PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

Modul 6

Pengurutan lanjutan

Disusun oleh:

DONI WAHYU SAPUTRO

L200200169

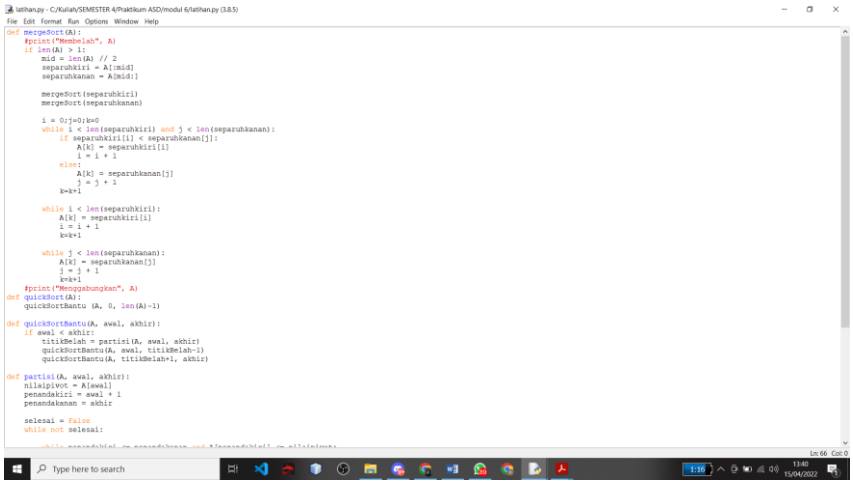
G

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

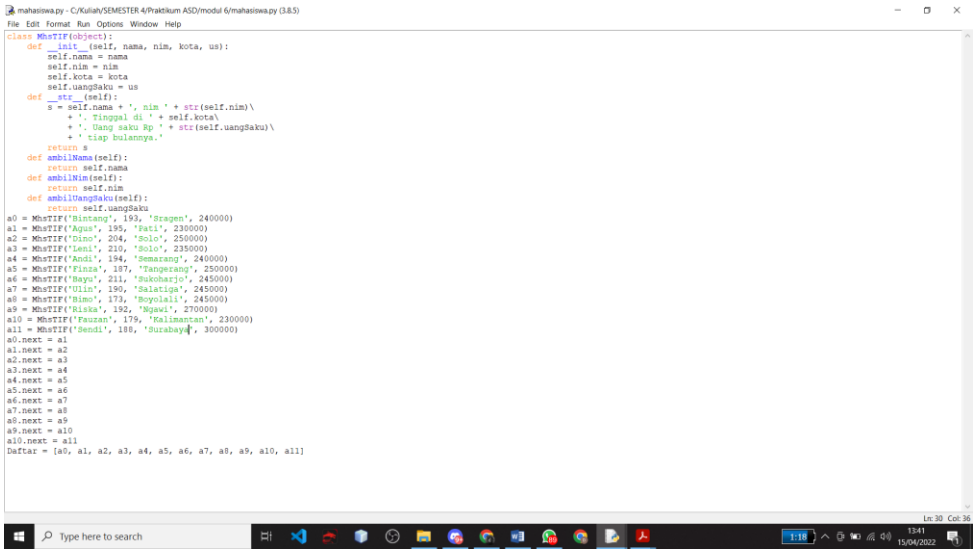
UNIVERSITAS MUHAMMADIYAH SURAKARTA

Latihan.py



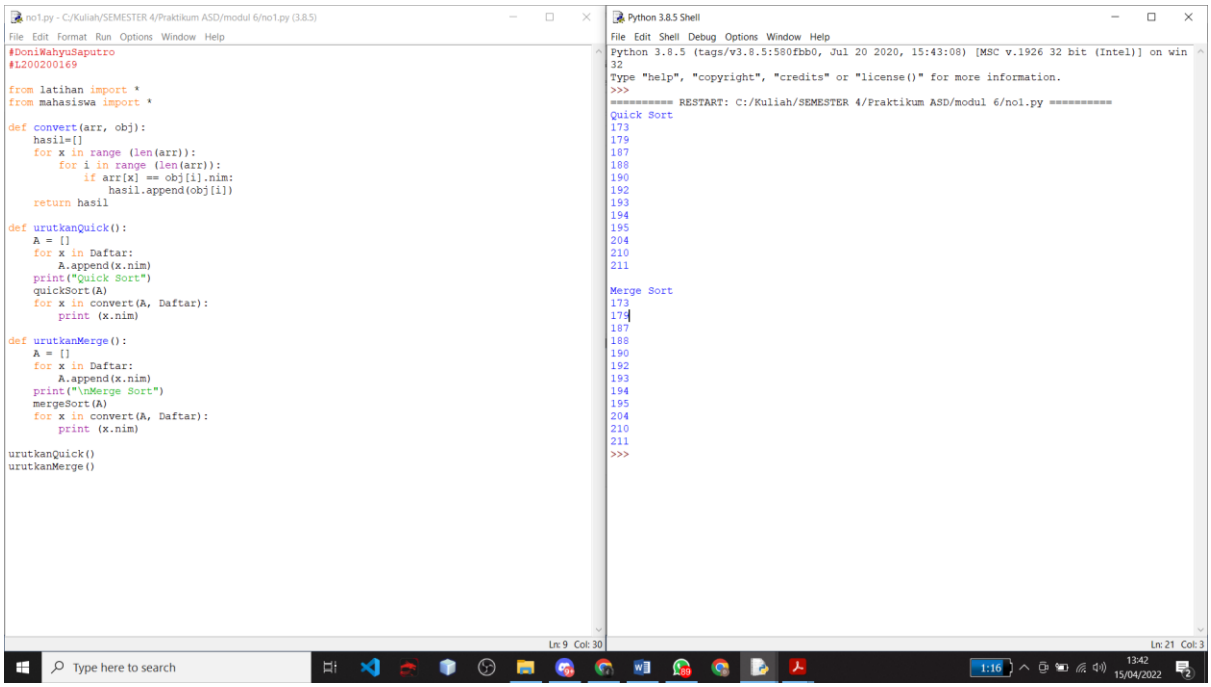
```
    selesai = False
    while not selesai:

        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
```

Mahasiswa.py



```
class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', nim ' + str(self.nim)\
            + '. Tinggal di ' + self.kota\
            + '. Uang saku Rp ' + str(self.uangSaku)\
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNim(self):
        return self.nim
    def ambilUangSaku(self):
        return self.uangSaku
a0 = MhsTIF('Bintang', 193, 'Sragen', 240000)
a1 = MhsTIF('Agus', 195, 'Pati', 230000)
a2 = MhsTIF('Dino', 204, 'Solo', 250000)
a3 = MhsTIF('Leni', 210, 'Solo', 235000)
a4 = MhsTIF('Andi', 194, 'Semarang', 240000)
a5 = MhsTIF('Finza', 187, 'Tangerang', 250000)
a6 = MhsTIF('Bayu', 211, 'Sukoharjo', 245000)
a7 = MhsTIF('Ulin', 190, 'Salatiga', 245000)
a8 = MhsTIF('Bimo', 173, 'Boyolali', 245000)
a9 = MhsTIF('Riska', 192, 'Ngawi', 270000)
a10 = MhsTIF('Fauzan', 179, 'Kalimantan', 230000)
a11 = MhsTIF('Sendi', 188, 'Surabaya', 300000)
a0.next = a1
a1.next = a2
a2.next = a3
a3.next = a4
a4.next = a5
a5.next = a6
a6.next = a7
a7.next = a8
a8.next = a9
a9.next = a10
a10.next = a11
Daftar = [a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11]
```

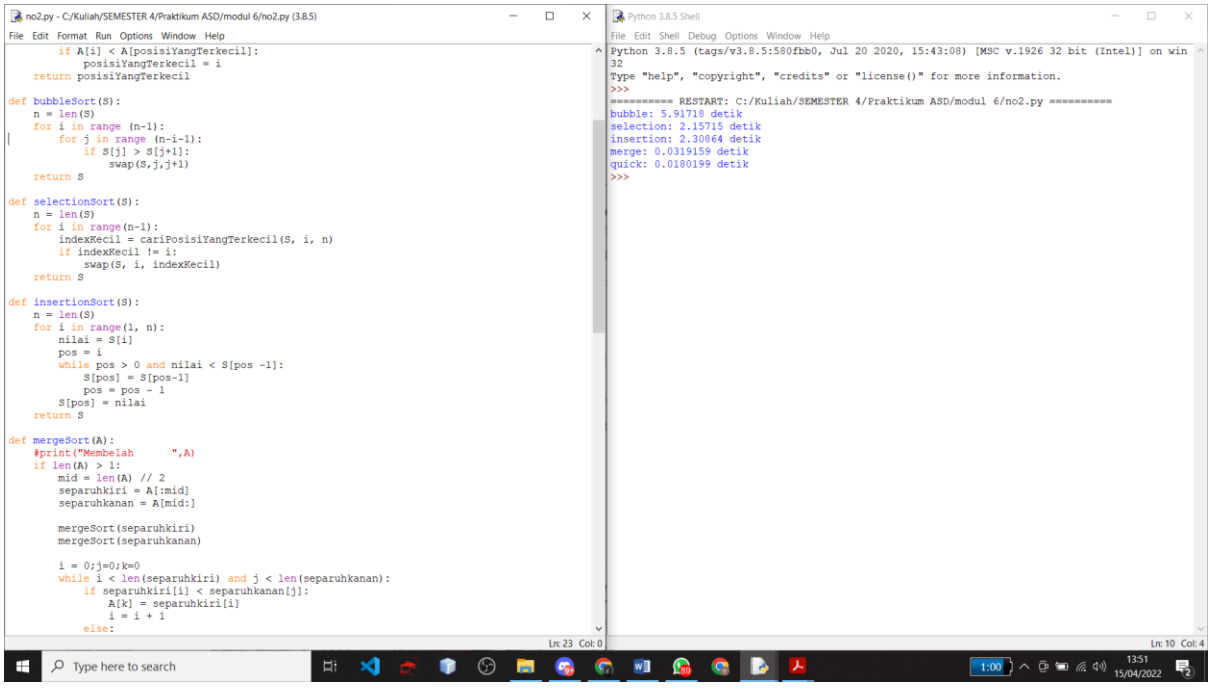
No1



```
#DoniWahyuSaputro
#L200200169

from latihan import *
from mahasiswa import *

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

def urutkanQuick():
    A = []
    for x in Daftar:
        A.append(x.nim)
    print("Quick Sort")
    quickSort(A)
    for x in convert(A, Daftar):
        print (x.nim)

def urutkanMerge():
    A = []
    for x in Daftar:
        A.append(x.nim)
    print("\nMerge Sort")
    mergeSort(A)
    for x in convert(A, Daftar):
        print (x.nim)

urutkanQuick()
urutkanMerge()
```

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no1.py ==========
Quick Sort
173
179
187
188
190
192
193
194
195
204
210
211

Merge Sort
173
179
187
188
190
192
193
194
195
204
210
211
>>>
```

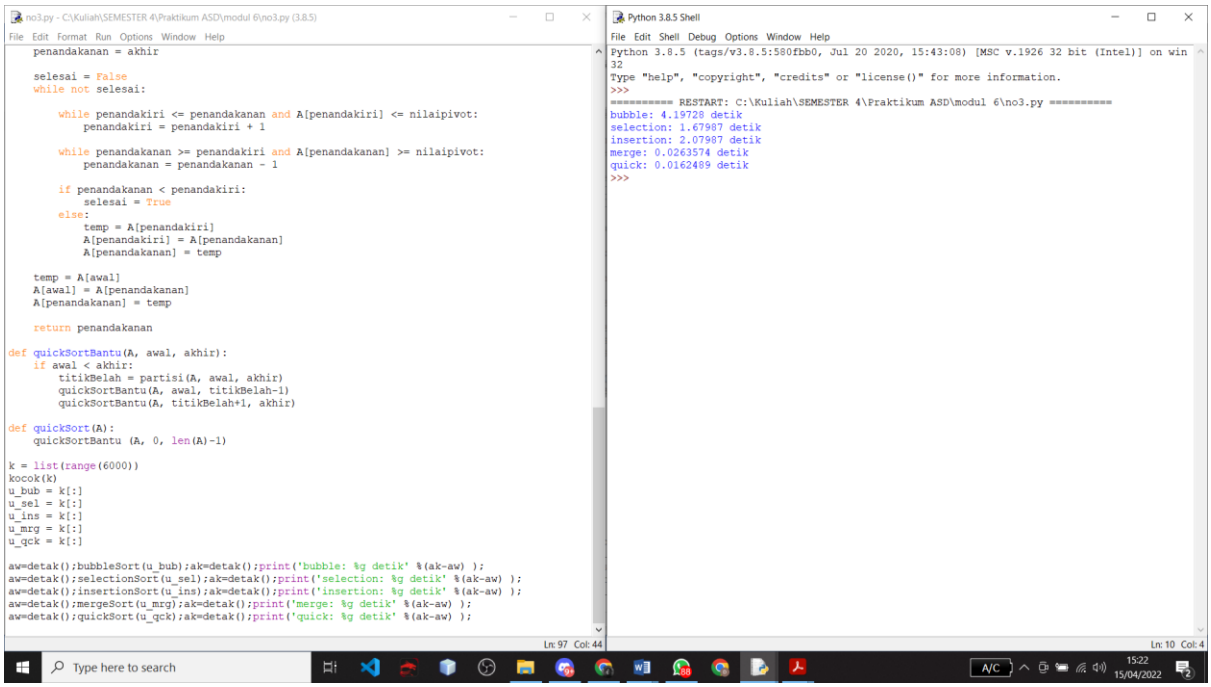
No2



```
        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
    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:
```

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no2.py ==========
bubble: 5.91718 detik
selection: 2.15715 detik
insertion: 2.30864 detik
merge: 0.0319159 detik
quick: 0.0180199 detik
>>>
```

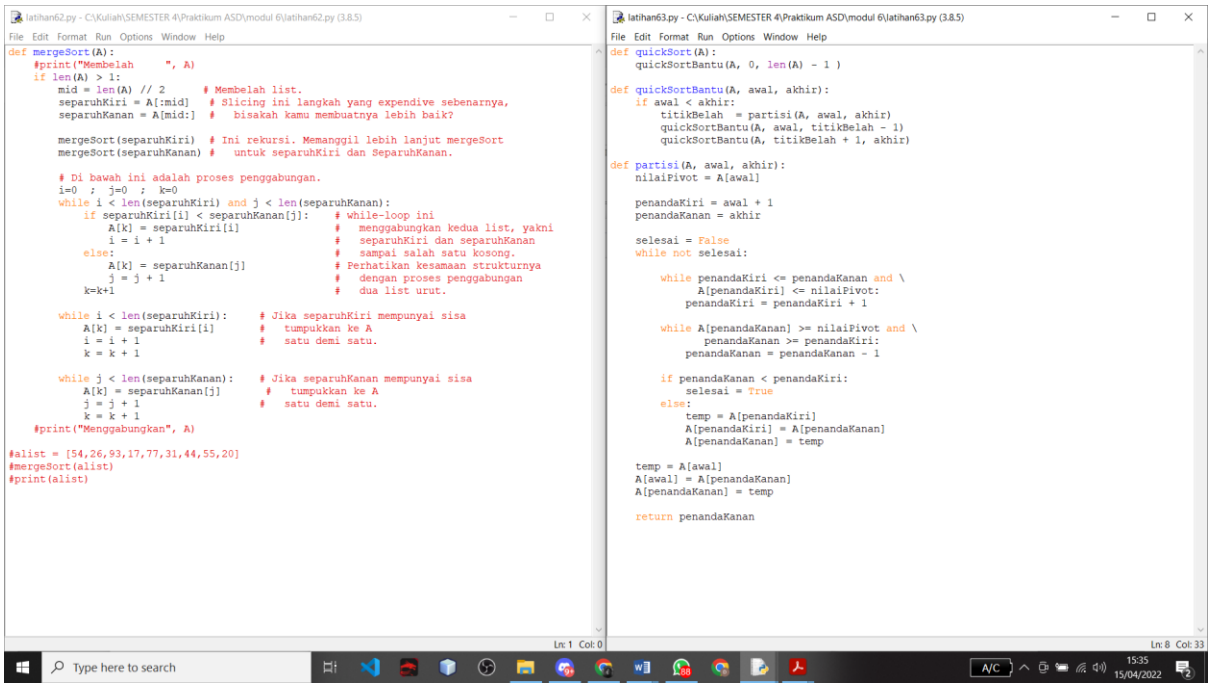
No3



```
    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)

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_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) );
```

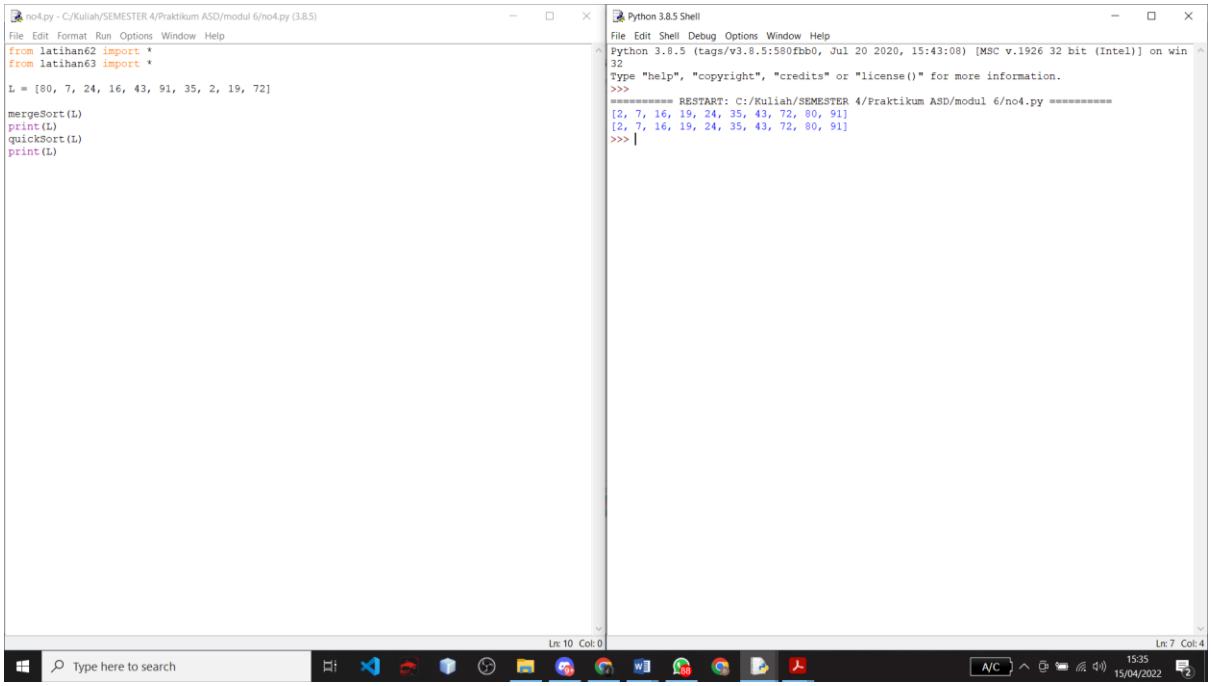
```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 6\no3.py ==========
bubble: 4.19728 detik
selection: 1.67987 detik
insertion: 2.07987 detik
merge: 0.0263574 detik
quick: 0.0162489 detik
>>>
```

No4

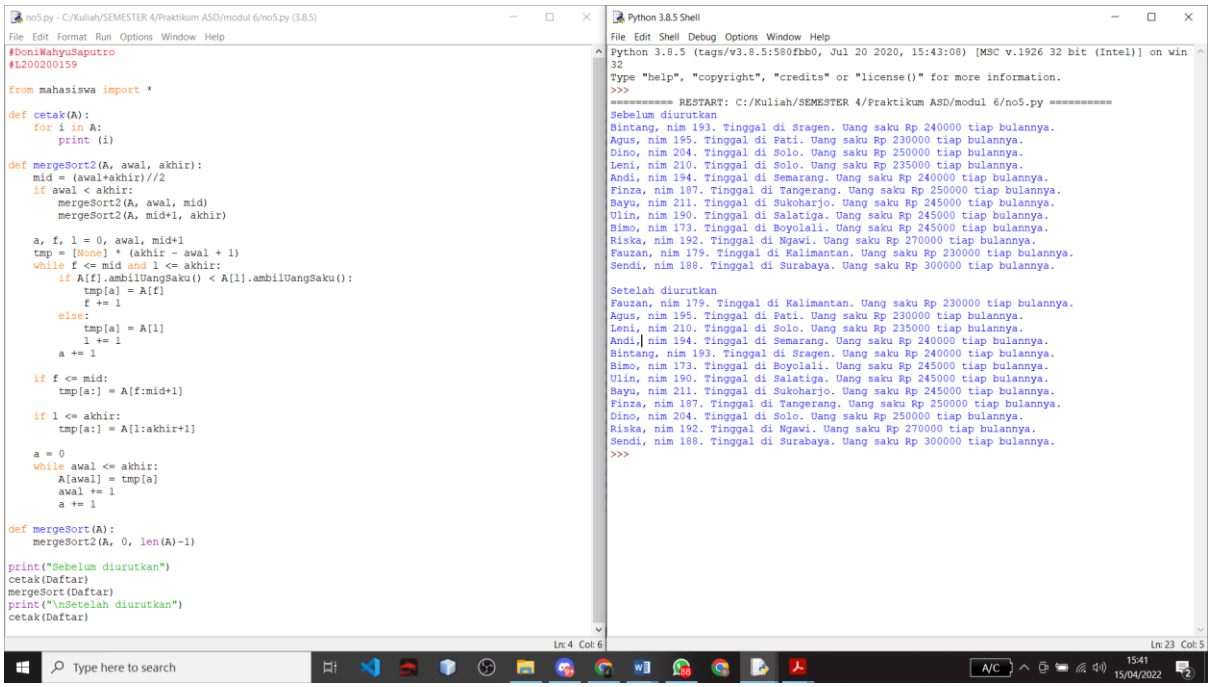


```
def mergeSort(A):
    #print("Membelah     ", A)
    if len(A) > 1:
        mid = len(A) // 2       # Membelah list.
        separuhKiri = A[:mid]   # Slicing ini langkah yang expendive sebenarnya,
        separuhKanan = A[mid:]  #   bisakah kamu membuatnya lebih baik?

        mergeSort(separuhKiri)  # Ini rekursi. Memanggil lebih lanjut mergeSort
        mergeSort(separuhKanan) #   untuk separuhKiri dan SeparuhKanan.

        # Di bawah ini adalah proses penggabungan.
        i=0  ;  j=0  ;  k=0
        while i < len(separuhKiri) and j < len(separuhKanan):
            if separuhKiri[i] < separuhKanan[j]:    # while-loop ini
                A[k] = separuhKiri[i]               #   menggabungkan kedua list, yakni
                i = i + 1                           #   separuhKiri dan separuhKanan
            else:                                   #   sampai salah satu kosong.
                A[k] = separuhKanan[j]              # Perhatikan kesamaan strukturnya
                j = j + 1                           #   dengan proses penggabungan
            k=k+1                                   #   dua list urut.

        while i < len(separuhKiri):     # Jika separuhKiri mempunyai sisa
            A[k] = separuhKiri[i]       #   tumpukkan ke A
            i = i + 1                   #   satu demi satu.
            k = k + 1

        while j < len(separuhKanan):    # Jika separuhKanan mempunyai sisa
            A[k] = separuhKanan[j]      #   tumpukkan ke A
            j = j + 1                   #   satu demi satu.
            k = k + 1
    #print("Menggabungkan", A)

#alist = [54,26,93,17,77,31,44,55,20]
#mergeSort(alist)
#print(alist)
```

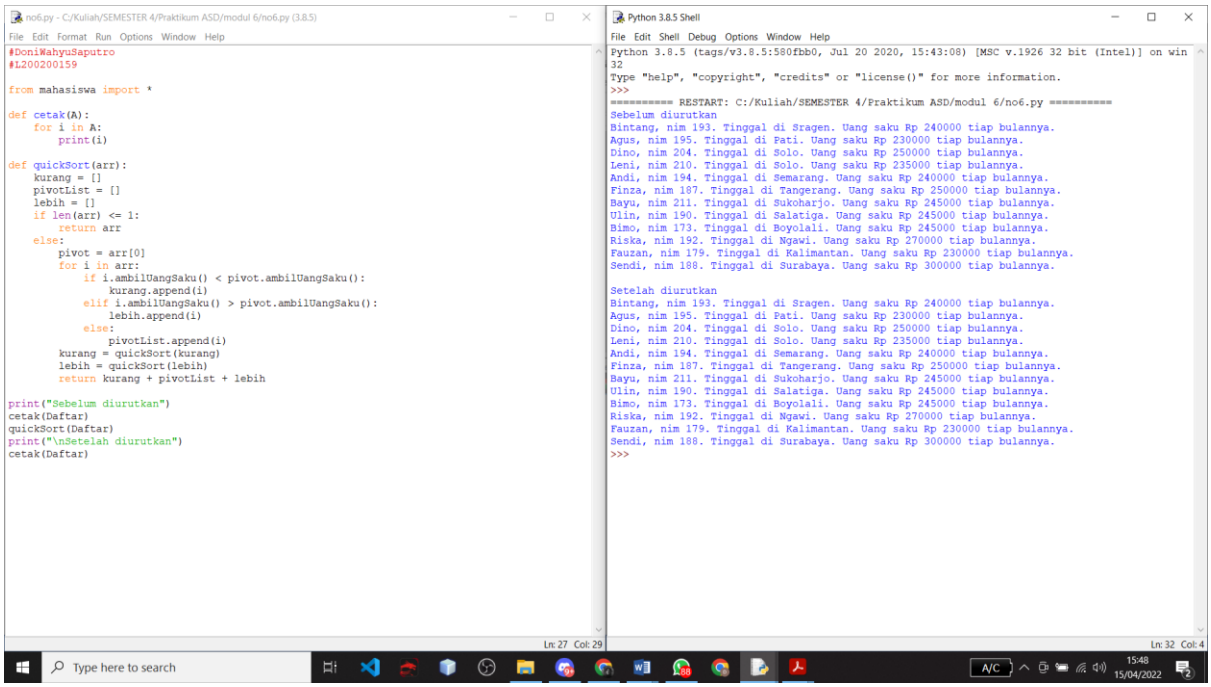
```
def quickSort(A):
    quickSortBantu(A, 0, len(A) - 1 )

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

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

    penandaKiri = awal + 1
    penandaKanan = akhir

    selesai = False
    while not selesai:

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

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

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

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

    return penandaKanan
```


no4.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no4.py (3.8.5)
File Edit Format Run Options Window Help
from latihan62 import *
from latihan63 import *
L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72]
mergeSort(L)
print(L)
quickSort(L)
print(L)
Ln: 10 Col: 0
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win 32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no4.py ==========
[2, 7, 16, 19, 24, 35, 43, 72, 80, 91]
[2, 7, 16, 19, 24, 35, 43, 72, 80, 91]
>>>
Ln: 7 Col: 4
Type here to search
A/C
15:35
15/04/2022

No5


no5.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no5.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200159
from mahasiswa import *
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)
print("Sebelum diurutkan")
cetak(Daftar)
mergeSort(Daftar)
print("\nSetelah diurutkan")
cetak(Daftar)
Ln: 4 Col: 6
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win 32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no5.py ==========
Sebelum diurutkan
Bintang, nim 193. Tinggal di Sragen. Uang saku Rp 240000 tiap bulannya.
Agus, nim 195. Tinggal di Pati. Uang saku Rp 230000 tiap bulannya.
Dino, nim 204. Tinggal di Solo. Uang saku Rp 250000 tiap bulannya.
Leni, nim 210. Tinggal di Solo. Uang saku Rp 235000 tiap bulannya.
Andi, nim 194. Tinggal di Semarang. Uang saku Rp 240000 tiap bulannya.
Finza, nim 187. Tinggal di Tangerang. Uang saku Rp 250000 tiap bulannya.
Bayu, nim 211. Tinggal di Sukoharjo. Uang saku Rp 245000 tiap bulannya.
Ulin, nim 190. Tinggal di Salatiga. Uang saku Rp 245000 tiap bulannya.
Bimo, nim 173. Tinggal di Boyolali. Uang saku Rp 245000 tiap bulannya.
Riska, nim 192. Tinggal di Ngawi. Uang saku Rp 270000 tiap bulannya.
Fauzan, nim 179. Tinggal di Kalimantan. Uang saku Rp 230000 tiap bulannya.
Sendi, nim 188. Tinggal di Surabaya. Uang saku Rp 300000 tiap bulannya.
Setelah diurutkan
Fauzan, nim 179. Tinggal di Kalimantan. Uang saku Rp 230000 tiap bulannya.
Agus, nim 195. Tinggal di Pati. Uang saku Rp 230000 tiap bulannya.
Leni, nim 210. Tinggal di Solo. Uang saku Rp 235000 tiap bulannya.
Andi, nim 194. Tinggal di Semarang. Uang saku Rp 240000 tiap bulannya.
Bintang, nim 193. Tinggal di Sragen. Uang saku Rp 240000 tiap bulannya.
Bimo, nim 173. Tinggal di Boyolali. Uang saku Rp 245000 tiap bulannya.
Ulin, nim 190. Tinggal di Salatiga. Uang saku Rp 245000 tiap bulannya.
Bayu, nim 211. Tinggal di Sukoharjo. Uang saku Rp 245000 tiap bulannya.
Finza, nim 187. Tinggal di Tangerang. Uang saku Rp 250000 tiap bulannya.
Dino, nim 204. Tinggal di Solo. Uang saku Rp 250000 tiap bulannya.
Riska, nim 192. Tinggal di Ngawi. Uang saku Rp 270000 tiap bulannya.
Sendi, nim 188. Tinggal di Surabaya. Uang saku Rp 300000 tiap bulannya.
>>>
Ln: 23 Col: 5
Type here to search
A/C
15:41
15/04/2022

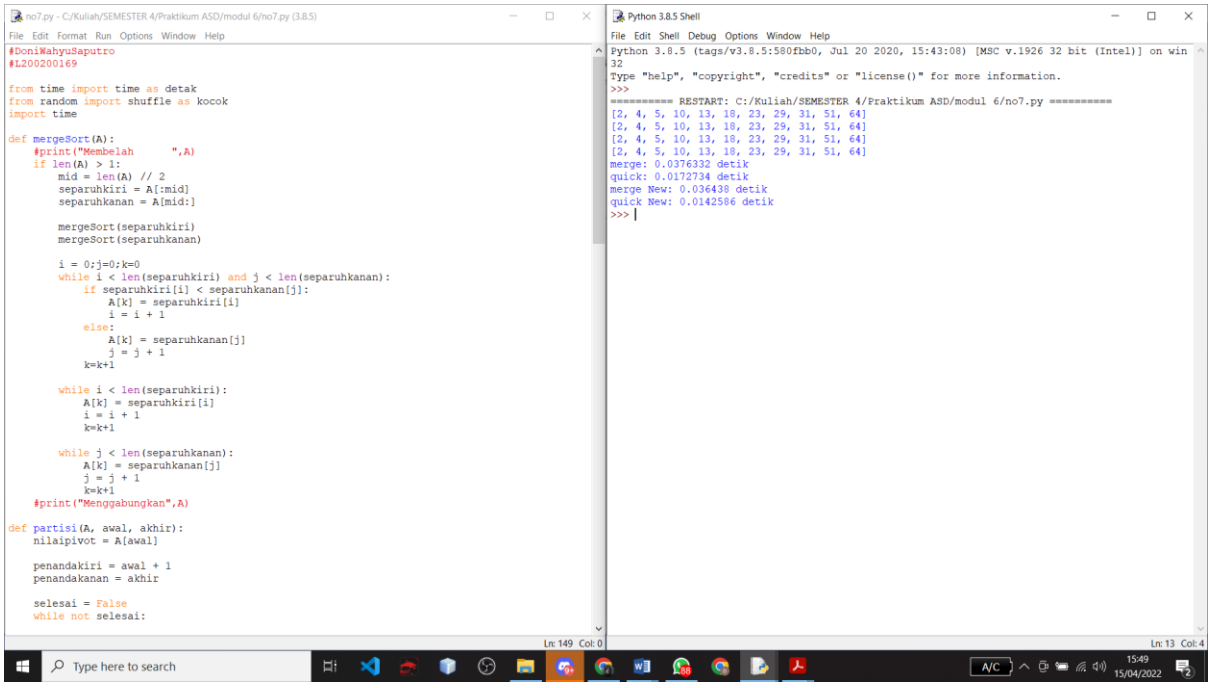
No6



```
#DoniWahyuSaputro
#L200200159

from mahasiswa import *

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

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

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

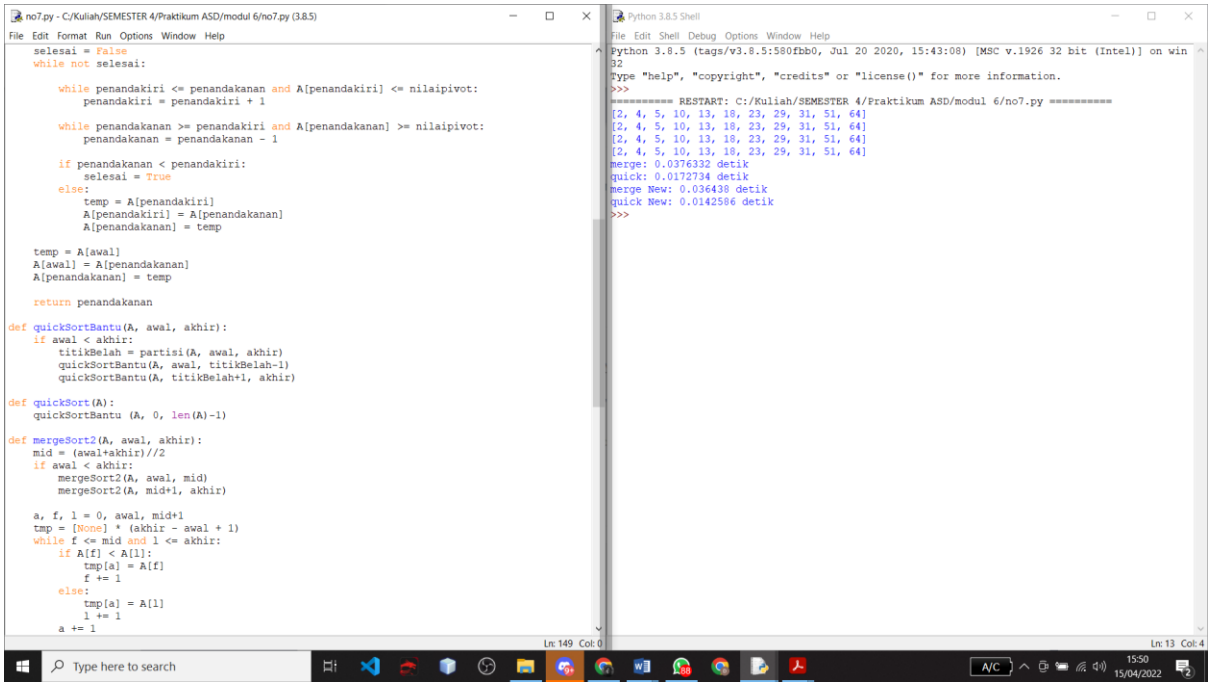
```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no6.py ==========
Sebelum diurutkan
Bintang, nim 193. Tinggal di Sragen. Uang saku Rp 240000 tiap bulannya.
Agus, nim 195. Tinggal di Pati. Uang saku Rp 230000 tiap bulannya.
Dino, nim 204. Tinggal di Solo. Uang saku Rp 250000 tiap bulannya.
Leni, nim 210. Tinggal di Solo. Uang saku Rp 235000 tiap bulannya.
Andi, nim 194. Tinggal di Semarang. Uang saku Rp 240000 tiap bulannya.
Finza, nim 187. Tinggal di Tangerang. Uang saku Rp 250000 tiap bulannya.
Bayu, nim 211. Tinggal di Sukoharjo. Uang saku Rp 245000 tiap bulannya.
Ulin, nim 190. Tinggal di Salatiga. Uang saku Rp 245000 tiap bulannya.
Bimo, nim 173. Tinggal di Boyolali. Uang saku Rp 245000 tiap bulannya.
Riska, nim 192. Tinggal di Ngawi. Uang saku Rp 270000 tiap bulannya.
Fauzan, nim 179. Tinggal di Kalimantan. Uang saku Rp 230000 tiap bulannya.
Sendi, nim 188. Tinggal di Surabaya. Uang saku Rp 300000 tiap bulannya.

Setelah diurutkan
Bintang, nim 193. Tinggal di Sragen. Uang saku Rp 240000 tiap bulannya.
Agus, nim 195. Tinggal di Pati. Uang saku Rp 230000 tiap bulannya.
Dino, nim 204. Tinggal di Solo. Uang saku Rp 250000 tiap bulannya.
Leni, nim 210. Tinggal di Solo. Uang saku Rp 235000 tiap bulannya.
Andi, nim 194. Tinggal di Semarang. Uang saku Rp 240000 tiap bulannya.
Finza, nim 187. Tinggal di Tangerang. Uang saku Rp 250000 tiap bulannya.
Bayu, nim 211. Tinggal di Sukoharjo. Uang saku Rp 245000 tiap bulannya.
Ulin, nim 190. Tinggal di Salatiga. Uang saku Rp 245000 tiap bulannya.
Bimo, nim 173. Tinggal di Boyolali. Uang saku Rp 245000 tiap bulannya.
Riska, nim 192. Tinggal di Ngawi. Uang saku Rp 270000 tiap bulannya.
Fauzan, nim 179. Tinggal di Kalimantan. Uang saku Rp 230000 tiap bulannya.
Sendi, nim 188. Tinggal di Surabaya. Uang saku Rp 300000 tiap bulannya.
>>>
```

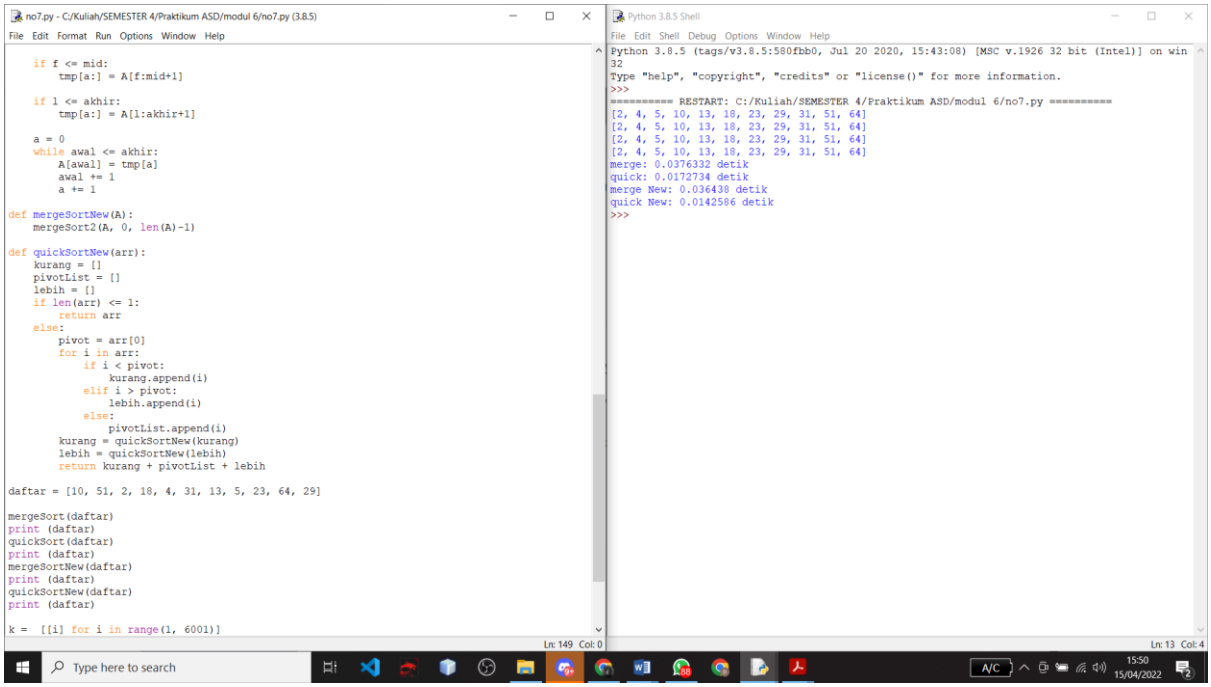
No7



```
#DoniWahyuSaputro
#L200200169

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:
```

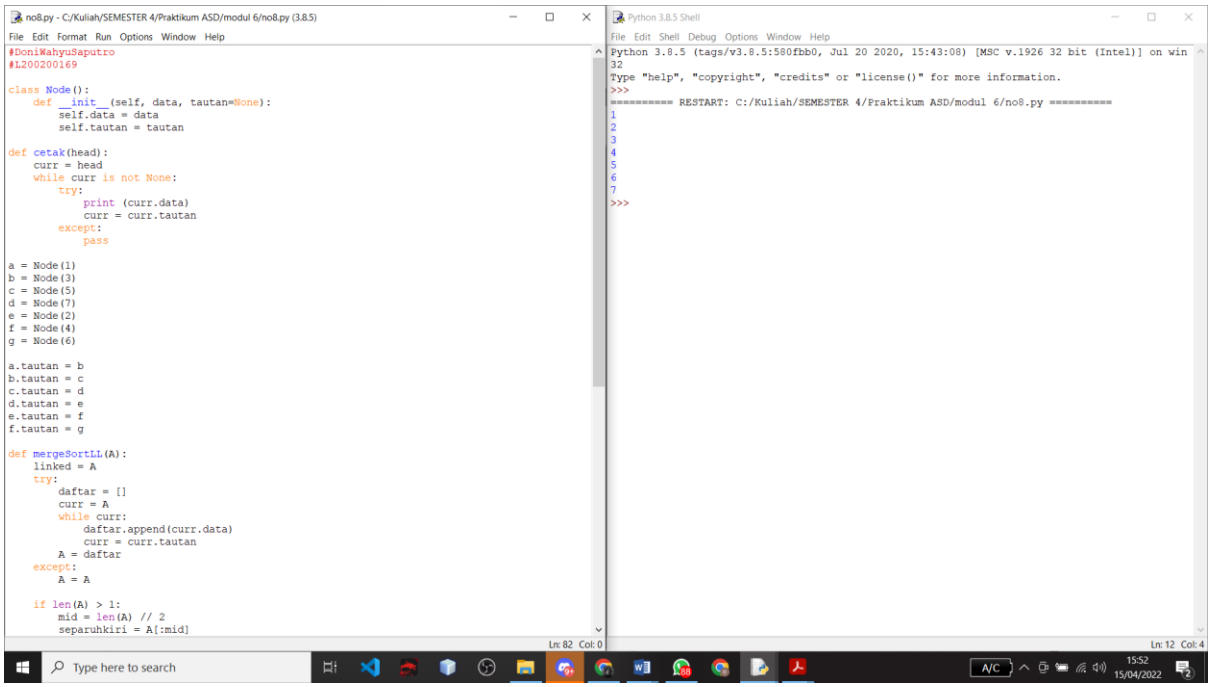
```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no7.py ==========
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 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: 0.0376332 detik
quick: 0.0172734 detik
merge New: 0.036438 detik
quick New: 0.0142586 detik
>>>
```

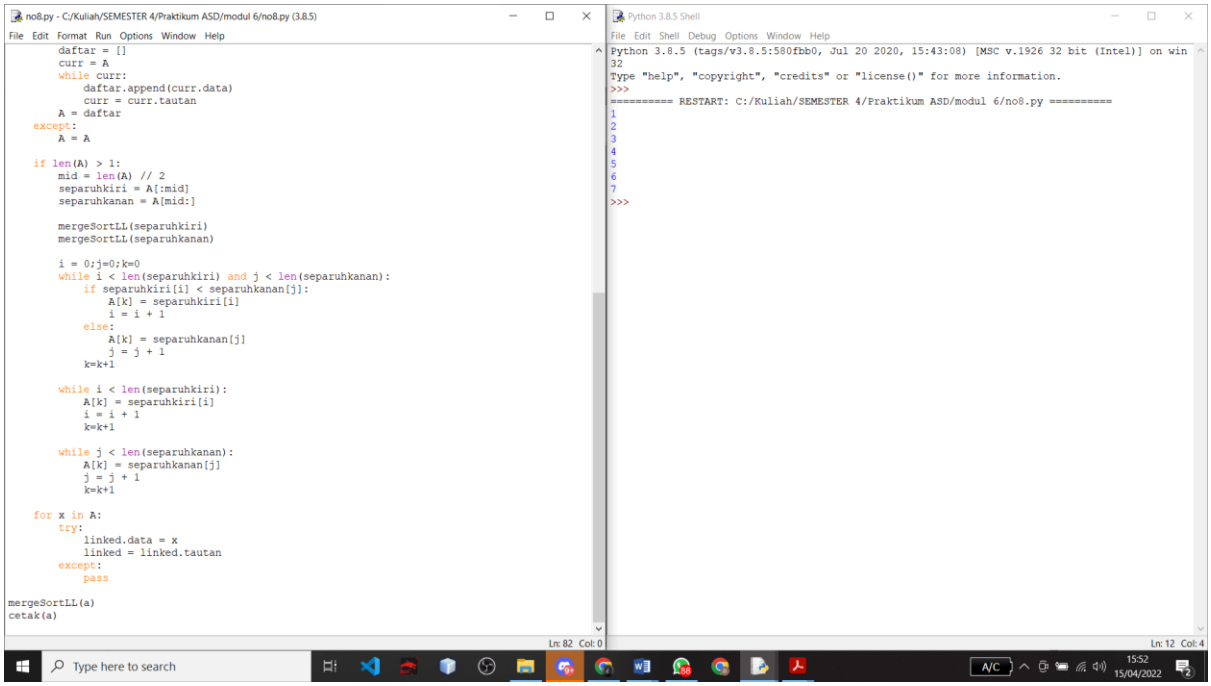





```
k =  [[i] for i in range(1, 6001)]
kocok(k)
u_mrg = k[:]
u_qck = k[:]
u_mrgNew = k[:]
u_qckNew = k[:]

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-aw));
aw=detak();quickSortNew(u_qckNew);ak=detak();print("quick New: %g detik" %(ak-aw));
```

No8


no8.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no8.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169
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]
Ln: 82 Col: 0
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no8.py ==========
1
2
3
4
5
6
7
>>>
Ln: 12 Col: 4
Type here to search
A/C
15:52
15/04/2022


no8.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no8.py (3.8.5)
File Edit Format Run Options Window Help
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)
Ln: 82 Col: 0
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no8.py ==========
1
2
3
4
5
6
7
>>>
Ln: 12 Col: 4
Type here to search
A/C
15:52
15/04/2022