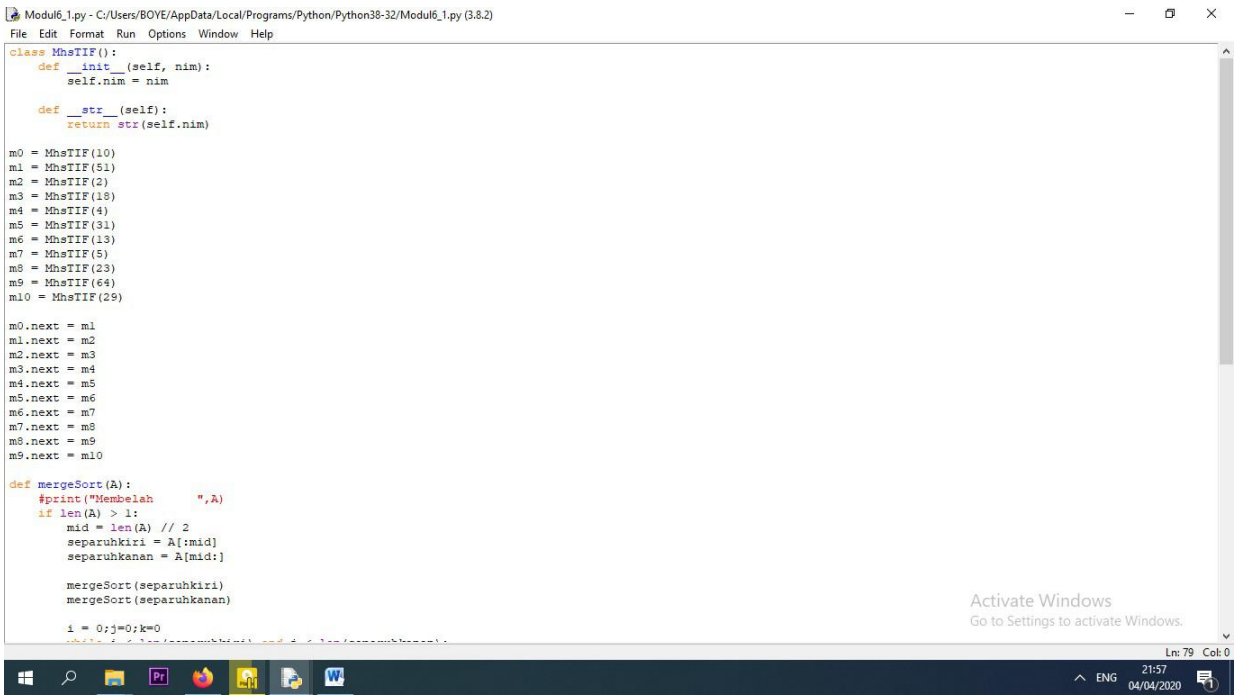
Afen Orleno S

L200180208

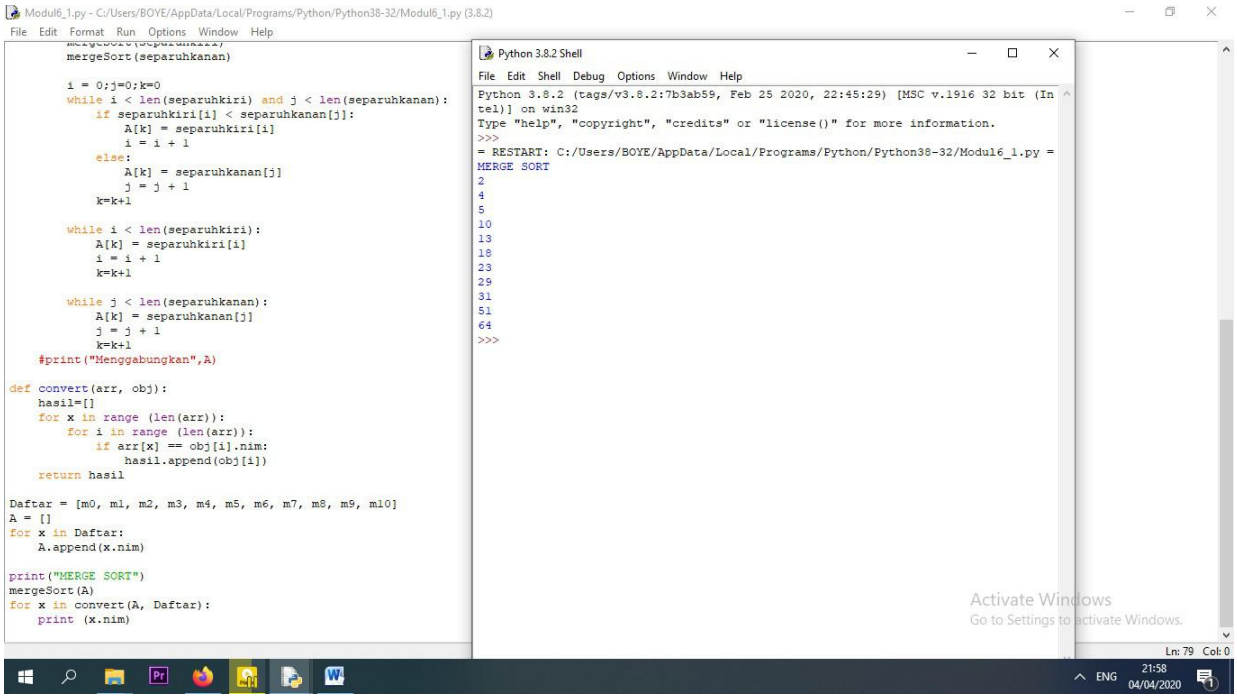
H

MODUL 6

1. Ubah kode mergeSort dan quickSort agar bias mengurutkan list yang berisi object-object mhsTIF
    - MergeSort



HASIL:


Modul6_1.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/Modul6_1.py (3.8.2)
File Edit Format Run Options Window Help
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
#print("Menggabungkan",A)
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
Daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
A = []
for x in Daftar:
A.append(x.nim)
print("MERGE SORT")
mergeSort(A)
for x in convert(A, Daftar):
print (x.nim)
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/Modul6_1.py =
MERGE SORT
2
4
5
10
13
18
23
29
31
51
64
>>>
Activate Windows
Go to Settings to activate Windows.
Ln: 79 Col: 0
ENG
21:58
04/04/2020

- quickSort

#2.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#2.py (3.8.2)

File Edit Format Run Options Window Help

```
class MhsTIF():
    def __init__(self, nim):
        self.nim = nim

    def __str__(self):
        return str(self.nim)

m0 = MhsTIF(10)
m1 = MhsTIF(51)
m2 = MhsTIF(2)
m3 = MhsTIF(18)
m4 = MhsTIF(4)
m5 = MhsTIF(31)
m6 = MhsTIF(13)
m7 = MhsTIF(5)
m8 = MhsTIF(23)
m9 = MhsTIF(64)
m10 = MhsTIF(29)

m0.next = m1
m1.next = m2
m2.next = m3
m3.next = m4
m4.next = m5
m5.next = m6
m6.next = m7
m7.next = m8
m8.next = m9
m9.next = m10

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
```


Ln: 85 Col: 0

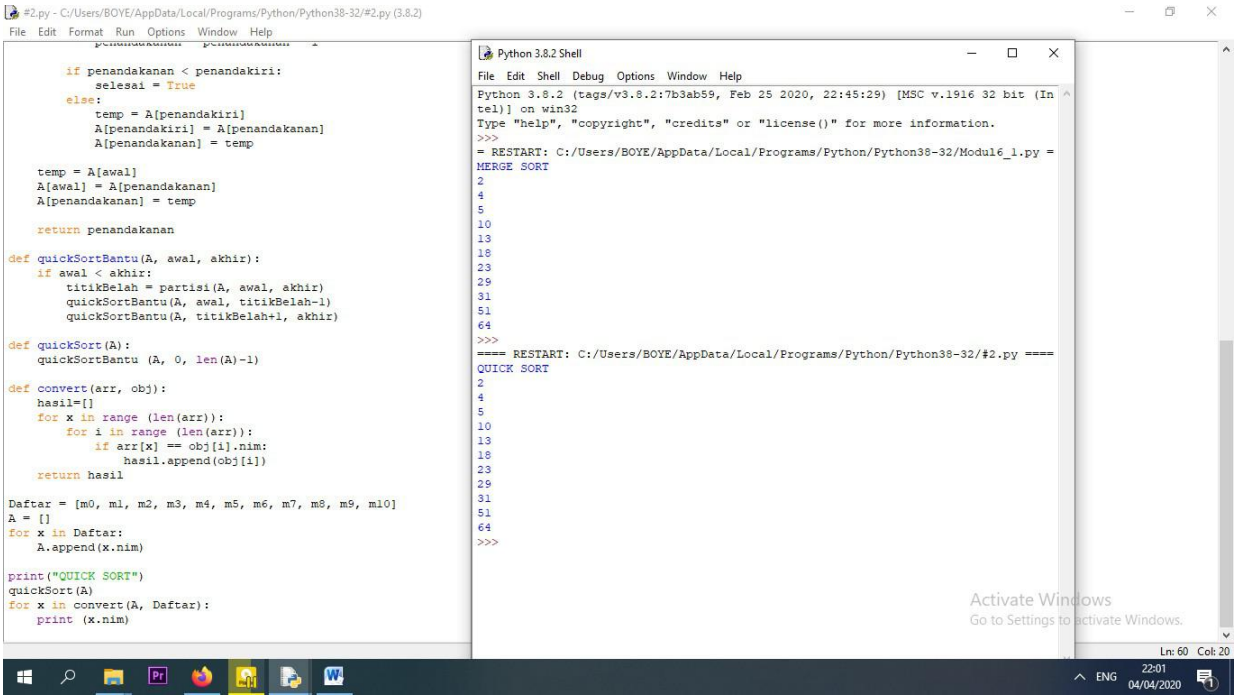
#2.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#2.py (3.8.2)

File Edit Format Run Options Window Help

```
    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 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

Daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
A = []
```


Ln: 60 Col: 20

HASIL:


#2.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#2.py (3.8.2)
File Edit Format Run Options Window Help
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
Daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
A = []
for x in Daftar:
A.append(x.nim)
print("QUICK SORT")
quickSort(A)
for x in convert(A, Daftar):
print (x.nim)
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/Modul6_1.py =
MERGE SORT
2
4
5
10
13
18
23
29
31
51
64
>>>
==== RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#2.py ====
QUICK SORT
2
4
5
10
13
18
23
29
31
51
64
>>>
Activate Windows
Go to Settings to activate Windows.
Ln: 60 Col: 20
ENG
22:01
04/04/2020

2. Menulis pakai bolpen merah dan biru
3. Uji kecepatan

#3.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#3.py (3.8.2)

File Edit Format Run Options Window Help

```
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(S):
    n = len(S)
    for i in range (n-1):
        for j in range (n-i-1):
            if S[j] > S[j+1]:
                swap(S,j,j+1)
    return S

def selectionSort(S):
    n = len(S)
    for i in range(n-1):
        indexKecil = cariPosisiYangTerkecil(S, i, n)
        if indexKecil != i:
            swap(S, i, indexKecil)
    return S

def insertionSort(S):
    n = len(S)
    for i in range(1, n):
        nilai = S[i]
        pos = i
        while pos > 0 and nilai < S[pos -1]:
            S[pos] = S[pos-1]
            pos = pos - 1
        S[pos] = nilai
```


Ln: 16 Col: 0

#3.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#3.py (3.8.2)

File Edit Format Run Options Window Help

```
        S[pos] = nilai
    return S

def mergeSort(A):
    #print("Membelah      ",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
    #print("Menggabungkan",A)

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

    penandakiri = awal + 1
    penandakanan = akhir

    selesai = False
```


Ln: 16 Col: 0

#3.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#3.py (3.8.2)

File Edit Format Run Options Window Help

```
    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)

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

print (bubbleSort(daftar))
print (selectionSort(daftar))
print (insertionSort(daftar))
```


Ln: 16 Col: 0

#3.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#3.py (3.8.2)

File Edit Format Run Options Window Help

```
            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)

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

print (bubbleSort(daftar))
print (selectionSort(daftar))
print (insertionSort(daftar))
mergeSort(daftar)
print (daftar)
quickSort(daftar)
print (daftar)

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));
```

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
>>>
= RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/Modul6_1.py =
MERGE SORT
2
4
5
10
13
18
23
29
31
51
64
>>>
==== RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#2.py ====
QUICK SORT
2
4
5
10
13
18
23
29
31
51
64
>>>
==== RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#3.py ====
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
bubble: 13.9267 detik
selection: 6.41604 detik
insertion: 6.09276 detik
merge: 0.0817814 detik
```


Ln: 43 Col: 4

Col: 0

HASIL:

```
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
bubble: 4.29523 detik
selection: 1.75247 detik
```

4. Diberikan list L
5. Tingkatkan efisiensi mergeSort dengan tidak menggunakan operator Slice dan lalu mem-pass indek awal dan index akhir bersama list-nya saat kita memanggil mergeSort secara rekursif.

HASIL:

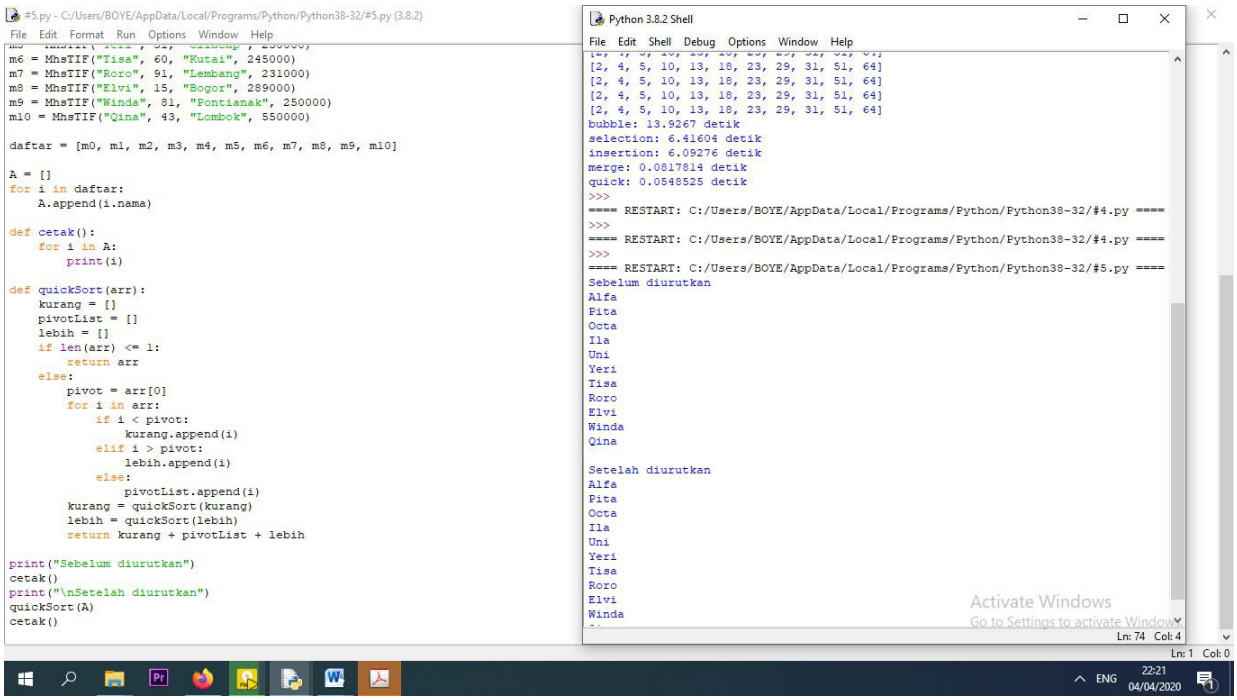
```
>>> cetak(daftar)
Alfa, NIM 76. Tinggal di Banyuwangi. Uang saku Rp. 249000 tiap bulannya.
Pita, NIM 53. Tinggal di Purwokerto. Uang saku Rp. 234000 tiap bulannya.
Octa, NIM 37. Tinggal di Purworejo. Uang saku Rp. 220000 tiap bulannya.
Ila, NIM 49. Tinggal di Surakarta. Uang saku Rp. 232000 tiap bulannya.
Uni, NIM 46. Tinggal di Demak. Uang saku Rp. 300000 tiap bulannya.
Yeri, NIM 31. Tinggal di Cilacap. Uang saku Rp. 250000 tiap bulannya.
Tisa, NIM 60. Tinggal di Kutai. Uang saku Rp. 245000 tiap bulannya.
Roro, NIM 91. Tinggal di Lembang. Uang saku Rp. 231000 tiap bulannya.
Elvi, NIM 15. Tinggal di Bogor. Uang saku Rp. 289000 tiap bulannya.
Winda, NIM 81. Tinggal di Pontianak. Uang saku Rp. 250000 tiap bulannya.
Qina, NIM 43. Tinggal di Lombok. Uang saku Rp. 550000 tiap bulannya.
>>> mergeSort(daftar)
>>> cetak(daftar)
Octa, NIM 37. Tinggal di Purworejo. Uang saku Rp. 220000 tiap bulannya.
Roro, NIM 91. Tinggal di Lembang. Uang saku Rp. 231000 tiap bulannya.
Ila, NIM 49. Tinggal di Surakarta. Uang saku Rp. 232000 tiap bulannya.
Pita, NIM 53. Tinggal di Purwokerto. Uang saku Rp. 234000 tiap bulannya.
Tisa, NIM 60. Tinggal di Kutai. Uang saku Rp. 245000 tiap bulannya.
Alfa, NIM 76. Tinggal di Banyuwangi. Uang saku Rp. 249000 tiap bulannya.
Winda, NIM 81. Tinggal di Pontianak. Uang saku Rp. 250000 tiap bulannya.
Yeri, NIM 31. Tinggal di Cilacap. Uang saku Rp. 250000 tiap bulannya.
Elvi, NIM 15. Tinggal di Bogor. Uang saku Rp. 289000 tiap bulannya.
Uni, NIM 46. Tinggal di Demak. Uang saku Rp. 300000 tiap bulannya.
Qina, NIM 43. Tinggal di Lombok. Uang saku Rp. 550000 tiap bulannya.
>>>
```

6. Meningkatkan efisiensi program quicksort dengan memakai metode median daritiga untuk memilih pivot

```
class MhsTIF():
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.us = us

    def __str__(self):
        s = self.nama +', NIM '+str(self.nim)\
            +'. Tinggal di '+ self.kota \
            +'. Uang saku Rp. '+ str(self.us)\
            +' tiap bulannya.'
        return s

    def ambilNama(self):
        return self.nama
    def ambilNim(self):
        return self.nim
    def ambilUangSaku(self):
        return self.us

m0 = MhsTIF("Alfa", 76, "Banyuwangi", 249000)
m1 = MhsTIF("Pita", 53, "Purwokerto", 234000)
m2 = MhsTIF("Octa", 37, "Purworejo", 220000)
m3 = MhsTIF("Ila", 49, "Surakarta", 232000)
m4 = MhsTIF("Uni", 46, "Demak", 300000)
m5 = MhsTIF("Yeri", 31, "Cilacap", 250000)
m6 = MhsTIF("Tisa", 60, "Kutai", 245000)
m7 = MhsTIF("Roro", 91, "Lembang", 231000)
m8 = MhsTIF("Elvi", 15, "Bogor", 289000)
m9 = MhsTIF("Winda", 81, "Pontianak", 250000)
m10 = MhsTIF("Qina", 43, "Lombok", 550000)

daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]

A = []
for i in daftar:
    A.append(i.nama)
```

```
def cetak():
    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 < pivot:
                kurang.append(i)
            elif i > pivot:
                lebih.append(i)
            else:
                pivotList.append(i)
        kurang = quickSort(kurang)
        lebih = quickSort(lebih)
        return kurang + pivotList + lebih

print("Sebelum diurutkan")
cetak()
print("\nSetelah diurutkan")
quickSort(A)
cetak()
```

HASIL:



7. Uji kecepatan keduanya dan perbandingkan juga dgn kode awalnya

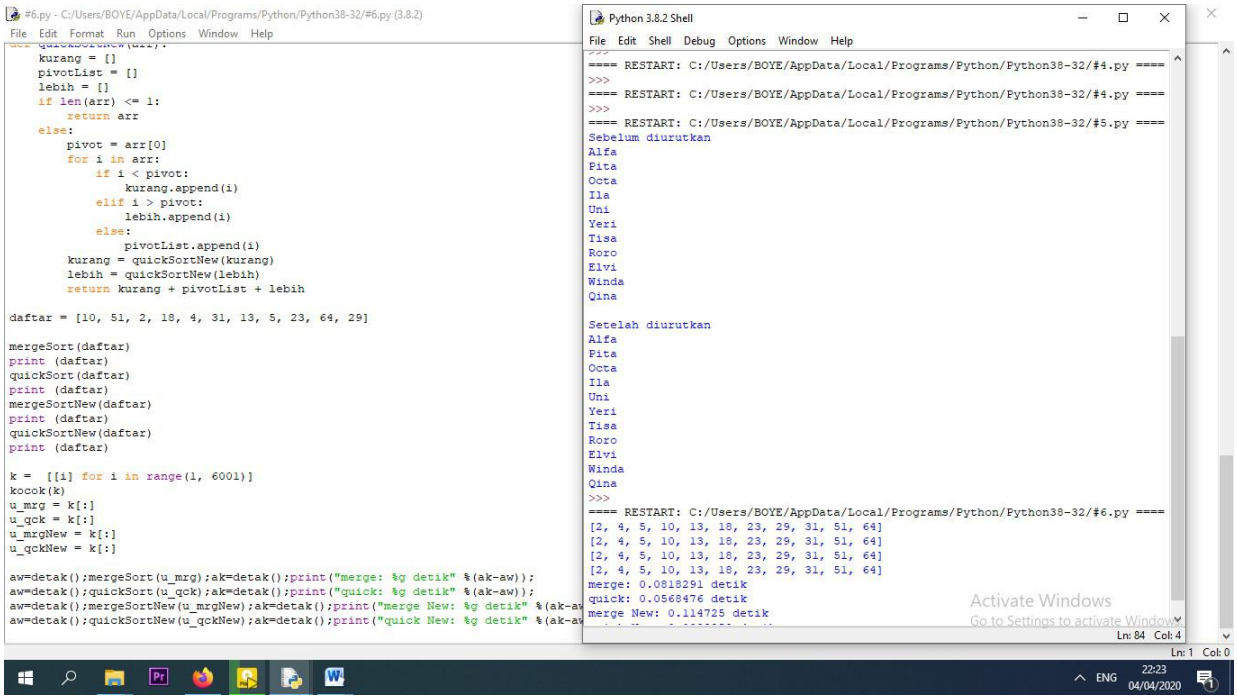
```
from time import time as detak
from random import shuffle as kocok
import time

def mergeSort(A):
    #print("Membelah      ",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
    #print("Menggabungkan",A)

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+l
    amp = [wove{ * akhiz - awal + 1} w-i
    e f <= mid a-d 1 <= alchiz:
        if A[f] < A[l]:
            zmp{a] = A{f]
            f += l

            0n>p[a]=A[1]
            l += l
        a += 1


        t:rap[a:]=A[f°:mid+1]

     Lt l <= akhiz:
        tmp{a:; = All:akhir+1;

    a = 0
    wicule awal aldhiz:
        A[awal{ := tmp[a
        awal += l
        a += l

def merge5ortNew A}:
    merge5ortZlA, 0, lenlA}-l)

def quickSozzNew aiz}:
    Azang = {}
    pivotLisB =
    [{ lebih = [{
    iI len{azi) <=l:
        ret iz azr

        pivn t: = a z x [ 0 ]
        :fo   i L r. a x z:
            z:f i < pivn t: :
                mix ang . append { i }

                1 ebih . append i }

                piwotLis£.append(i)
            kurang = quick5ortNew{kurang} lebih
            = quick5ortNew{lebih)
            ret r- kurang + pivotList + lebih
```

Qal'caz = }10 , 51, 2, 18, 9, 31, J3, 5, 23, 69, 29]

```
mergeSort(daftar)

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

k = [}i] for i iz range{1,6001}]

u_mrg = k[:]

u_mrgNew = k[:]

aw=detak{}:mergeSort{u_mrg}:ak=detak{};print{'merge: &g detik' 4{ak-aw}}:
aw=detak{}:quickSort{u_qck}:ak=detak{};print{'quick: 8g detik' 8{ak-aw}}:
aw=detak{}:mergeSortNew{u_mrgNew}:ak=detak{}:print{'merge New: &g detik' 8{ak-aw}}:
aw=detak{}:quickSortNew{u_qckNew):ak=detak{}:print{'quick New: 4g aetik' &{ak-aw}}:
```

HASIL:


#6.py - C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#6.py (3.8.2)
File Edit Format Run Options Window Help
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("merge: %g detik" %(ak-aw));
aw=detak();quickSort(u_qck);ak=detak();print("quick: %g detik" %(ak-aw));
aw=detak();mergeSortNew(u_mrgNew);ak=detak();print("merge New: %g detik" %(ak-a
aw=detak();quickSortNew(u_qckNew);ak=detak();print("quick New: %g detik" %(ak-a
Ln: 1 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
==== RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#4.py ====
>>>
==== RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#4.py ====
>>>
==== RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#5.py ====
Sebelum diurutkan
Alfa
Pita
Octa
Ila
Uni
Yeri
Tisa
Roro
Elvi
Winda
Qina
Setelah diurutkan
Alfa
Pita
Octa
Ila
Uni
Yeri
Tisa
Roro
Elvi
Winda
Qina
>>>
==== RESTART: C:/Users/BOYE/AppData/Local/Programs/Python/Python38-32/#6.py ====
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
merge: 0.0818291 detik
quick: 0.0568476 detik
merge New: 0.114725 detik
Activate Windows
Go to Settings to activate Windows.
Ln: 84 Col: 4
ENG 22:23 04/04/2020

8. Buat versi linked list untuk program mergeSort di atas

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

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

a = Node(1)
b = Node(3)
c = Node(5)
d = Node(7)
e = Node(2)
f = Node(4)
g = Node(6)

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

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

mergeSortLL(a)
cetak(a)
```

HASIL: