUL 6
#CINDI DILA APRILIANA_L200200106

#NOMOR 5

class MhsTIF(object):
    def __init__(self,nama,nim,kota,saku):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.saku = saku


c0 = MhsTIF('Ika',100,'Sukoharjo', 240000)
c1 = MhsTIF('Budi',101,'Sragen', 230000)
c2 = MhsTIF('Ahmad',103,'Surakarta', 250000)
c3 = MhsTIF('Chandra',104,'Surakarta', 235000)
c4 = MhsTIF('Eka',105,'Boyolali', 240000)
c5 = MhsTIF('Fandi',106,'Salatiga', 250000)
c6 = MhsTIF('Deni',107,'Klaten', 245000)
c7 = MhsTIF('Galuh',108,'Wonogiri', 245000)
c8 = MhsTIF('Janto',109,'Klaten', 245000)
c9 = MhsTIF('Hasan',109,'Karanganyar', 270000)
c10 = MhsTIF('Khalid',110,'Purwodadi', 265000)

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

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

    i, j, k = 0, awal, mid + 1
    tmp = [None] * (akhir - awal + 1)
    while j <= mid and k <= akhir:
        if A[j] < A[k]:
            tmp[i] = A[j]
            j += 1
        else:
            tmp[i] = A[k]
            k += 1
        i += 1

    if j <= mid:
        tmp[i:] = A[j:mid + 1]

    if k <= akhir:
        tmp[i:] = A[k:akhir + 1]
        awal += 1
        i += 1

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

def con(a, b):
    baru = []
    for x in range(len(a)):
        for y in range(len(a)):
            if a[x] == b[y].nama:
                baru.append(b[y])
    return baru

ls = []
for x in Daftar:
    ls.append(x.nama)

print('Pengurutan berdasarkan nama')
mergeSort(ls)
for x in con(ls, Daftar):
    print("=>", x.nama, x.kota, x.saku, x.nim)
```

Ln: 49 Col: 32

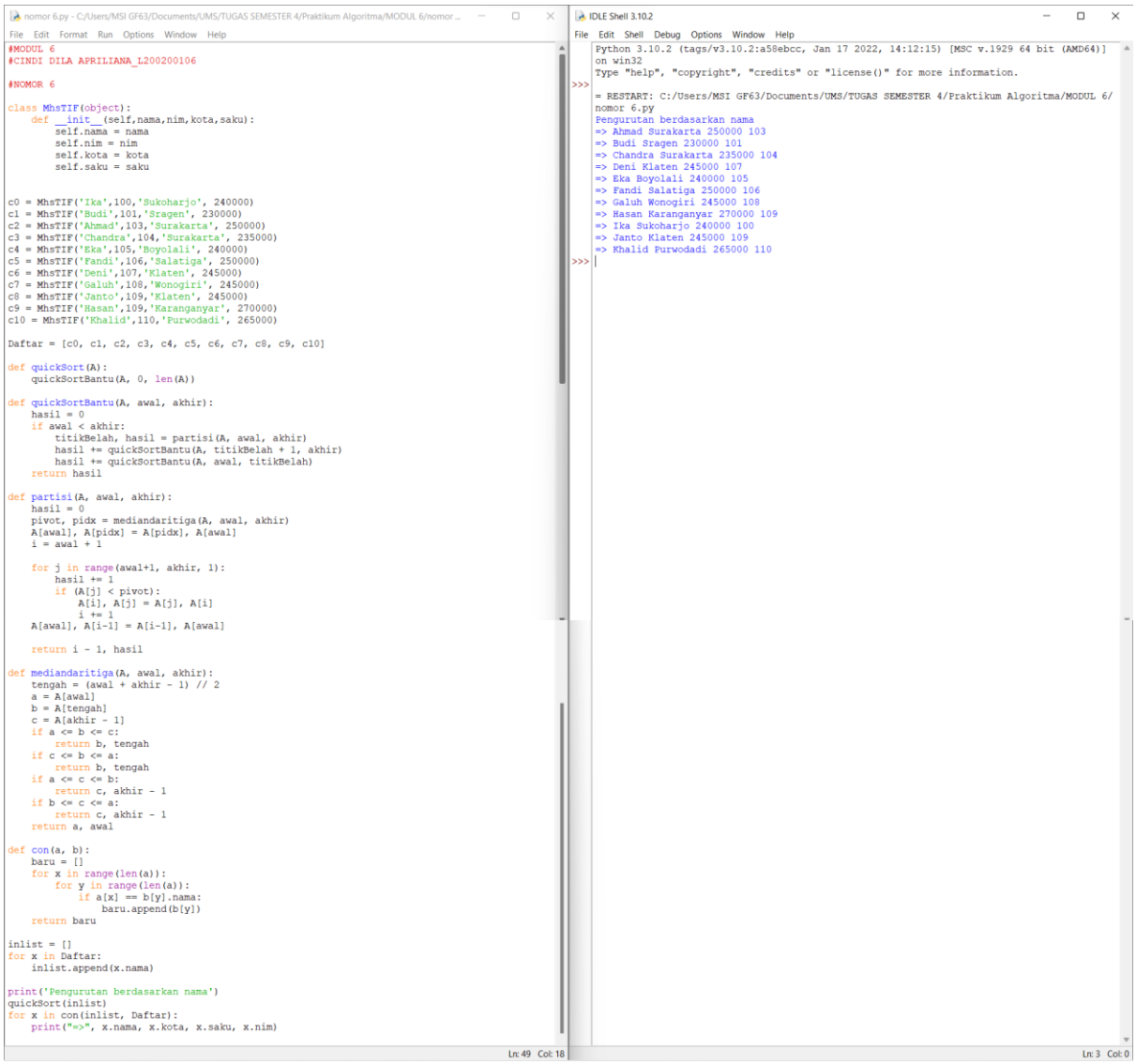
IDLE Shell 3.10.2

File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/
nomor 5.py
Pengurutan berdasarkan nama
=> Ahmad Surakarta 250000 103
=> Budi Sragen 230000 101
=> Chandra Surakarta 235000 104
=> Deni Klaten 245000 107
=> Eka Boyolali 240000 105
=> Fandi Salatiga 250000 106
=> Galuh Wonogiri 245000 108
=> Hasan Karanganyar 270000 109
=> Ika Sukoharjo 240000 100
=> Janto Klaten 245000 109
=> Khalid Purwodadi 265000 110
>>>
```

Ln: 17 Col: 0

6. Apakah kita bisa meningkatkan efisiensi program quickSort dengan memakai metode median-dari-tiga untuk memilih pivotnya? Ubahlah kodenya dan ujilah.



nomor 6.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/nomor ...

File Edit Format Run Options Window Help

```
#MODUL 6
#CINDI DILA APRILIANA_L200200106

#NOMOR 6

class MhsTIF(object):
    def __init__(self,nama,nim,kota,saku):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.saku = saku


c0 = MhsTIF('Ika',100,'Sukoharjo', 240000)
c1 = MhsTIF('Budi',101,'Sragen', 230000)
c2 = MhsTIF('Ahmad',103,'Surakarta', 250000)
c3 = MhsTIF('Chandra',104,'Surakarta', 235000)
c4 = MhsTIF('Eka',105,'Boyolali', 240000)
c5 = MhsTIF('Fandi',106,'Salatiga', 250000)
c6 = MhsTIF('Deni',107,'Klaten', 245000)
c7 = MhsTIF('Galuh',108,'Wonogiri', 245000)
c8 = MhsTIF('Janto',109,'Klaten', 245000)
c9 = MhsTIF('Hasan',109,'Karanganyar', 270000)
c10 = MhsTIF('Khalid',110,'Purwodadi', 265000)

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

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

def quickSortBantu(A, awal, akhir):
    hasil = 0
    if awal < akhir:
        titikBelah, hasil = partisi(A, awal, akhir)
        hasil += quickSortBantu(A, titikBelah + 1, akhir)
        hasil += quickSortBantu(A, awal, titikBelah)
    return hasil

def partisi(A, awal, akhir):
    hasil = 0
    pivot, pidx = mediandaritiga(A, awal, akhir)
    A[awal], A[pidx] = A[pidx], A[awal]
    i = awal + 1

    for j in range(awal+1, akhir, 1):
        hasil += 1
        if (A[j] < pivot):
            A[i], A[j] = A[j], A[i]
            i += 1
    A[awal], A[i-1] = A[i-1], A[awal]

    return i - 1, hasil

def mediandaritiga(A, awal, akhir):
    tengah = (awal + akhir - 1) // 2
    a = A[awal]
    b = A[tengah]
    c = A[akhir - 1]
    if a <= b <= c:
        return b, tengah
    if c <= b <= a:
        return b, tengah
    if a <= c <= b:
        return c, akhir - 1
    if b <= c <= a:
        return c, akhir - 1
    return a, awal

def con(a, b):
    baru = []
    for x in range(len(a)):
        for y in range(len(a)):
            if a[x] == b[y].nama:
                baru.append(b[y])
    return baru

inlist = []
for x in Daftar:
    inlist.append(x.nama)

print('Pengurutan berdasarkan nama')
quickSort(inlist)
for x in con(inlist, Daftar):
    print("=>", x.nama, x.kota, x.saku, x.nim)
```

Ln: 49 Col: 18

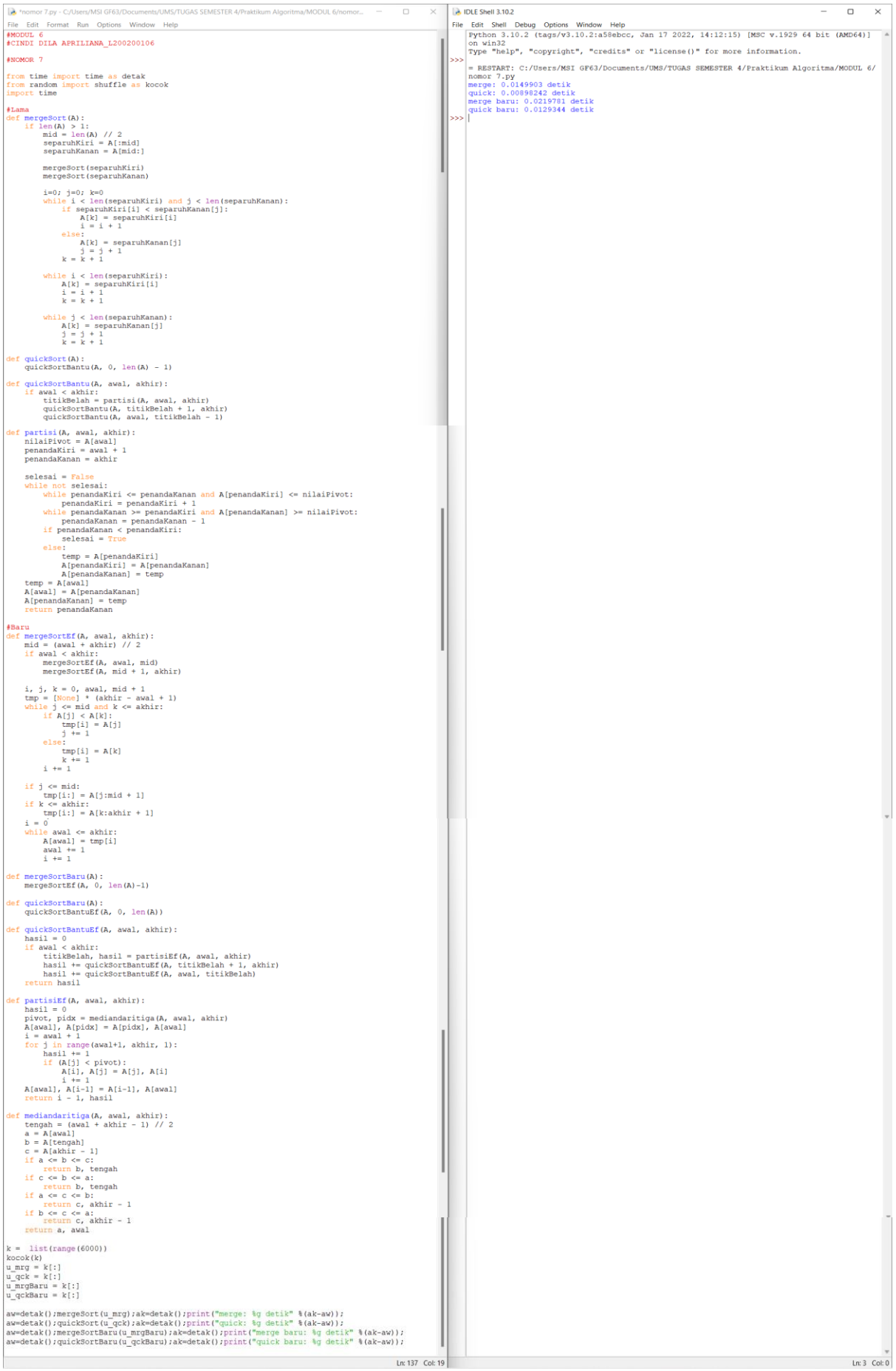
IDLE Shell 3.10.2

File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/
nomor 6.py
Pengurutan berdasarkan nama
=> Ahmad Surakarta 250000 103
=> Budi Sragen 230000 101
=> Chandra Surakarta 235000 104
=> Deni Klaten 245000 107
=> Eka Boyolali 240000 105
=> Fandi Salatiga 250000 106
=> Galuh Wonogiri 245000 108
=> Hasan Karanganyar 270000 109
=> Ika Sukoharjo 240000 100
=> Janto Klaten 245000 109
=> Khalid Purwodadi 265000 110
>>>
```

Ln: 3 Col: 0

7. Uji-kecepatan keduanya dan perbandingkan juga dengan kode awalnya.



*nomor 7.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/nomor...

File Edit Format Run Options Window Help

```
#MODUL 6
#CINDI DILA APRILIANA_L200200106

#NOMOR 7

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

#Lama
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, titikBelah + 1, akhir)
        quickSortBantu(A, awal, titikBelah - 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

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

    i, j, k = 0, awal, mid + 1
    tmp = [None] * (akhir - awal + 1)
    while j <= mid and k <= akhir:
        if A[j] < A[k]:
            tmp[i] = A[j]
            j += 1
        else:
            tmp[i] = A[k]
            k += 1
        i += 1

    if j <= mid:
        tmp[i:] = A[j:mid + 1]
    if k <= akhir:
        tmp[i:] = A[k:akhir + 1]
    i = 0
    while awal <= akhir:
        A[awal] = tmp[i]
        awal += 1
        i += 1

def mergeSortBaru(A):
    mergeSortEf(A, 0, len(A)-1)

def quickSortBaru(A):
    quickSortBantuEf(A, 0, len(A))

def quickSortBantuEf(A, awal, akhir):
    hasil = 0
    if awal < akhir:
        titikBelah, hasil = partisiEf(A, awal, akhir)
        hasil += quickSortBantuEf(A, titikBelah + 1, akhir)
        hasil += quickSortBantuEf(A, awal, titikBelah)
    return hasil

def partisiEf(A, awal, akhir):
    hasil = 0
    pivot, pidx = mediandaritiga(A, awal, akhir)
    A[awal], A[pidx] = A[pidx], A[awal]
    i = awal + 1
    for j in range(awal+1, akhir, 1):
        hasil += 1
        if (A[j] < pivot):
            A[i], A[j] = A[j], A[i]
            i += 1
    A[awal], A[i-1] = A[i-1], A[awal]
    return i - 1, hasil

def mediandaritiga(A, awal, akhir):
    tengah = (awal + akhir - 1) // 2
    a = A[awal]
    b = A[tengah]
    c = A[akhir - 1]
    if a <= b <= c:
        return b, tengah
    if c <= b <= a:
        return b, tengah
    if a <= c <= b:
        return c, akhir - 1
    if b <= c <= a:
        return c, akhir - 1
    return a, awal

k =  list(range(6000))
kocok(k)
u_mrg = k[:]
u_qck = k[:]
u_mrgBaru = k[:]
u_qckBaru = k[:]

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));
aw=detak();mergeSortBaru(u_mrgBaru);ak=detak();print("merge baru: %g detik" %(ak-aw));
aw=detak();quickSortBaru(u_qckBaru);ak=detak();print("quick baru: %g detik" %(ak-aw));
```

Ln: 137 Col: 19

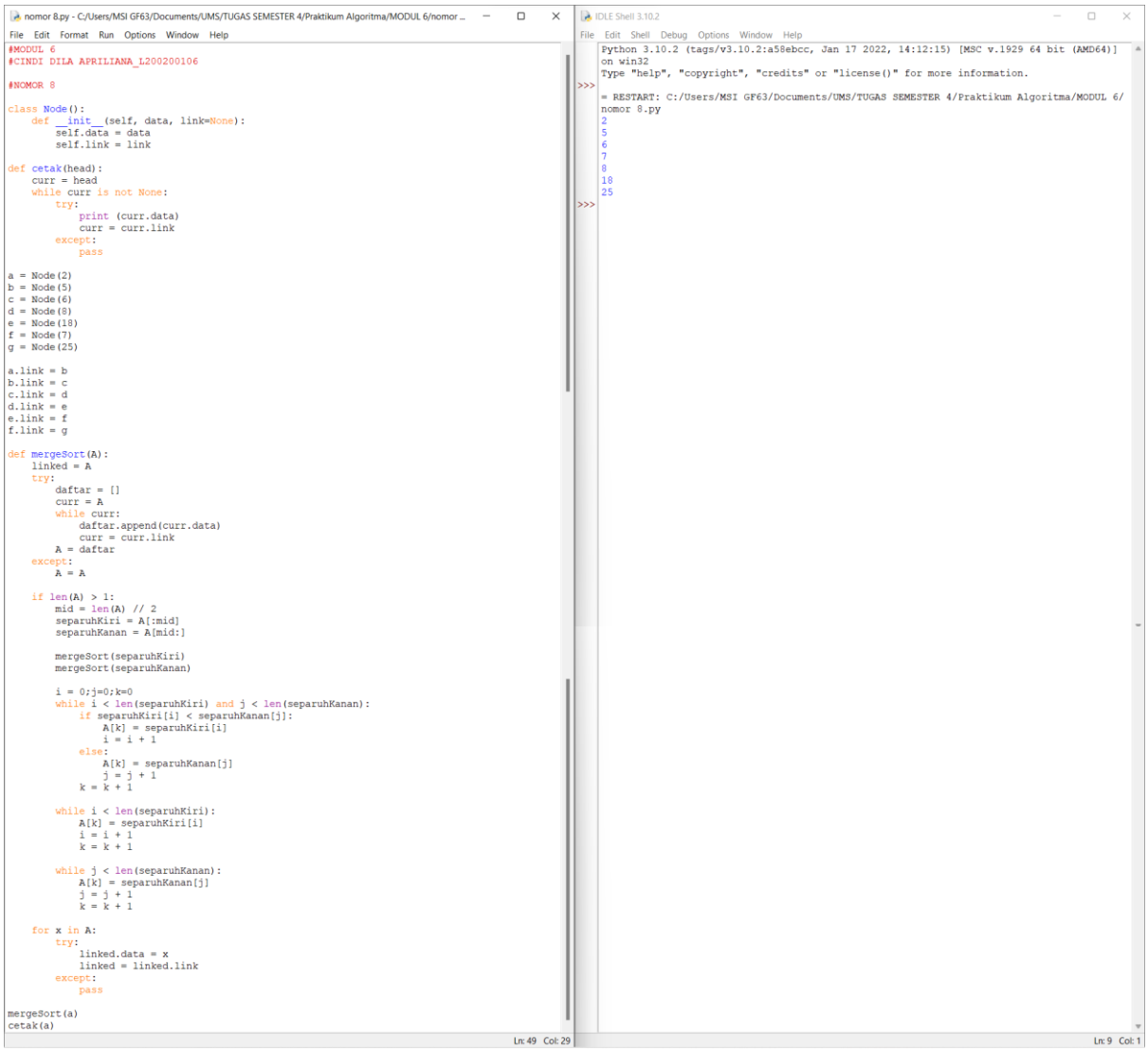
IDLE Shell 3.10.2

File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/
nomor 7.py
merge: 0.0149903 detik
quick: 0.00898242 detik
merge baru: 0.0219781 detik
quick baru: 0.0129344 detik
>>>
```

Ln: 3 Col: 0

8. Buatlah versi linked-list untuk program mergeSort di atas.



nomor 8.py - C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/nomor ...

File Edit Format Run Options Window Help

```
#MODUL 6
#CINDI DILA APRILIANA_L200200106

#NOMOR 8

class Node():
    def __init__(self, data, link=None):
        self.data = data
        self.link = link

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

a = Node(2)
b = Node(5)
c = Node(6)
d = Node(8)
e = Node(18)
f = Node(7)
g = Node(25)

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

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

    for x in A:
        try:
            linked.data = x
            linked = linked.link
        except:
            pass

mergeSort(a)
cetak(a)
```

Ln: 49 Col: 29

IDLE Shell 3.10.2

File Edit Shell Debug Options Window Help

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 6/
nomor 8.py
2
5
6
7
8
18
25
>>>
```

Ln: 9 Col: 1