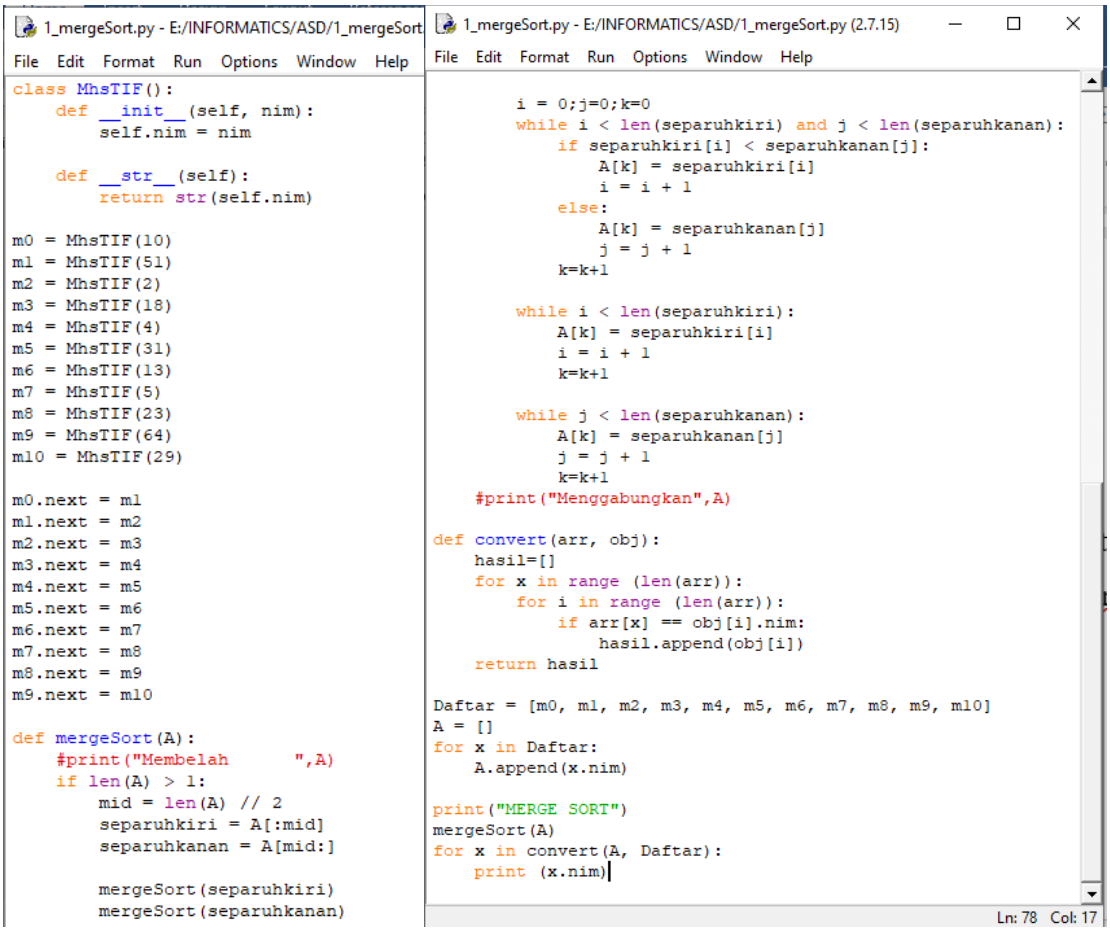
Nama : Aviza Ayuni Wulan

NIM : L200180187 / G

Praktikum Algostruk dan Struktur Data Modul 6

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



1_mergeSort.py - E:/INFORMATICS/ASD/1_mergeSort

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 mergeSort(A):
    #print("Membelah      ",A)
    if len(A) > 1:
        mid = len(A) // 2
        separuhkiri = A[:mid]
        separuhkanan = A[mid:]

        mergeSort(separuhkiri)
        mergeSort(separuhkanan)
```

1_mergeSort.py - E:/INFORMATICS/ASD/1_mergeSort.py (2.7.15)

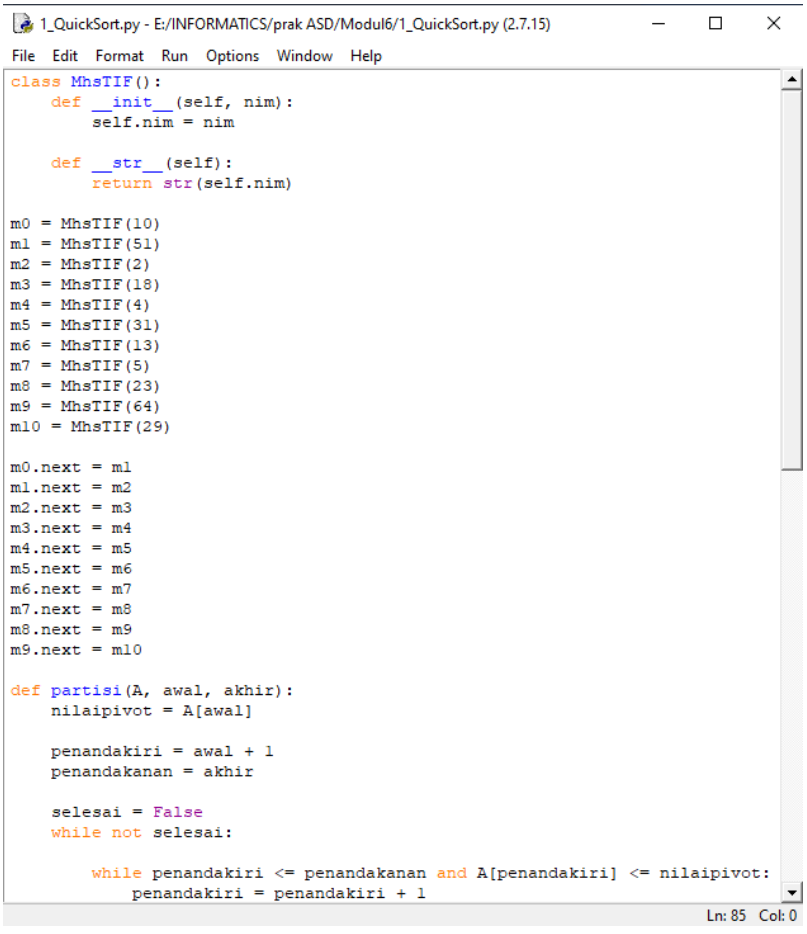
File Edit Format Run Options Window Help

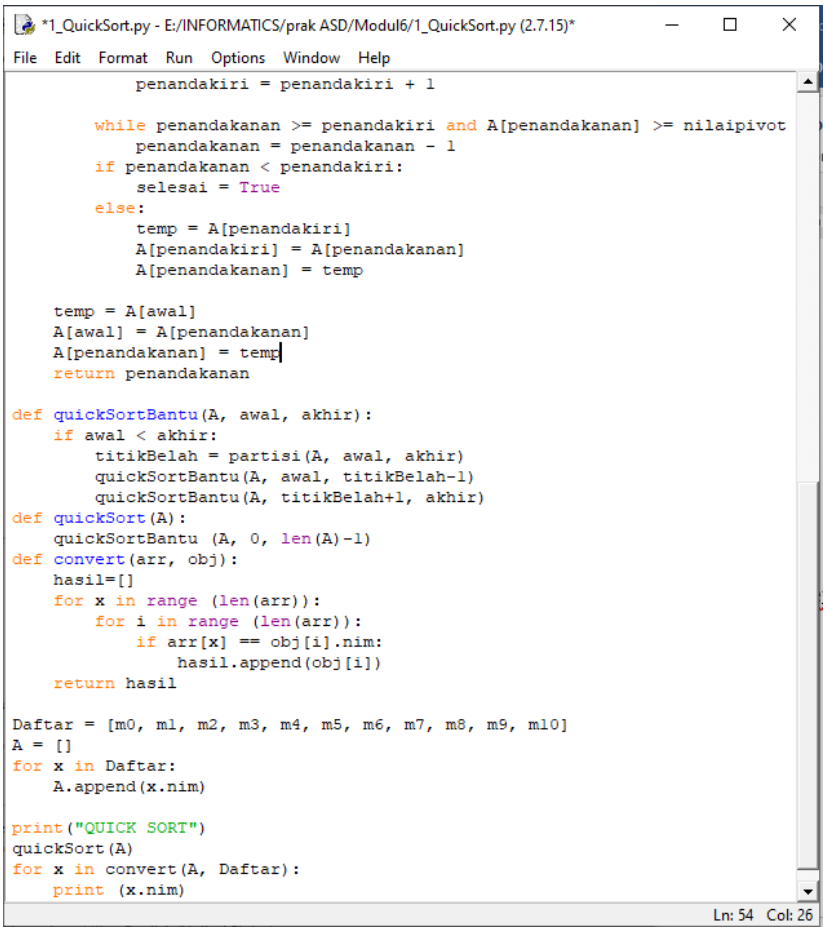
```
        i = 0;j=0;k=0
        while i < len(separuhkiri) and j < len(separuhkanan):
            if separuhkiri[i] < separuhkanan[j]:
                A[k] = separuhkiri[i]
                i = i + 1
            else:
                A[k] = separuhkanan[j]
                j = j + 1
            k=k+1

        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)
```

Ln: 78 Col: 17

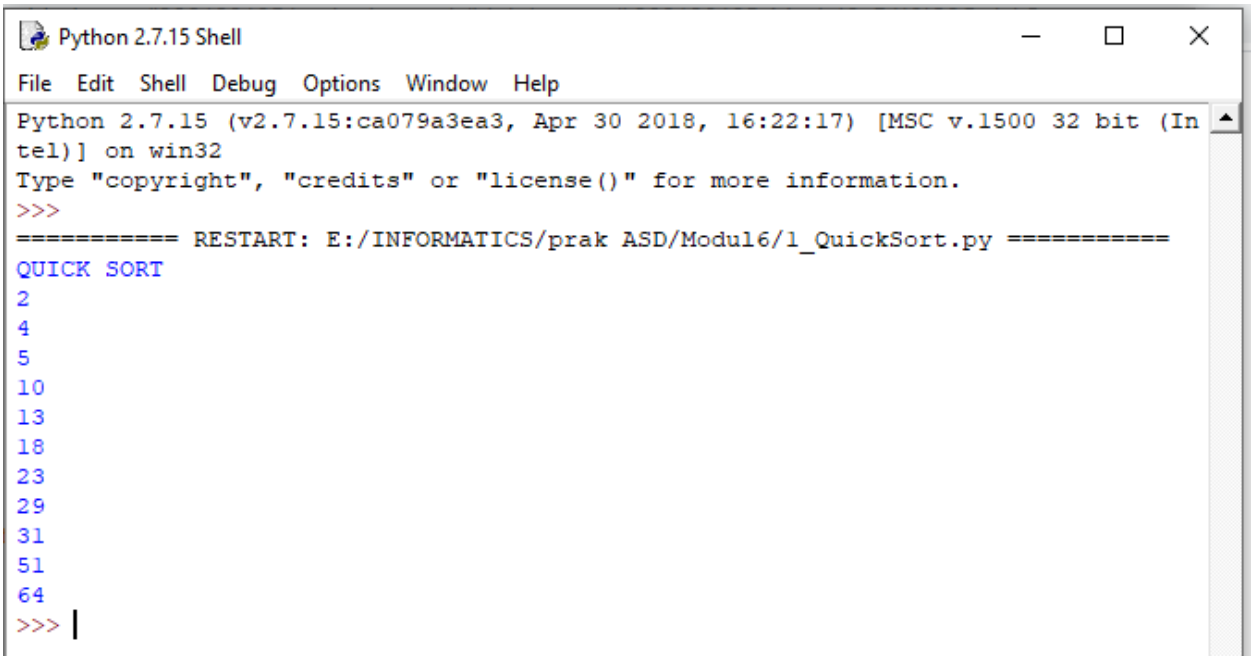
Hasil:

- QuickSort

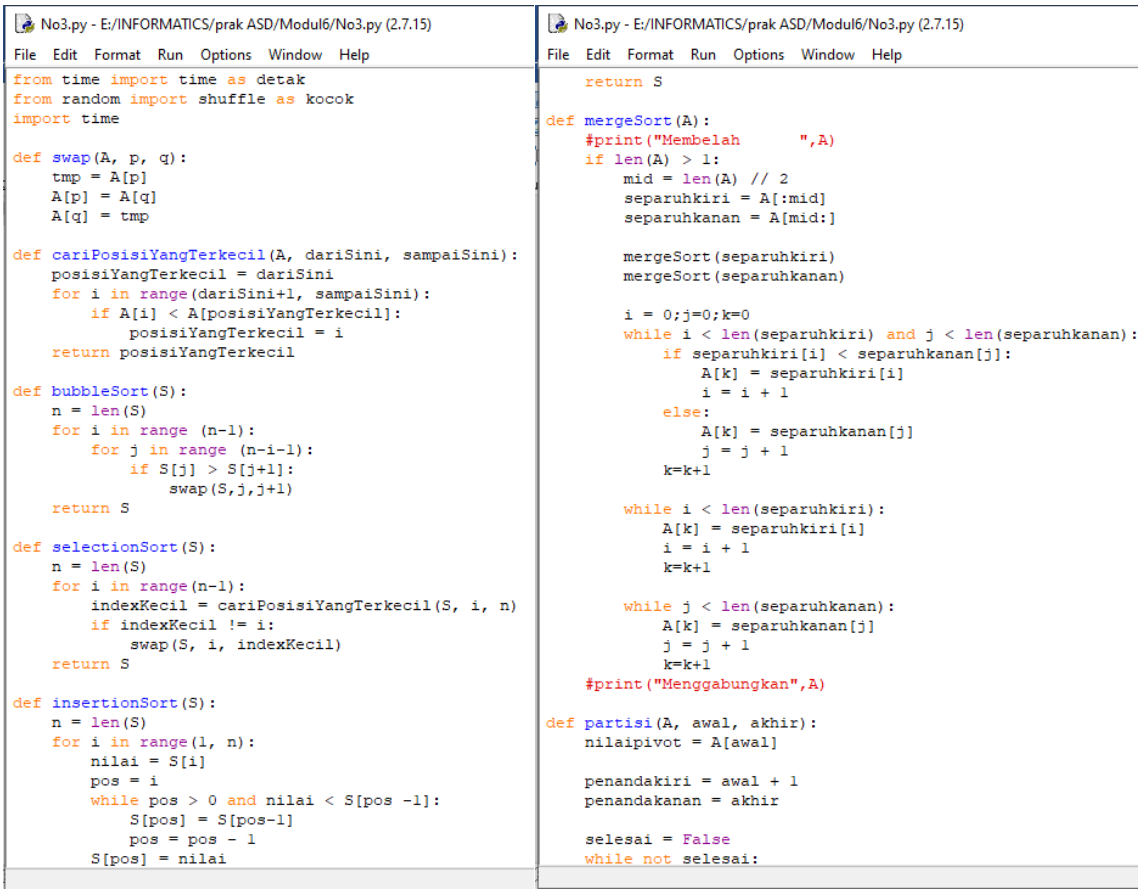



1_QuickSort.py - E:/INFORMATICS/prak ASD/Modul6/1_QuickSort.py (2.7.15)
File Edit Format Run Options Window Help
penandakiri = penandakiri + 1
while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot
penandakanan = penandakanan - 1
if penandakanan < penandakiri:
selesai = True
else:
temp = 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 = []
for x in Daftar:
A.append(x.nim)
print("QUICK SORT")
quickSort(A)
for x in convert(A, Daftar):
print (x.nim)
Ln: 54 Col: 26

Hasil:


Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:/INFORMATICS/prak ASD/Modul6/1_QuickSort.py ==========
QUICK SORT
2
4
5
10
13
18
23
29
31
51
64
>>>

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



No3.py - E:/INFORMATICS/prak ASD/Modul6/No3.py (2.7.15)

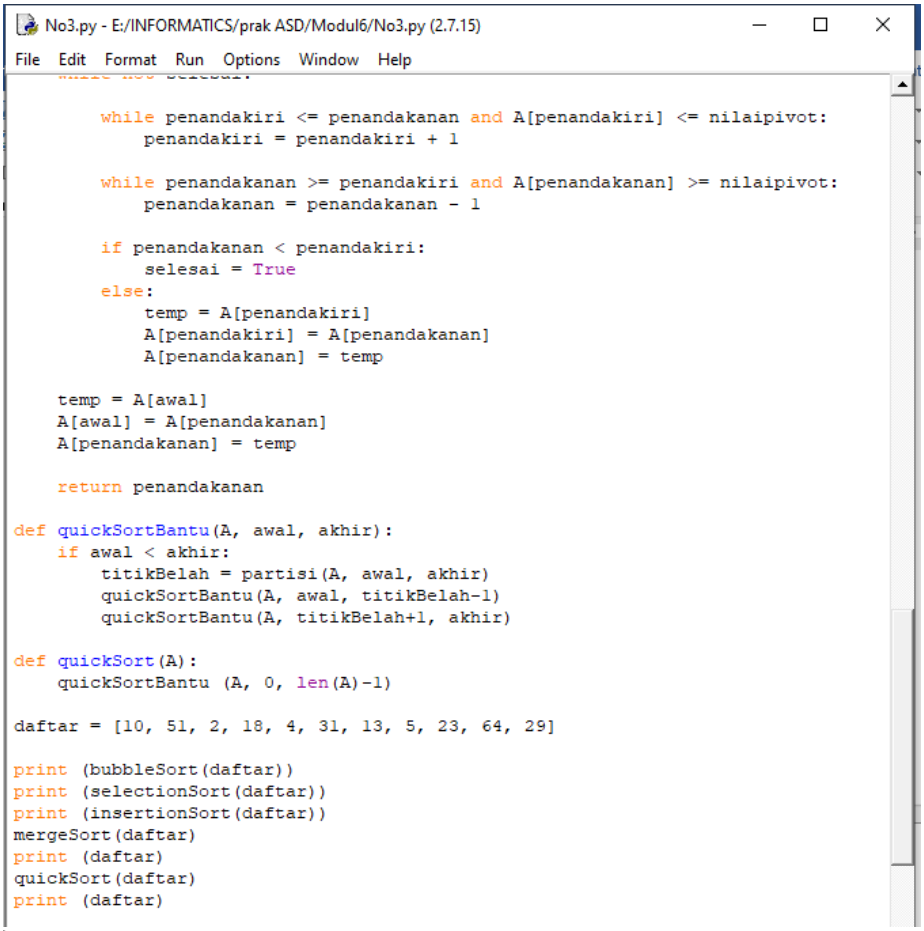
File Edit Format Run Options Window Help

```
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
```

No3.py - E:/INFORMATICS/prak ASD/Modul6/No3.py (2.7.15)

File Edit Format Run Options Window Help

```
    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
    while not selesai:
```




```
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));
```

Ln: 135 Col: 0

Hasil:

```
=============== RESTART: E:/INFORMATICS/prak ASD/Modul6/No3.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: 5.261 detik
selection: 2.129 detik
insertion: 2.279 detik
merge: 0.03 detik
quick: 0.027 detik
>>>
```

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

No5.py - E:/INFORMATICS/prak ASD/Modul6/No5.py (2.7.15)

File Edit Format Run Options Window Help

```
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]

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

def mergeSort2(A, awal, akhir):
    mid = (awal+akhir)//2
```

No5.py - E:/INFORMATICS/prak ASD/Modul6/No5.py (2.7.15)

File Edit Format Run Options Window Help

```
m9 = MhsTIF("Winda", 81, "Pontianak", 250000)
m10 = MhsTIF("Qina", 43, "Lombok", 550000)

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

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, f, l = 0, awal, mid+1
    tmp = [None] * (akhir - awal + 1)
    while f <= mid and l <= akhir:
        if A[f].ambilUangSaku() < A[l].ambilUangSaku():
            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 mergeSort(A):
    mergeSort2(A, 0, len(A)-1)
```

Hasil :

```
=============== RESTART: E:/INFORMATICS/prak ASD/Modul6/No5.py ===============
>>> 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 dari tiga untuk memilih pivot.

No6.py - E:/INFORMATICS/prak ASD/Modul6/No6.py (2.7.15)

File Edit Format Run Options Window Help

```
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():
```

No6.py - E:/INFORMATICS/prak ASD/Modul6/No6.py (2.7.15)

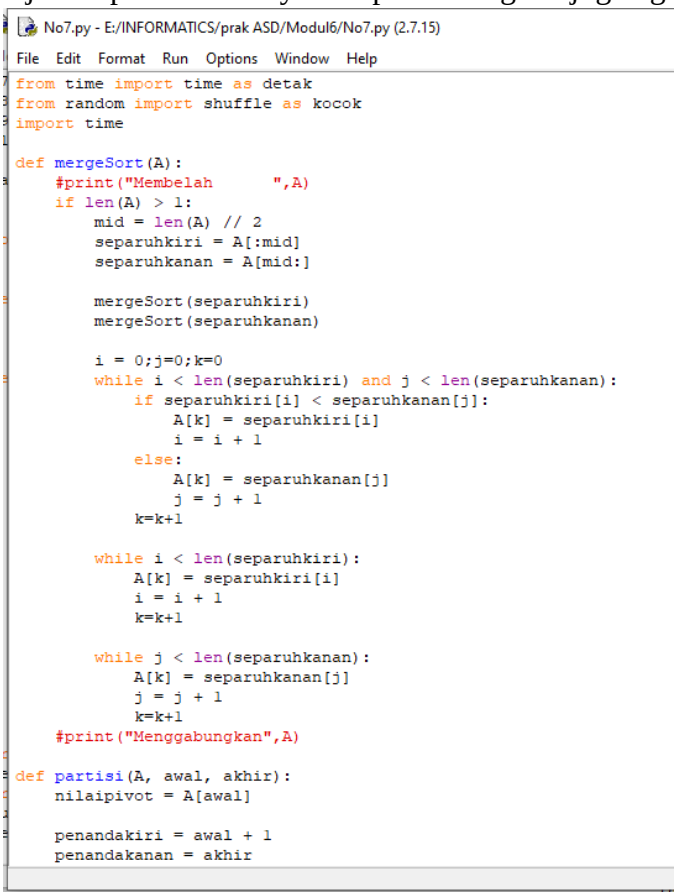
File Edit Format Run Options Window Help

```
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()
```

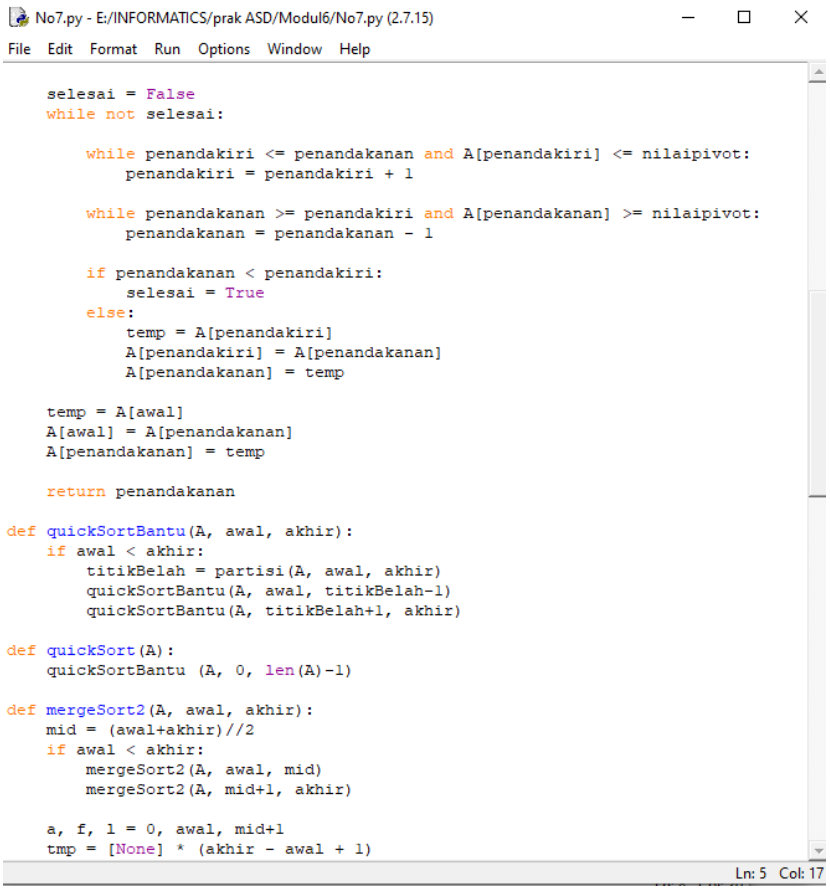
Hasil :

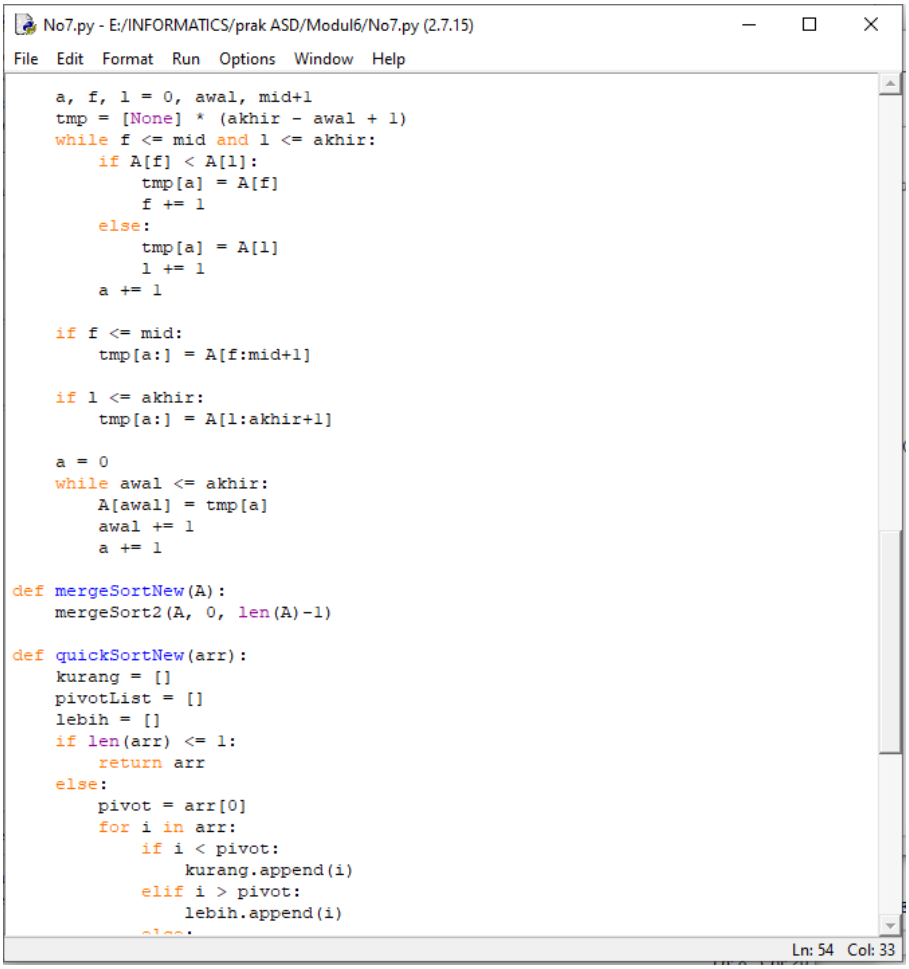
```
=============== RESTART: E:/INFORMATICS/prak ASD/Modul6/No6.py ===============
Sebelum diurutkan
Alfa
Pita
Octa
Ila
Uni
Yeri
Tisa
Roro
Elvi
Winda
Qina

Setelah diurutkan
Alfa
Pita
Octa
Ila
Uni
Yeri
Tisa
Roro
Elvi
Winda
Qina
>>>
```

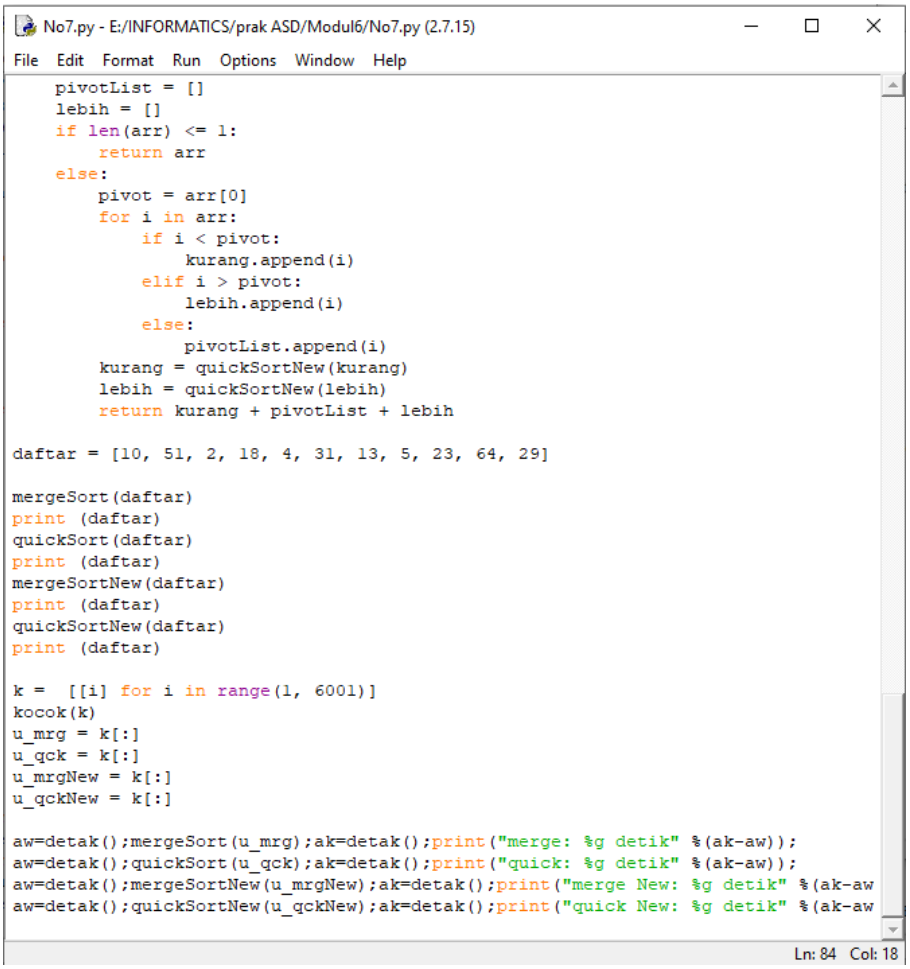
7. Uji kecepatan keduanya dan perbandingkan juga dgn kode awalnya.




```
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
```


No7.py - E:/INFORMATICS/prak ASD/Modul6/No7.py (2.7.15)
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 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)
Ln: 5 Col: 17


No7.py - E:/INFORMATICS/prak ASD/Modul6/No7.py (2.7.15)
File Edit Format Run Options Window Help
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)
Ln: 54 Col: 33



Hasil :

```
=============== RESTART: E:/INFORMATICS/prak ASD/Modul6/No7.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: 0.0340002 detik
quick: 0.0180001 detik
merge New: 0.049 detik
quick New: 0.026 detik
>>>
```

8. Buat versi linked list untuk program mergeSort di atas

No8.py - E:/INFORMATICS/prak ASD/Modul6/No8.py (2.7.15

File Edit Format Run Options Window Help

```
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
```

No8.py - E:/INFORMATICS/prak ASD/Modul6/No8.py (2.7.15)

File Edit Format Run Options Window Help

```
    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 :

```
=============== RESTART: E:/INFORMATICS/prak ASD/Modul6/No8.py ===============
1
2
3
4
5
6
7
>>>
```