

PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
MODUL 6



Nama : NICKY JULYATRIKA SARI

NIM : L200200101

PROGRAM STUDI
INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2021/2022

- 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.

a. mergeSort

```
nomer 1.py - D:/MATKUL SMT 4/Praktikum ASD/modul 6/nomer 1.py (3.8.6)
File Edit Format Run Options Window Help
#Nicky Julyatrika Sari NIM L200200101
#Modul 6

#Nomer 1
class MhsTIF(object):
    def __init__(self,nama,umur,kota,saku,nim):
        self.nama = nama
        self.umur = umur
        self.kota = kota
        self.saku = saku
        self.nim = nim

c0 = MhsTIF('Ika',10,'Sukoharjo', 240000, 101)
c1 = MhsTIF('Budi',19,'Sragen', 230000, 102)
c2 = MhsTIF('Ahmad',12,'Surakarta', 250000, 103)
c3 = MhsTIF('Chandra',17,'Surakarta', 235000, 104)
c4 = MhsTIF('Eka',13,'Boyolali', 240000, 105)
c5 = MhsTIF('Fandi',11,'Salatiga', 250000, 106)
c6 = MhsTIF('Deni',16,'Klaten', 245000, 107)
c7 = MhsTIF('Galuh',18,'Wonogiri', 245000, 108)
c8 = MhsTIF('Janto',20,'Klaten', 245000, 109)
c9 = MhsTIF('Hasan',14,'Karanganyar', 270000, 110)
c10 = MhsTIF('Khalid',15,'Purwodadi', 265000, 111)

#Lalu kita membuat daftar mahasiswa dalam bentuk list seperti ini:

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

```
nomer 1.py - D:/MATKUL SMT 4/Praktikum ASD/modul 6/nomer 1.py (3.8.6)
File Edit Format Run Options Window Help

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

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

print("Pengurutan Berdasarkan Nama Adalah Berikut")
mergeSort(ls)
for x in con(ls, Daftar):
    print("~~~", x.nama, x.umur, x.kota, x.saku, x.nim)
```

```
Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020,
Type "help", "copyright", "credits" or "license
>>>
===== RESTART: D:/MATKUL SMT 4/Praktikum
Pengurutan Berdasarkan Nama Adalah Berikut
~~ Ahmad 12 Surakarta 250000 103
~~ Budi 19 Sragen 230000 102
~~ Chandra 17 Surakarta 235000 104
~~ Deni 16 Klaten 245000 107
~~ Eka 13 Boyolali 240000 105
~~ Fandi 11 Salatiga 250000 106
~~ Galuh 18 Wonogiri 245000 108
~~ Hasan 14 Karanganyar 270000 110
~~ Ika 10 Sukoharjo 240000 101
~~ Janto 20 Klaten 245000 109
~~ Khalid 15 Purwodadi 265000 111
>>>
```

b. quickSort

```
*nomer1(2).py - D:/MATKUL SMT 4/Praktikum ASD/modul 6/nomer1(2).py (3.8.6)*
File Edit Format Run Options Window Help
#Nicky Julyatnika Sari NIM L200200101
#Modul 6

#Nomer 1b(quickSort)|
class MhsTIF(object):
    def __init__(self,nama,umur,kota,saku,nim):
        self.nama = nama
        self.umur = umur
        self.kota = kota
        self.saku = saku
        self.nim = nim

c0 = MhsTIF('Ika',10,'Sukoharjo', 240000, 101)
c1 = MhsTIF('Budi',19,'Sragen', 230000, 102)
c2 = MhsTIF('Ahmad',12,'Surakarta', 250000, 103)
c3 = MhsTIF('Chandra',17,'Surakarta', 235000, 104)
c4 = MhsTIF('Eka',13,'Boyolali', 240000, 105)
c5 = MhsTIF('Fandi',11,'Salatiga', 250000, 106)
c6 = MhsTIF('Deni',16,'Klaten', 245000, 107)
c7 = MhsTIF('Galuh',18,'Wonogiri', 245000, 108)
c8 = MhsTIF('Janto',20,'Klaten', 245000, 109)
c9 = MhsTIF('Hasan',14,'Karanganyar', 270000, 110)
c10 = MhsTIF('Khalid',15,'Purwodadi', 265000, 111)

#Lalu kita membuat daftar mahasiswa dalam bentuk list seperti ini:
Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]
```

```
*nomer1(2).py - D:/MATKUL SMT 4/Praktikum ASD/modul 6/nomer1(2).py (3.8.6)*
File Edit Format Run Options Window Help

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):
    new = []
    for x in range(len(a)):
        for y in range(len(b)):
            if a[x] == b[y].nama:
                new.append(b[y])
    return new

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

print('Pengurutan berdasarkan nama sebagai berikut : ')
quickSort(ls)
for x in con(ls, Daftar):
    print("%-10s" % x.nama, x.umur, x.kota, x.saku, x.nim)

```

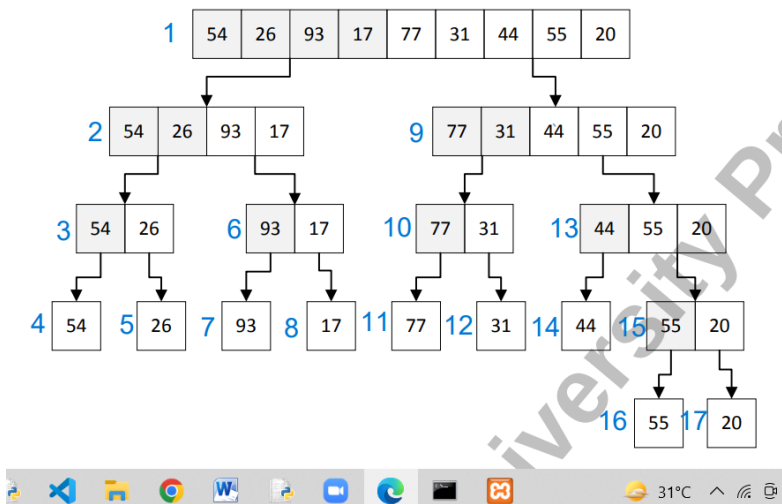
```

Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020,
Type "help", "copyright", "credits" or "license(
>>>
===== RESTART: D:/MATKUL SMT 4/Praktikum As
Pengurutan berdasarkan nama sebagai berikut :
~~ Ahmad 12 Surakarta 250000 103
~~ Budi 19 Sragen 230000 102
~~ Chandra 17 Surakarta 235000 104
~~ Deni 16 Klaten 245000 107
~~ Eka 13 Boyolali 240000 105
~~ Fandi 11 Salatiga 250000 106
~~ Galuh 18 Wonogiri 245000 108
~~ Hasan 14 Karanganyar 270000 110
~~ Ika 10 Sukoharjo 240000 101
~~ Janto 20 Klaten 245000 109
~~ Khalid 15 Purwodadi 265000 111
>>>

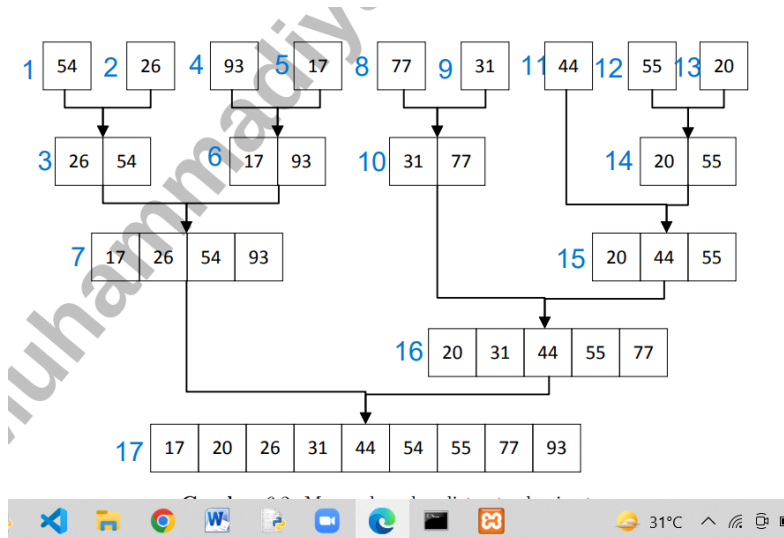
```

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.

a. mergeSort



b. quickSort



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

```

nomer3.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer3.py (3.8.6)
File Edit Format Run Options Window Help
#Nicky Julyatrika Sari NIM L200200101
#Modul 6

#Nomer 3

from time import time as detik
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)

nomer3.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer3.py (3.8.6)
File Edit Format Run Options Window Help
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:

```

nomer3.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer3.py (3.8.6)
File Edit Format Run Options Window Help

```

        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)

```

nomer3.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer3.py (3.8.6)
File Edit Format Run Options Window Help

```

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));

```

Python 3.8.6 Shell

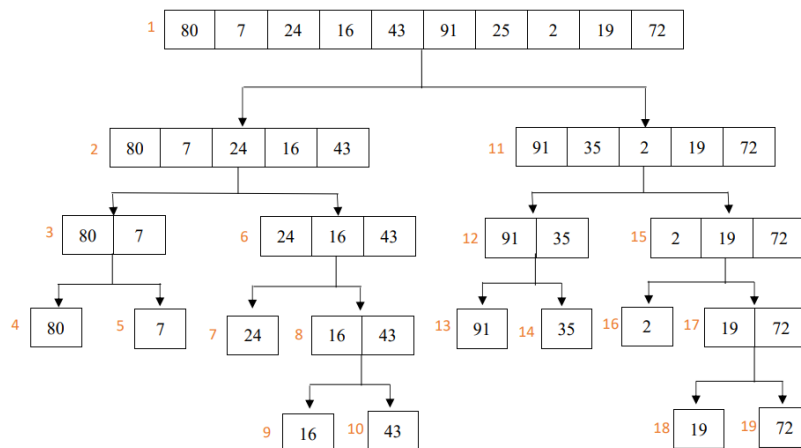
```

File Edit Shell Debug Options Window
Python 3.8.6 (tags/v3.8.6:db4552
Type "help", "copyright", "credi
>>>
===== RESTART: D:\MATKUL S
bubble: 7.16404 detik
selection: 2.18653 detik
insertion: 0 detik
merge: 0.0464032 detik
quick: 0.0312104 detik
>>> |

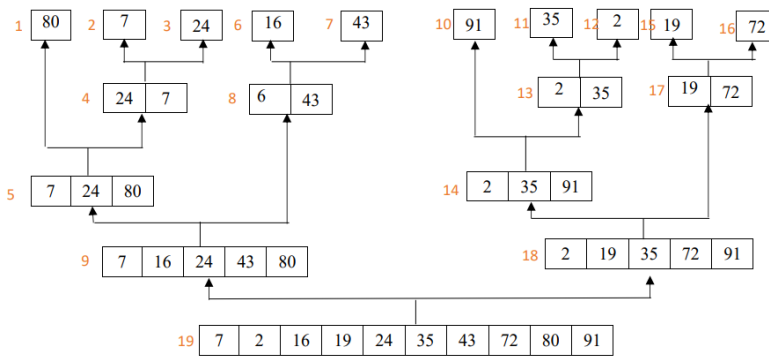
```

4. Diberikan list $L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72]$, gambarlah trace pengurutan10 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 sort.

nomer5.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer5.py (3.8.6)

File Edit Format Run Options Window Help

```
#Nicky Julyatrika Sari NIM L200200101
#Modul 6
```

```
#Nomer 5
class MhsTIF(object):
    def __init__(self,nama,umur,kota,saku,nim):
        self.nama = nama
        self.umur = umur
        self.kota = kota
        self.saku = saku
        self.nim = nim

c0 = MhsTIF('Ika',10,'Sukoharjo', 240000, 101)
c1 = MhsTIF('Budi',19,'Sragen', 230000, 102)
c2 = MhsTIF('Ahmad',12,'Surakarta', 250000, 103)
c3 = MhsTIF('Chandra',17,'Surakarta', 235000, 104)
c4 = MhsTIF('Eka',13,'Boyolali', 240000, 105)
c5 = MhsTIF('Fandi',11,'Salatiga', 250000, 106)
c6 = MhsTIF('Deni',16,'Klaten', 245000, 107)
c7 = MhsTIF('Galuh',18,'Wonogiri', 245000, 108)
c8 = MhsTIF('Janto',20,'Klaten', 245000, 109)
c9 = MhsTIF('Hasan',14,'Karanganyar', 270000, 110)
c10 = MhsTIF('Khalid',15,'Purwodadi', 265000, 111)

#Lalu kita membuat daftar mahasiswa dalam bentuk list seperti ini:
Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]
```

nomer5.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nor

File Edit Format Run Options Window Help

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

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

ls = []
for x in Daftar:
    ls.append(x.nama)
print('Pengurutan berdasarkan nama sebagai berikut :')
mergeSort(ls)
for x in con(ls, Daftar):
    print("~~", x.nama, x.umur, x.kota, x.saku, x.nim)
```



```

Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020,
Type "help", "copyright", "credits" or "license
>>>
===== RESTART: D:\MATKUL SMT 4\Praktikum
Pengurutan berdasarkan nama sebagai berikut :
~~ Ahmad 12 Surakarta 250000 103
~~ Budi 19 Sragen 230000 102
~~ Chandra 17 Surakarta 235000 104
~~ Deni 16 Klaten 245000 107
~~ Eka 13 Boyolali 240000 105
~~ Fandi 11 Salatiga 250000 106
~~ Galuh 18 Wonogiri 245000 108
~~ Hasan 14 Karanganyar 270000 110
~~ Ika 10 Sukoharjo 240000 101
~~ Janto 20 Klaten 245000 109
~~ Khalid 15 Purwodadi 265000 111
>>> |

```

6. Apakah kita bisa meningkatkan efisiensi program quickSort dengan memakai metode median-dari-tiga untuk memilih pivotnya? Ubahlah kodenya dan ujilah. 7

```

nomer6.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer6.py (3.8.6)
File Edit Format Run Options Window Help
#Nicky Julyatnika Sari NIM L200200101
#Modul 6

#Nomer 6
class MhsTIF(object):
    def __init__(self, nama, umur, kota, saku, nim):
        self.nama = nama
        self.umur = umur
        self.kota = kota
        self.saku = saku
        self.nim = nim

c0 = MhsTIF('Ika',10,'Sukoharjo', 240000, 101)
c1 = MhsTIF('Budi',19,'Sragen', 230000, 102)
c2 = MhsTIF('Ahmad',12,'Surakarta', 250000, 103)
c3 = MhsTIF('Chandra',17,'Surakarta', 235000, 104)
c4 = MhsTIF('Eka',13,'Boyolali', 240000, 105)
c5 = MhsTIF('Fandi',11,'Salatiga', 250000, 106)
c6 = MhsTIF('Deni',16,'Klaten', 245000, 107)
c7 = MhsTIF('Galuh',18,'Wonogiri', 245000, 108)
c8 = MhsTIF('Janto',20,'Klaten', 245000, 109)
c9 = MhsTIF('Hasan',14,'Karanganyar', 270000, 110)
c10 = MhsTIF('Khalid',15,'Purwodadi', 265000, 111)

#Lalu kita membuat daftar mahasiswa dalam bentuk list seperti ini:

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

```

```

nomer6.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer6.py (3.8.6)
File Edit Format Run Options Window Help
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 = mediandartiga(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):
    new = []
    for x in range(len(a)):
        for y in range(len(b)):
            if a[x] == b[y].nama:
                new.append(b[y])
    return new

ls = []
for x in Daftar:
    ls.append(x.nama)
print('Pengurutan berdasarkan nama sebagai berikut :')
quickSort(ls)
for x in con(ls, Daftar):
    print("~~", x.nama, x.umur, x.kota, x.saku, x.nim)

```

Python 3.8.6 Shell

File Edit Shell Debug Options Window Help

Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, Type "help", "copyright", "credits" or "license")

```

>>>
===== RESTART: D:\MATKUL SMT 4\Praktikum
Pengurutan berdasarkan nama sebagai berikut :
~~ Ahmad 12 Surakarta 250000 103
~~ Budi 19 Sragen 230000 102
~~ Chandra 17 Surakarta 235000 104
~~ Deni 16 Klaten 245000 107
~~ Eka 13 Boyolali 240000 105
~~ Fandi 11 Salatiga 250000 106
~~ Galuh 18 Wonogiri 245000 108
~~ Hasan 14 Karanganyar 270000 110
~~ Ika 10 Sukoharjo 240000 101
~~ Janto 20 Klaten 245000 109
~~ Khalid 15 Purwodadi 265000 111
>>> |

```

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

nomer7.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer7.py (3.8.6)

File Edit Format Run Options Window Help

```

#Nicky Julyatrika Sari NIM L200200101
#Modul 6

#Nomer 7

from time import time as detik
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

```

nomer7.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer7.py (3.8.6)

File Edit Format Run Options Window Help

```
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
```

nomer7.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer7.py (3.8.6)

File Edit Format Run Options Window Help

```
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
```

nomer7.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer7.py (3.8.6)

File Edit Format Run Options Window Help

```
#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
```

```
nomer7.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer7.py (3.8.6)
File Edit Format Run Options Window Help

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));

Python 3.8.6 Shell
File Edit Shell Debug Options Window
Python 3.8.6 (tags/v3.8.6:db45c64, Sep 24 2020)
Type "help", "copyright", "credits()" or "quit()" to quit.
>>>
===== RESTART: D:\MATKUL
merge: 0.0620487 detik
quick: 0.046386 detik
merge baru: 0.0624819 detik
quick baru: 0.0307291 detik
>>>
```

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

```

*nomer8.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer8.py (3.8.6)*
File Edit Format Run Options Window Help
#Nicky Julyatrika Sari NIM L200200101
#Modul 6

#Nomer 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

```

```

*nomer8.py - D:\MATKUL SMT 4\Praktikum ASD\modul 6\nomer8.py (3.8.6)*
File Edit Format Run Options Window Help
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)

```

```
>>>  
-----  
2  
5  
6  
7  
8  
18  
25  
>>>
```

