Nama : **Muhammad Khalif Rizaldi Wibowo**

Mata Kuliah : **Praktikum Algoritma dan Struktur Data**

Kelas : **H**

NIM : **L200180217**

Modul 1

1. Fungsi cetakSiku(x)

```
#1
def cetakSiku(x):
    for i in range (1,x+1):
        print('*'*i)
```

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 1
tel)] on win32
Type "help", "copyright", "credits" or "license()
>>>
== RESTART: C:/Users/ASUS/AppData/Local/Programs/
>>> cetakSiku(5)
*
**
***
****
*****
```

2. Fungsi 2 integer positif

```
#2
def gambarlahPersegiEmpat(a, b):
    for i in range (a):
        if i==0 or i==a-1:
            print (b * '@')
        else:
            print ('@' + " " * (b-2) + '@')
```

```
>>>
== RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul1.py ==
>>> gambarlahPersegiEmpat(4, 5)
@@@@@
@   @
@   @
@@@@@
>>>
```

3. Fungsi yan menerima string, mengembalikan list yang terdiri dari 2 integer dan menghitung jumlah vokal

A.

```
def jumlahHurufVokal(huruf):
    vokal = 'aiueoAIUEO'
    a = 0
    hasil = 0
    for i in huruf :
        if i in vokal:
            a += len(i)
        else:
            a += 0
    hasil = len(huruf), a
    return hasil
```

```
>>>
== RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul1.py ==
>>> jumlahHurufVokal('semarang')
(8, 3)
>>>
```

B.

```
#3b
def jumlahHurufKonsonan(huruf):
    konsonan = 'bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ'
    b = 0
    hasil = 0
    for i in huruf:
        if i in konsonan:
            b +=len(i)
        else:
            b +=0
    hasil = len(huruf),b
    return hasil
```

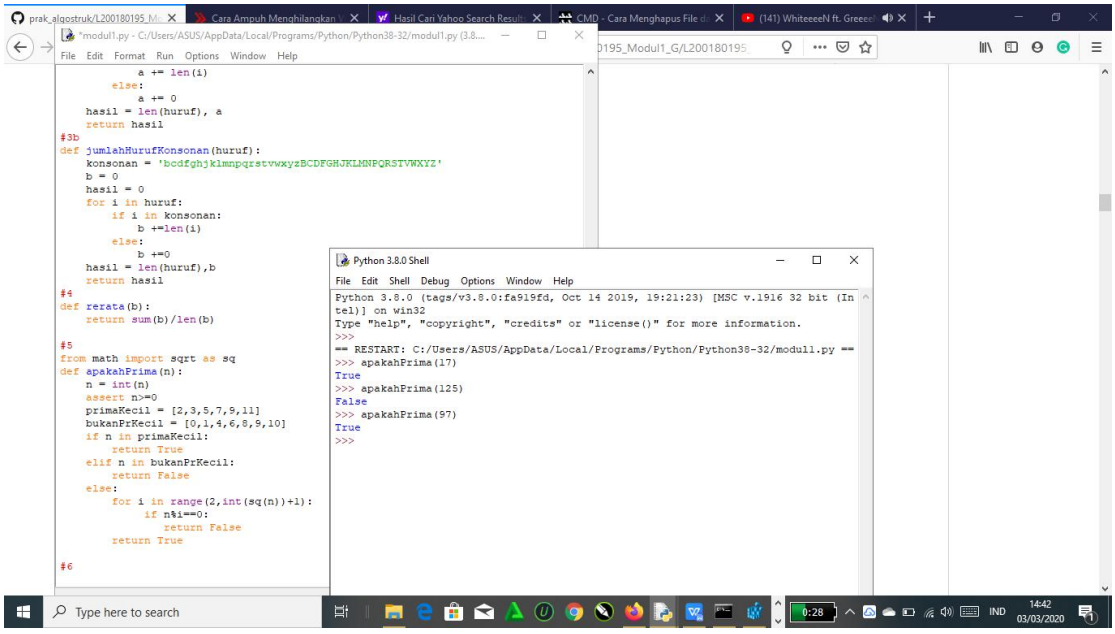
```
>>> jumlahHurufKonsonan('semarang')
(8, 5)
>>>
```

4. Fungsi rerata sebuah array yang berisi bilangan

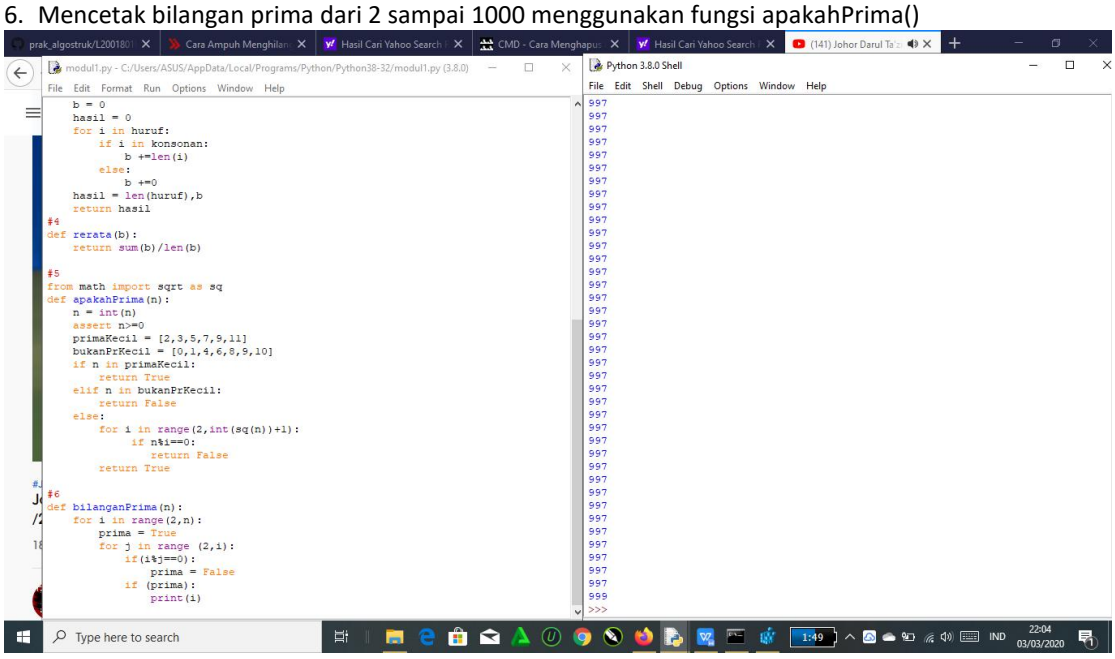
```
    return hasil
#4
def rerata(b):
    return sum(b)/len(b)
```

```
>>> rerata([1,2,3,4,5,6])
3.5
>>> rerata([3,4,6,7,9,90])
19.833333333333332
>>>
```

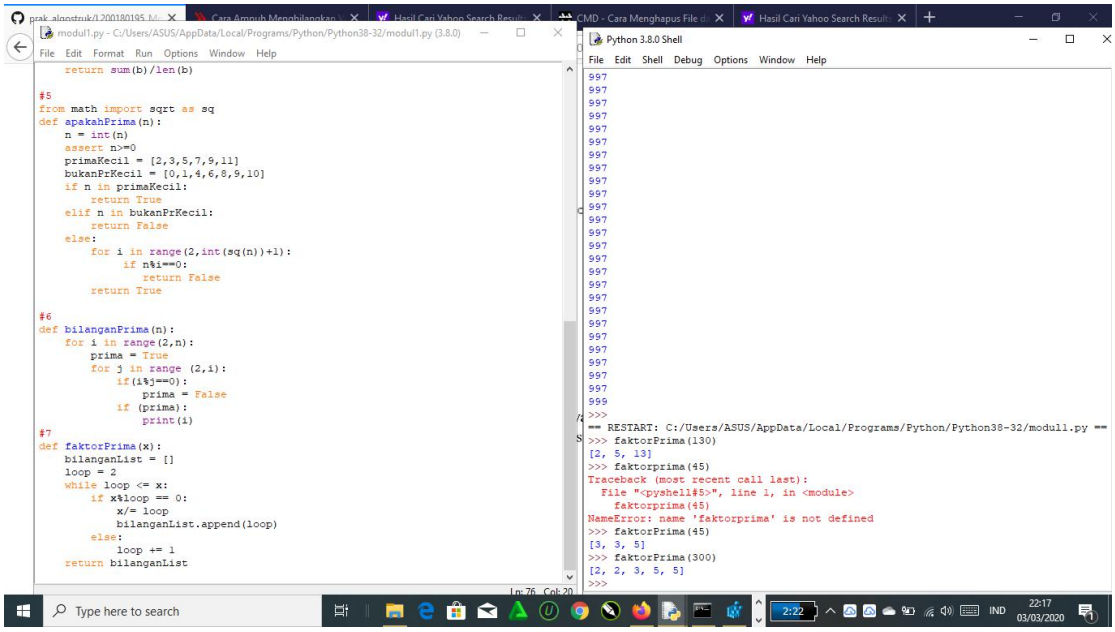
5. Fungsi bilangan prima atau bukan



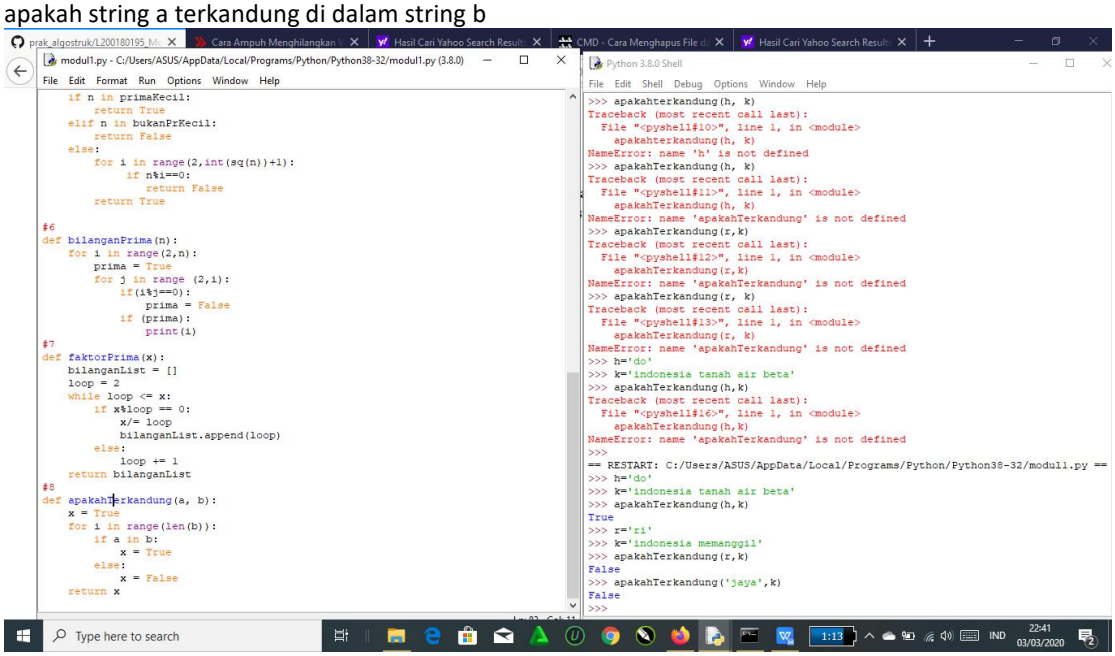
6. Mencetak bilangan prima dari 2 sampai 1000 menggunakan fungsi apakahPrima()



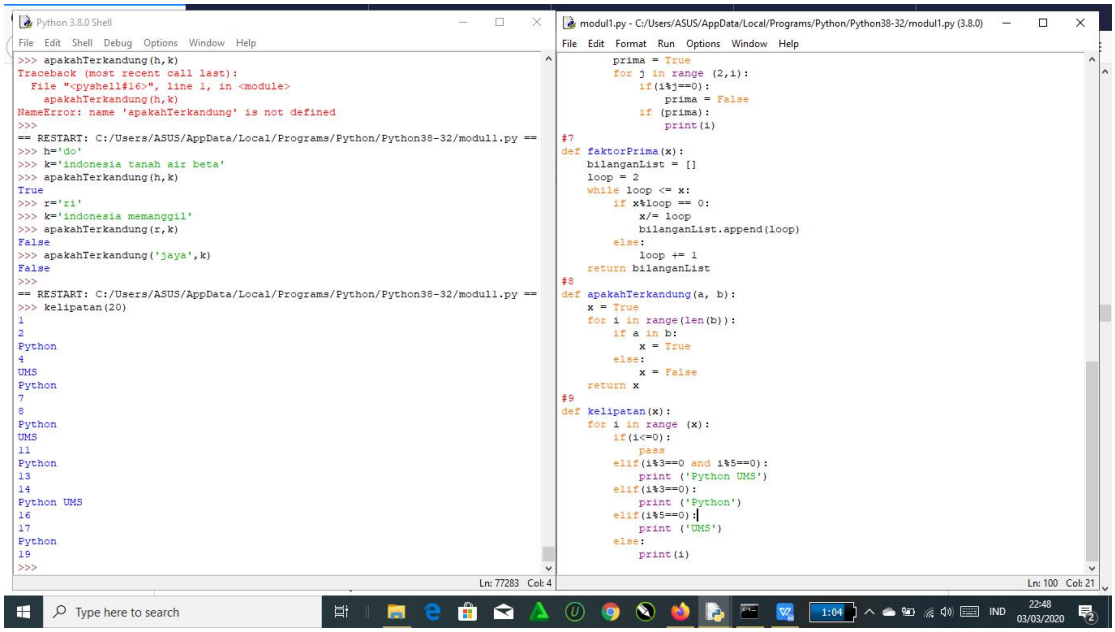
7. Membuat program yang menerima bilangan positif dan memberikan faktorisasi prima nya



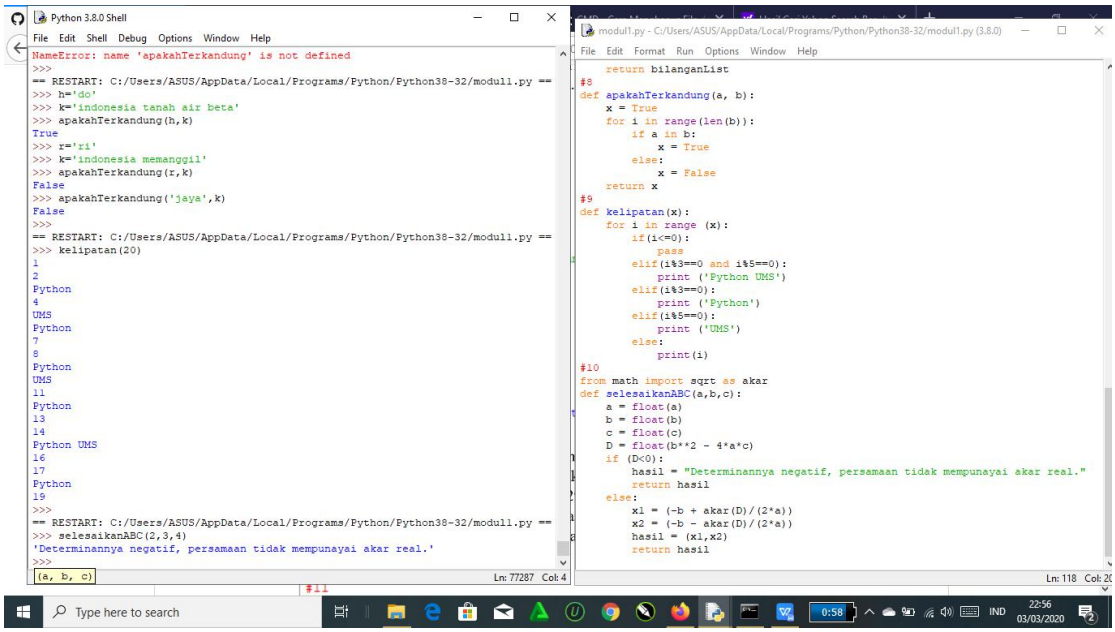
8. Buat suatu fungsi apakah terkandung (a, b) yang menerima 2 string a dan b, lalu menentukan apakah string a terkandung di dalam string b



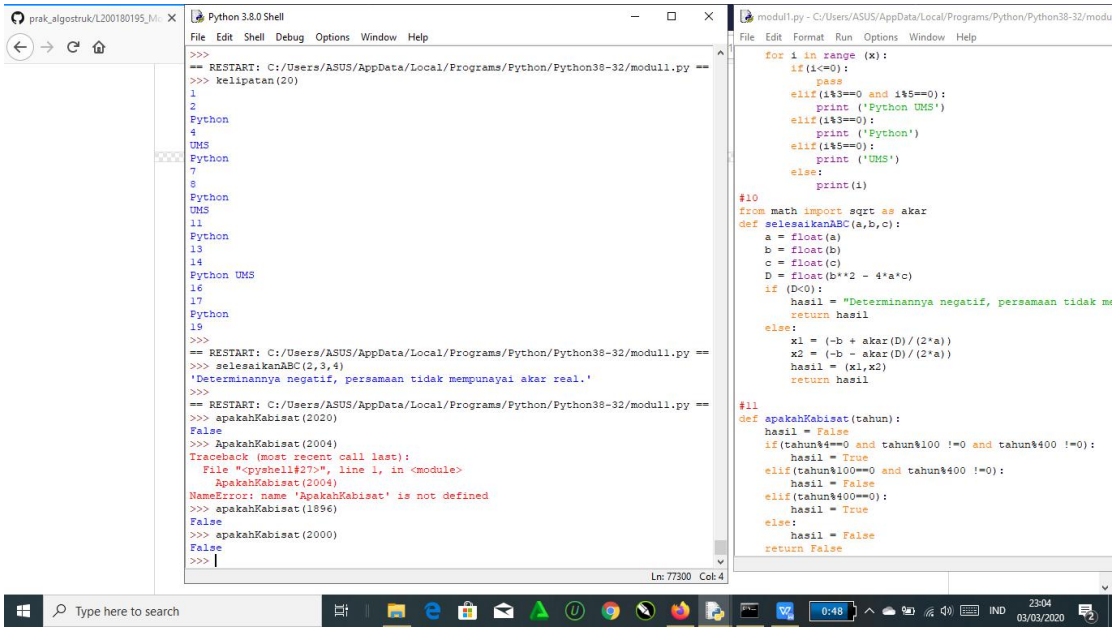
9. Buatlah program untuk mencetak angka dari 1 sampai 100. kalau angka nya pas kelipatan 3 maka cetak ‘python’ kalau pas kelipatan 5 cetak ‘ums’ kalau pas kelipatan 3 dan 5 maka cetak ‘python ums’



10. Buat modifikasi pada contoh 1.4 agar bisa menangkap kasus di mana determinannya kurang dari nol. Jika ini terjadi maka tampilkan di layar seperti ini : Determinannya negatif tidak mempunyai akar real.



11. Fungsi apakahKabisat() yang menerima suatu angka(tahun). jika tahun itu kabisat kembalikan True , jika tidak kembalikan False. Tahun kabisat tahun yang memiliki tanggal 29 februari


prak_algostruk/L20018019...
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
>>>
== RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul1.py ==
>>> kelipatan(20)
1
2
Python
4
UMS
Python
7
8
Python
UMS
11
Python
13
14
Python UMS
16
17
Python
19
>>>
== RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul1.py ==
>>> selesaikanABC(2,3,4)
'Determinannya negatif, persamaan tidak mempunayai akar real.'
>>>
== RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul1.py ==
>>> apakahKabisat(2020)
False
>>> ApakahKabisat(2004)
Traceback (most recent call last):
File "<pyshell#27>", line 1, in <module>
ApakahKabisat(2004)
NameError: name 'ApakahKabisat' is not defined
>>> apakahKabisat(1896)
False
>>> apakahKabisat(2000)
False
>>>
Ln: 77300 Col: 4
modul1.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modu
File Edit Format Run Options Window Help
for i in range (x):
if(i<=0):
pass
elif(i%3==0 and i%5==0):
print ('Python UMS')
elif(i%3==0):
print ('Python')
elif(i%5==0):
print ('UMS')
else:
print(i)
#10
from math import sqrt as akar
def selesaikanABC(a,b,c):
a = float(a)
b = float(b)
c = float(c)
D = float(b**2 - 4*a*c)
if (D<0):
hasil = "Determinannya negatif, persamaan tidak m
return hasil
else:
x1 = (-b + akar(D)/(2*a))
x2 = (-b - akar(D)/(2*a))
hasil = (x1,x2)
return hasil
#11
def apakahKabisat(tahun):
hasil = False
if(tahun%4==0 and tahun%100 !=0 and tahun%400 !=0):
hasil = True
elif(tahun%100==0 and tahun%400 !=0):
hasil = False
elif(tahun%400==0):
hasil = True
else:
hasil = False
return False
Type here to search
0:48
IND
23:04
03/03/2020

12.

Muh Khalif Rizaldi W
L200180217 / Kelas H

Modul 2

Tugas

Kode .py

Latoop2.py - C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pertemuan2\Latoop2.py (3.7.0)

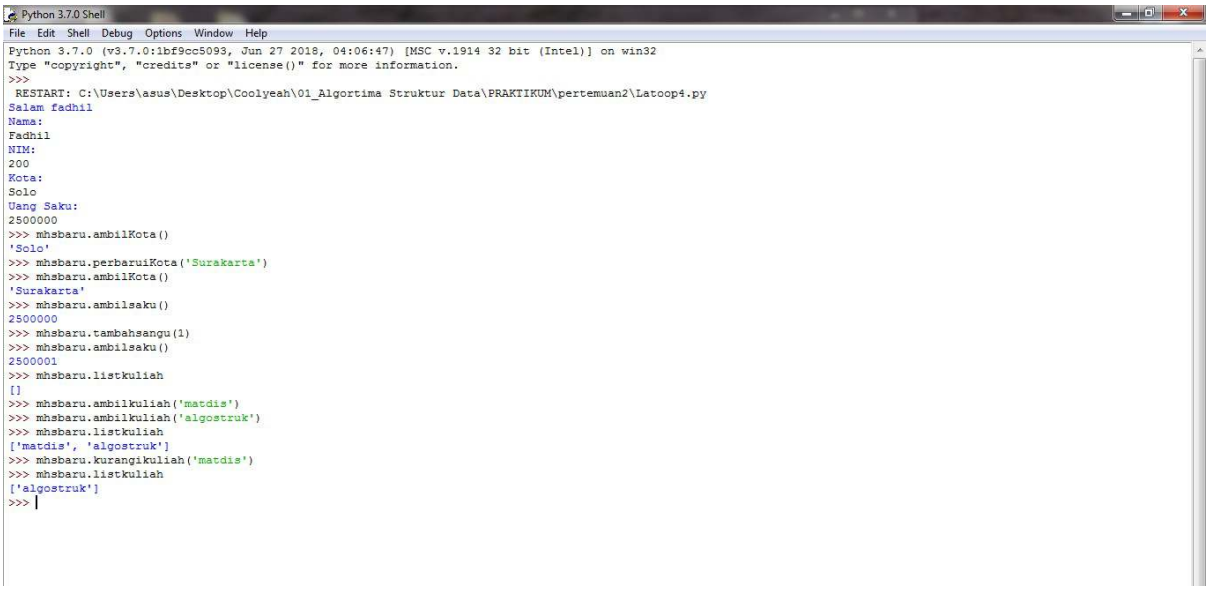
File Edit Format Run Options Window Help

```
class Pesan(object):
        """docstring for Pesan"""
        def __init__(self, string):
                self.string = string
        def cetakIni(self):
                print (self.string)
        def kapital(self):
                print (str.upper(self.string))
        def lower(self):
                print (str.lower(self.string))
        def jumKar(self):
                return len(self.string)
        def cetakjumkar(self):
                print ('kalimat', len(self.string),' karakter')
        def perbarui(self, stringbaru):
                self.string = stringbaru
        def apakahTerkandung(self, x):
                if x in self.string:
                        print ('true')
                else:
                        print ('false')
        def vokal(self):
                hvokal=['a','i','u','e','o','A','I','U','E','O']
                count=0
                for i in self.string:
                        if i in hvokal:
                                count+=1
                return (count)

        def konsonan(self):
                hvokal=['a','i','u','e','o','A','I','U','E','O']
                count=0
                for i in self.string:
                        if i not in hvokal:
                                count+=1
                return (count)
```

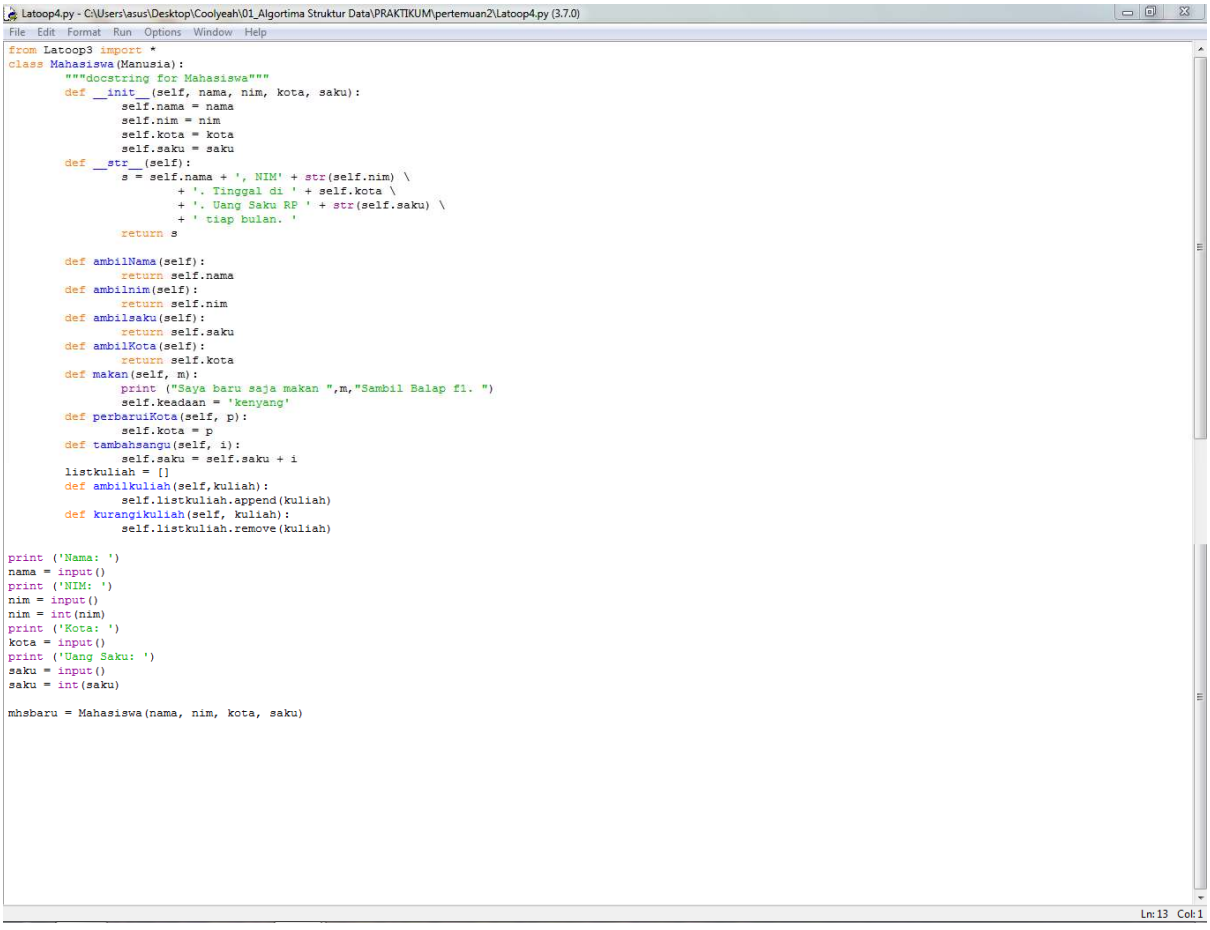
Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

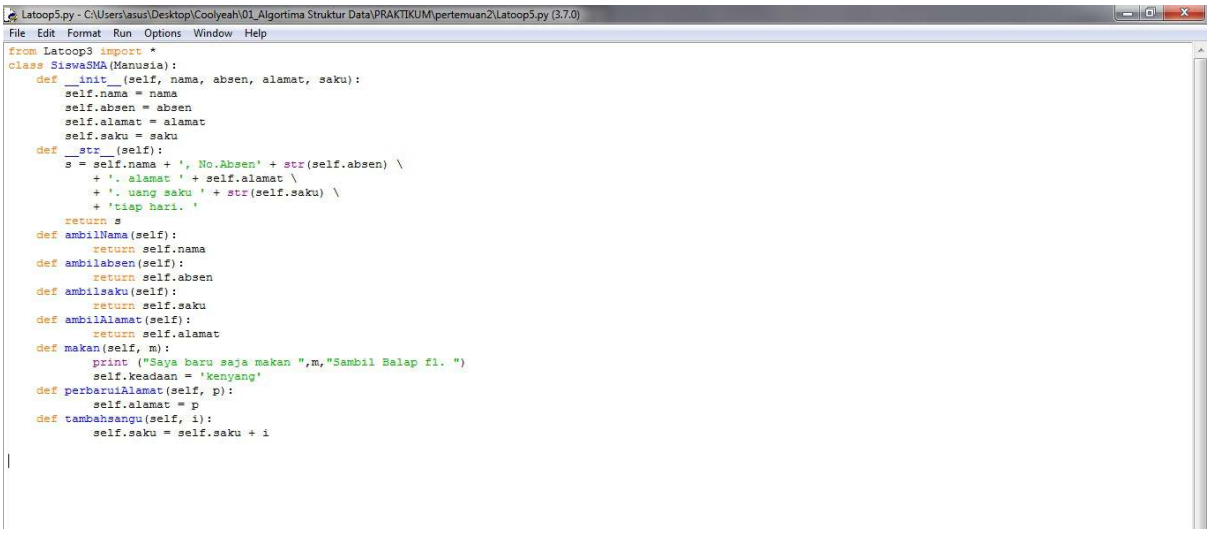
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pe
rtemuan2\Latoop2.py
>>> p = Pesan('indonesia adalah negeri yg indah')
>>> p.apakahTerkandung('klpo')
false
>>> p.apakahTerkandung('indo')
true
>>> p.konsonan()
19
>>> p.vokal()
13
>>>
```



```
Python 3.7.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pertemuan2\Latoop4.py
Salam fadhil
Nama:
Fadhil
NIM:
200
Kota:
Solo
Uang Saku:
2500000
>>> mhsbaru.ambilKota()
'Solo'
>>> mhsbaru.perbaruiKota('Surakarta')
>>> mhsbaru.ambilKota()
'Surakarta'
>>> mhsbaru.ambilsaku()
2500000
>>> mhsbaru.tambahsangu(1)
>>> mhsbaru.ambilsaku()
2500001
>>> mhsbaru.listkuliah
[]
>>> mhsbaru.ambilkuliah('matdis')
>>> mhsbaru.ambilkuliah('algostruk')
>>> mhsbaru.listkuliah
['matdis', 'algostruk']
>>> mhsbaru.kurangikuliah('matdis')
>>> mhsbaru.listkuliah
['algostruk']
>>>
```



```
Latoop4.py - C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pertemuan2\Latoop4.py (3.7.0)
File  Edit  Format  Run  Options  Window  Help
from Latoop3 import *
class Mahasiswa(Manusia):
        """docstring for Mahasiswa"""
        def __init__(self, nama, nim, kota, saku):
                self.nama = nama
                self.nim = nim
                self.kota = kota
                self.saku = saku
        def __str__(self):
                s = self.nama + ', NIM' + str(self.nim) \
                        + '. Tinggal di ' + self.kota \
                        + '. Uang Saku RP ' + str(self.saku) \
                        + ' tiap bulan. '
                return s

        def ambilNama(self):
                return self.nama
        def ambilnim(self):
                return self.nim
        def ambilsaku(self):
                return self.saku
        def ambilKota(self):
                return self.kota
        def makan(self, m):
                print ("Saya baru saja makan ",m,"Sambil Balap f1. ")
                self.keadaan = 'kenyang'
        def perbaruiKota(self, p):
                self.kota = p
        def tambahsangu(self, i):
                self.saku = self.saku + i
        listkuliah = []
        def ambilkuliah(self,kuliah):
                self.listkuliah.append(kuliah)
        def kurangikuliah(self, kuliah):
                self.listkuliah.remove(kuliah)

print ('Nama: ')
nama = input()
print ('NIM: ')
nim = input()
nim = int(nim)
print ('Kota: ')
kota = input()
print ('Uang Saku: ')
saku = input()
saku = int(saku)

mhsbaru = Mahasiswa(nama, nim, kota, saku)
Ln: 13  Col: 1
```



NO7

NIM berasal dari class Mahasiswa ambilNIM
berasal dari class Mahasiswa ambilNama
berasal dari class Mahasiswa ambilUangSaku
berasal dari class Mahasiswa katakanPy
berasal dari class MhsTIF
keadaan berasal dari class Manusia
kotaTinggal berasal dari class Mahasiswa
makan berasal dari class Manusia
mengalikanDenganDua berasal dari class Manusia
nama berasal dari class Mahasiswa dan Manusia
listKuliah berasal dari class Mahasiswa
ambilKuliah berasal dari class Mahasiswa
hapusKuliah berasal dari class Mahasiswa
tambahUangSaku berasal dari class Mahasiswa
uangSaku berasal dari class Mahasiswa
ambilKotaTinggal dari class Mahasiswa
olahraga berasal dari class Manusia
perbaruiKotaTinggal berasal dari class Mahasiswa
ucapkanSalam berasal dari class Manusia

Nama : Muhammad Khalif Rizaldi Wibowo

NIM : L200180217

Kelas : H

Mata Kuliah : Praktikum Algoritma dan Struktur Data

Modul 3

1.

```
#1
#konsistensi isi dan ukuran matriks
N = 5
M = 4

res = [ [ 0 for i in range(N) ] for j in range(M) ]

print("The matrix after initializing : " + str(res))

print (' ')

#ukuran matriks
res = [sum(len(row) > idx for row in B)
    for idx in range(max(map(len, B)))]

print ("The size of matrix : " + str(res))

print (' ')

#menjumlahkan dua matriks
for x in range(0, len(A)):
    for y in range(0, len(A[0])):
        print (A[x][y] + B[x][y], end=' ')
    print ()

print (' ')
```

```
#mengalikan dua matriks
X = []

for x in range(0, len(A)):
    row = []
    for y in range(0, len(A[0])):
        total = 0
        for z in range(0, len(A)):
            total = total + (A[x][z] * B[z][y])
        row.append(total)
    X.append(row)

for x in range(0, len(X)):
    for y in range(0, len(X[0])):
        print (X[x][y], end=' ')
    print ()

print (' ')


#menghitung determinan matriks
def determinantOfMatrix(A,n):
    temp = [0]*n
    total=1
    det=1

    for i in range(0,n):
        index=i

        while(A[index][i] == 0 and index < n):
            index+=1

        if(index == n):
            continue

        if(index != i):
            for j in range(0,n):
                A[index][j],A[i][j] = A[i][j],A[index][j]
            det = det*int(pow(-1,index-i))

        for j in range(0,n):
            temp[j] = A[i][j]

        for j in range(i+1,n):
            num1 = temp[i]
            num2 = A[j][i]

            for k in range(0,n):

                A[j][k] = (num1*A[j][k]) - (num2*temp[k])

            total = total * num1

    for i in range(0,n):
        det = det*A[i][i]


    return int(det/total)

print("Determinant of the matrix is : ",determinantOfMatrix(A,a))
```

2.

```
#2
#membangkitkan matrix 0
def buatNol(m):
    print ([[0 for j in range(m)] for i in range(m)])

#membangkitkan matrix identitas
def buatIdentitas(size):
    for row in range(0, size):
        for col in range(0, size):

            # Here end is used to stay in same line
            if (row == col):
                print("1 ", end=" ")
            else:
                print("0 ", end=" ")
        print()
```

3.

```
#3. Linked List
#mencari data isinya tertentu
class node:
    def __init__(self, next=None, data=None):
        self.next = next
        self.data = data
    def getdata(self):
        return self.data
    def setnext(self, newNext):
        self.next = newNext
    def recSearch(node, l, r, x):
        if r < l:
            return -1
        if node[l] == x:
            return l
        if node[r] == x:
            return r
        return LinkedList.recSearch(node, l+1, r-1, x)

#menambah suatu simpul diawal
    def tambahDepan(self, i):
        self.i = i
        node.append(i)

#menambah suatu simpul diakhir
    def tambahAkhir(self, i):
        self.i = i
        node.prepend(i)

#menyisipkan simpul dimana saja
class LinkedList:
    def __init__(self, head=None):
        self.head = head
    def tambah(self, prev, baru):
        baru.next = prev.next
        prev.next = baru
```

```
#menghapus simpul dimana saja
    def hapus(self, item):
        current = self.head
        previous = None
        found = False
        while current != None and not found:
            if current.getData() == item:
                found = True
                print(item, "Ditemukan")
            else:
                previous = current
                current = current.getNext()
        if found == False:
            print(item, "tidak Ditemukan")
        elif previous == None:
            self.head = current.getNext()
        else:
            previous.setNext(current.getNext())
```

4.

```
#4. Double Linked List
#mengunjungi dan mencetak data tiap simpul dari depan maupun belakang
    def cetakdepan(self):
        ini = self.head
        while ini is not None:
            print(ini.data)
            ini = ini.next

    def cetakbelakang(self):
        for i in data(len(data),0):
            return i

#menambah suatu simpul diawal
    def TambahDepan(self, i):
        self.i = i
        node.append(i)

#menambah suatu simpul diawal
    def tambahAkhir(self, i):
        self.i = i
        node.prepend(i)


a = node(2)
b = node(7)
c = node(15)
d = node(28)
e = node(33)
f = node(49)
g = node(56)

a.next = b
b.prev = a
b.next = c
c.prev = b
c.next = d
d.prev = c
d.next = e
e.prev = d
e.next = f
f.prev = e
```

```
a = node(2)
b = node(7)
c = node(15)
d = node(28)
e = node(33)
f = node(49)
g = node(56)

a.next = b
b.prev = a
b.next = c
c.prev = b
c.next = d
d.prev = c
d.next = e
e.prev = d
e.next = f
f.prev = e
f.next = g
g.prev = f

node = [2,7,15,28,33,49,56]
```

Nama : Muhammad Khalif Rizaldi Wibowo

Kelas : H

NIM : L200180217

Mata Kuliah : Praktikum Algoritma dan Struktur Data

Modul 4

modul4.py - D:/semester 4/Structure Data & Algorithm/modul4.py (3.8.2)

File Edit Format Run Options Window Help

```
class mhaTIF():
    def __init__(self, nama, asal, saku):
        self.nama = nama
        self.asal = asal
        self.saku = saku
c0 = mhaTIF('a','sukohuajo',240000)
c1 = mhaTIF('b','klaten', 260000)
c2 = mhaTIF('c','salatiga',250000)
c3 = mhaTIF('d','klaten',200000)
c4 = mhaTIF('e','surakarta',200000)
c5 = mhaTIF('f','salatiga',280000)
c6 = mhaTIF('g','sukoharjo',230000)

daftar = [c0,c1,c2,c3,c4,c5,c6]

print('\nNOMOR 1')
def cari(n):
    baru = []
    for i in range(len(n)):
        if(n[i].asal.lower() == 'klaten'):
            baru.append(i)
    return baru
print(cari(daftar))

print('\nNOMOR 2')
def sakuKcl(n):
    baru = n[0].saku
    for i in range(len(n)):
        if(n[i].saku<baru):
            baru = n[i].saku
    return baru
print(sakuKcl(daftar))

print('\nNOMOR 3')
def sakuKcl2(n):
    baru = n[0].saku
    list = []
    for i in range(len(n)):
        if(n[i].saku==baru):
            list.append(n[i].nama)
        elif(n[i].saku<baru):
            baru = n[i].saku
```

modul4.py - D:/semester 4/Structure Data & Algorythm/modul4.py (3.8.2)

File Edit Format Run Options Window Help

```
        elif(n[i].saku<baru):
            baru = n[i].saku
            list = []
            list.append(n[i].nama)
    return list
print(sakuKol2(daftar))

print('\nNOMOR 4')
def sakuKrg(n):
    batas = 250000
    list = []
    for i in range(len(n)):
        if(n[i].saku < batas):
            list.append(n[i].nama)
    return list
print(sakuKrg(daftar))

print('\nNOMOR 5')
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def pushNw(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
        return self.head
    def search(self, x):
        current = self.head
        while current != None:
            if current.data == x:
                return "True"
            current = current.next
        return "False"
    def display(self):
        current = self.head
        while current is not None:
            print(current.data, end = ' ')
            current = current.next
```

modul4.py - D:/semester 4/Structure Data & Algorythm/modul4.py (3.8.2)

File Edit Format Run Options Window Help

```
            print(current.data, end = ' ')
            current = current.next

llist = LinkedList()
llist.pushAw(21)
llist.pushAw(22)
llist.pushAw(12)
llist.pushAw(14)
llist.pushAw(2)
llist.pushAw(19)
print(llist.search(21))
print(llist.search(29))

print('\nNOMOR 6')
def binSe(list, target):
    low = 0
    high = len(list) - 1
    while(low<=high):
        mid = (low+high)//2
        if(list[mid] == target):
            return "target di index "+str(mid)
        elif(target<list[mid]):
            high = mid - 1
        else:
            low = mid +1
    return "target tidak ditemukan di index berapapun"
list = [2,4,6,9,12,27,39,46,59,77]
target = 12
print(binSe(list,target))
list = [2,4,6,9,12,27,39,46,59,77]
target = 133
print(binSe(list,target))

print('\nNOMOR 7')
def binSe(kumpulan, target):
    temp = []
    low = 0
    high = len(kumpulan)-1
    while low <= high :
        mid = (high+low)//2
        if kumpulan[mid] == target:
            midKiri = mid-1
```

```
print('\nNOMOR 7')
def binSe(kumpulan, target):
    temp = []
    low = 0
    high = len(kumpulan)-1
    while low <= high :
        mid = (high+low)//2
        if kumpulan[mid] == target:
            midKiri = mid-1
            while kumpulan[midKiri] == target:
                temp.append(midKiri)
                midKiri = midKiri-1
            temp.append(mid)
            midKanan = mid+1
            while kumpulan[midKanan] == target:
                temp.append(midKanan)
                midKanan = midKanan+1
            return temp
        elif target < kumpulan[mid]:
            high = mid-1
        else:
            low = mid+1
    return False
kumpulan = [2,3,4,5,8,8,9,12]
target = 8
print(binSe(kumpulan,target))
```


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: D:/semester 4/Structure Data & Algorythm/modul4.py ==========
NOMOR 1
[1, 3]
NOMOR 2
200000
NOMOR 3
['d', 'e']
NOMOR 4
['a', 'd', 'e', 'g']
NOMOR 5
True
False
NOMOR 6
target di index 4
target tidak ditemukan di index berapapun
NOMOR 7
[4, 5]
>>>

Nama : Muhammad Khalif Rizaldi Wibowo

NIM : L200180217

Kelas : H

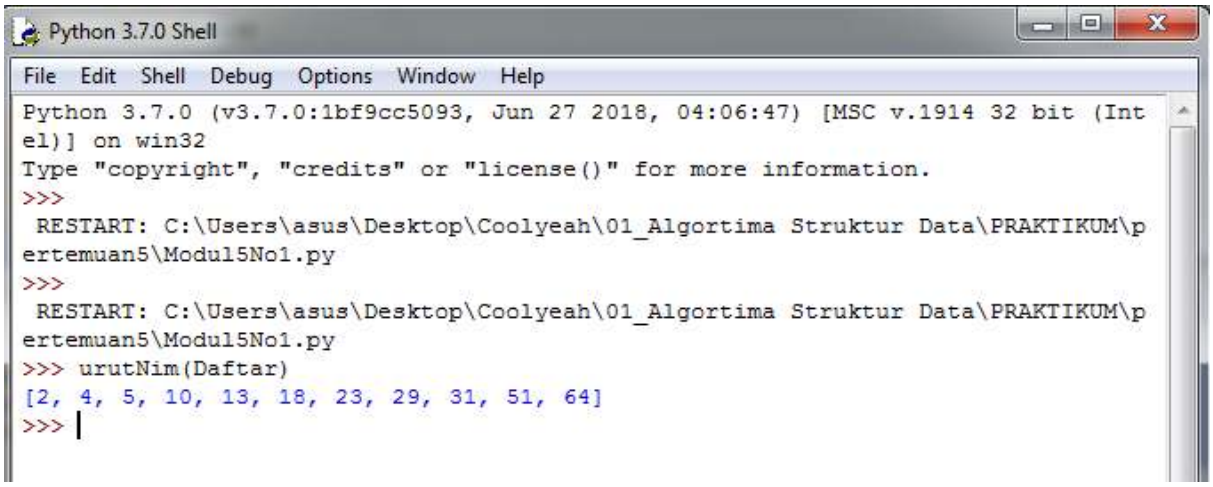
Mata Kuliah : Praktikum Algoritma dan Struktur Data

Modul 5

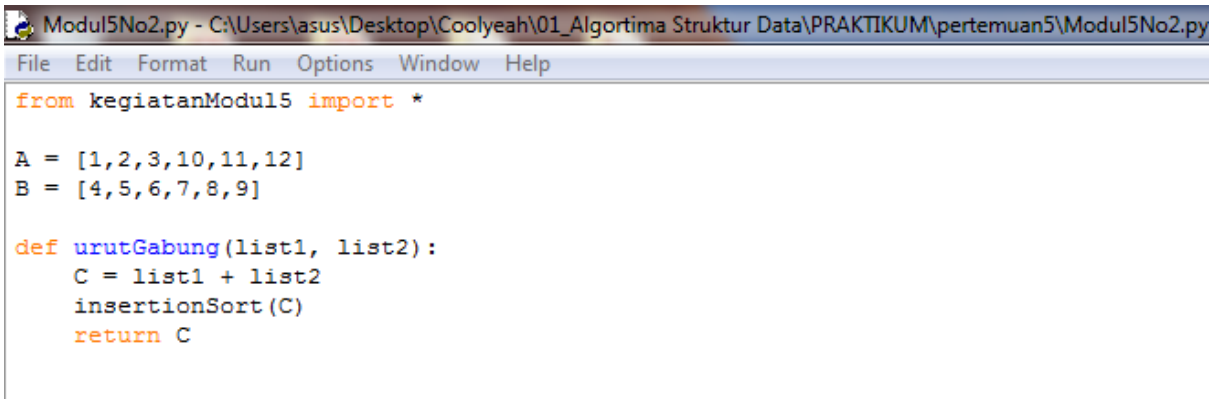
1.

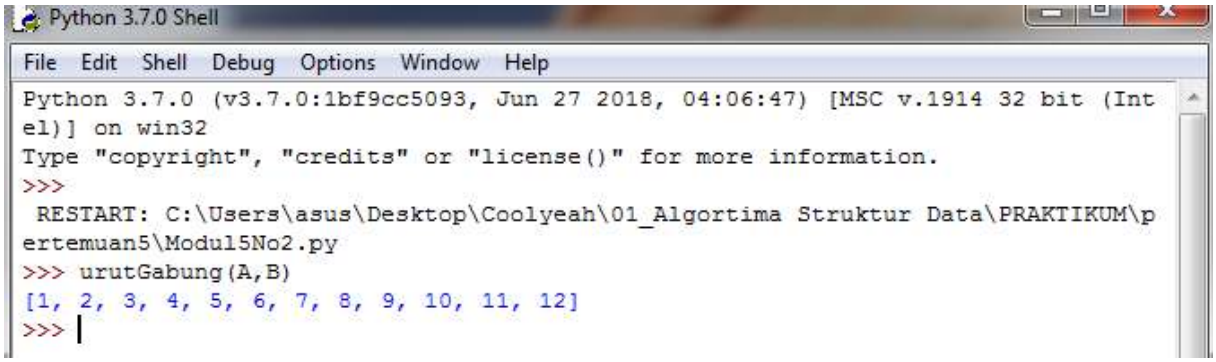
File Edit Format Run Options Window Help

```
from kegiatanModul5 import *

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

c0 = MhsTIF("Ika", 10, "Sukoharjo", 240000)
c1 = MhsTIF("Budi", 51, "Sragen", 230000)
c2 = MhsTIF("Ahmad", 2, "Surakarta", 250000)
c3 = MhsTIF("Chandra", 18, "Surakarta", 235000)
c4 = MhsTIF("Eka", 4, "Boyolali", 240000)
c5 = MhsTIF("Fandi", 31, "Salatiga", 250000)
c6 = MhsTIF("Deni", 13, "Klaten", 245000)
c7 = MhsTIF("Galuh", 5, "Wonogiri", 245000)
c8 = MhsTIF("Janto", 23, "Klaten", 245000)
c9 = MhsTIF("Hasan", 64, "Karanganyar", 270000)
c10 = MhsTIF("Khalid", 29, "Purwodadi", 265000)

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

def urutNim(wadah):
    nim = []
    for i in wadah:
        nim.append(i.nim)
    insertionSort(nim)
    return nim
```


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pertemuan5\Modul5No1.py
>>>
RESTART: C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pertemuan5\Modul5No1.py
>>> urutNim(Daftar)
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>>

No.2


Modul5No2.py - C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pertemuan5\Modul5No2.py
File Edit Format Run Options Window Help
from kegiatanModul5 import *
A = [1,2,3,10,11,12]
B = [4,5,6,7,8,9]
def urutGabung(list1, list2):
C = list1 + list2
insertionSort(C)
return C

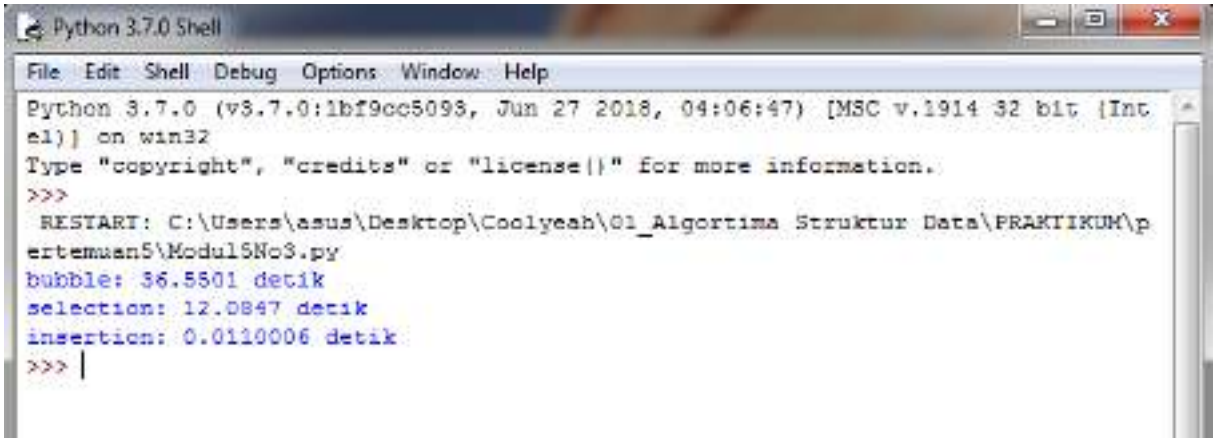

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pertemuan5\Modul5No2.py
>>> urutGabung(A,B)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
>>>

No.3

Modul5No3.py - C:\Users\asus\Desktop\Coolyeah\01_Algortima Struktur Data\PRAKTIKUM\pertemuan5\Modul5No3.py (3.7

File Edit Format Run Options Window Help

```
from time import time as detak
from random import shuffle as kocok
from kegiatanModul5 import *

k = list(range(1,6001))
kocok(k)
u_bub = k[:]
u_sel = k[:]
u_ins = 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) );
```



Insertion lebih cepat , di urutan kedua ada selection sort dan yang palin lambat adalah Bubble sort

Nama : Muhammad Khalif Rizaldi Wibowo

Kelas : H

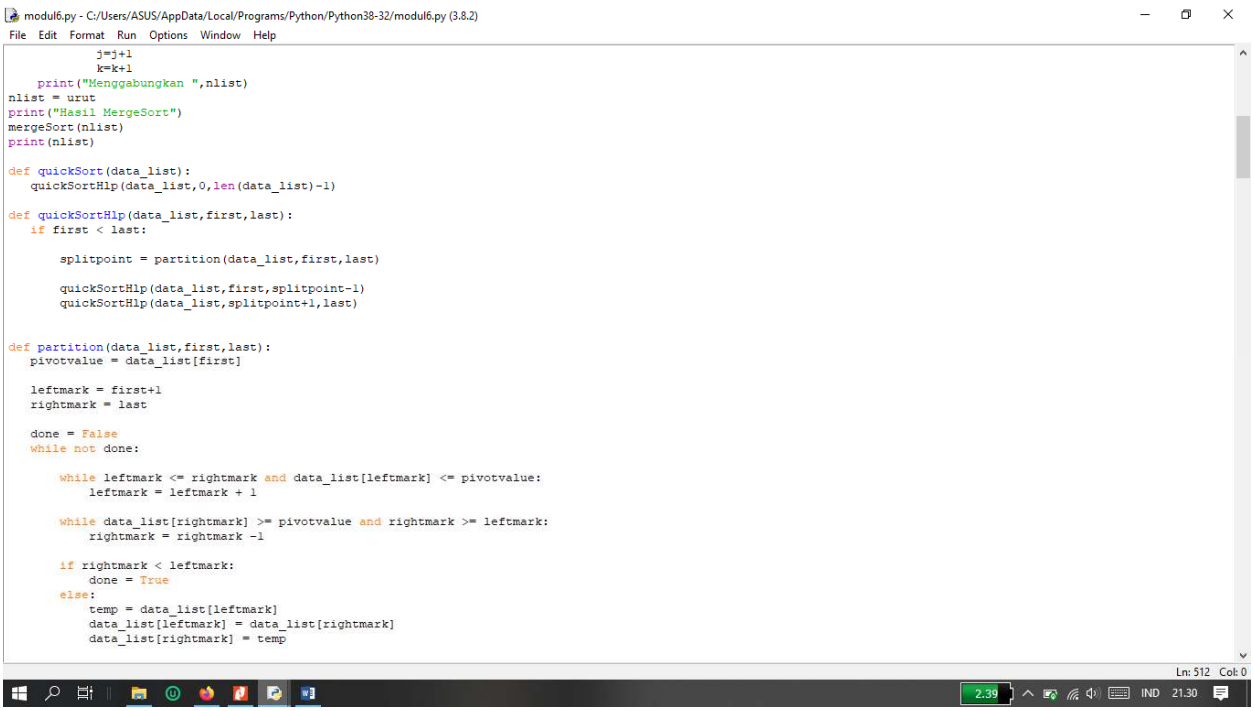
NIM : L200180217

Praktikum Algoritma dan Struktur Data

Modul 6

```
#NO 1
print ('No 1: ')
class MhsTIF(object) :
    def __init__(self, nama, nim, asal, uangsaku) :
        self.nama = nama
        self.nim = nim
        self.asal = asal
        self.uangsaku = uangsaku

m0 = MhsTIF('Alfianto', 9, 'Boylali', 300000)
m1 = MhsTIF('Hari', 10, 'Semarang', 320000)
m2 = MhsTIF('Mifta', 23, 'Kartasura', 350000)
m3 = MhsTIF('Desi', 45, 'Solo', 290000)
m4 = MhsTIF('Dewi', 27, 'Karanganyar', 310000)
m5 = MhsTIF('Lia', 56, 'Wonogiri', 380000)
m6 = MhsTIF('Bagus', 2, 'Boyolali', 280000)
m7 = MhsTIF('Wahyu', 8, 'Sragen', 330000)
m8 = MhsTIF('Lusiana', 34, 'Purwodadi', 340000)
m9 = MhsTIF('Alfina', 60, 'Sleman', 390000)
m10 = MhsTIF('Akbar', 51, 'Magelang', 370000)

urut =[m0.nim, m1.nim, m2.nim, m3.nim, m4.nim, m5.nim,
       m6.nim, m7.nim,m8.nim, m9.nim, m10.nim]

def mergeSort(nlist):
    print("Membelah ", nlist)
    if len(nlist)>1:
        mid = len(nlist)//2
        lefthalf = nlist[:mid]
        righthalf = nlist[mid:]

        mergeSort(lefthalf)
        mergeSort(righthalf)
        i=j=k=0
        while i < len(lefthalf) and j < len(righthalf):
            if lefthalf[i] < righthalf[j]:
                nlist[k]=lefthalf[i]
                i=i+1
            else:
                nlist[k]=righthalf[j]
                j=j+1
            k=k+1
```

```
                j=j+1
            k=k+1

        while i < len(lefthalf):
            nlist[k]=lefthalf[i]
            i=i+1
            k=k+1

        while j < len(righthalf):
            nlist[k]=righthalf[j]
            j=j+1
            k=k+1
    print("Menggabungkan ",nlist)
nlist = urut
print("Hasil MergeSort")
mergeSort(nlist)
print(nlist)

def quickSort(data_list):
   quickSortHlp(data_list,0,len(data_list)-1)

def quickSortHlp(data_list,first,last):
   if first < last:

       splitpoint = partition(data_list,first,last)

       quickSortHlp(data_list,first,splitpoint-1)
       quickSortHlp(data_list,splitpoint+1,last)


def partition(data_list,first,last):
   pivotvalue = data_list[first]

   leftmark = first+1
   rightmark = last

   done = False
   while not done:

       while leftmark <= rightmark and data_list[leftmark] <= pivotvalue:
           leftmark = leftmark + 1
```



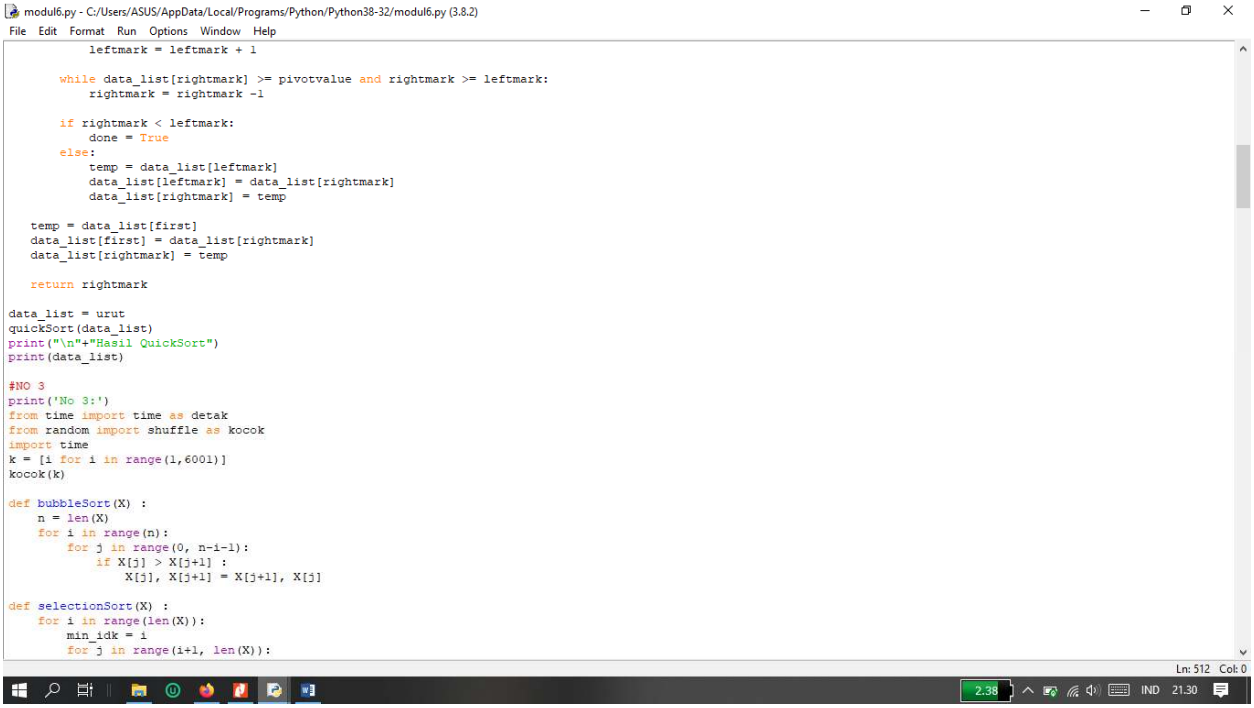
modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)

File Edit Format Run Options Window Help

```
            j=j+1
            k=k+1
    print("Menggabungkan ",nlist)
nlist = urut
print("Hasil MergeSort")
mergeSort(nlist)
print(nlist)

def quickSort(data_list):
   quickSortHlp(data_list,0,len(data_list)-1)

def quickSortHlp(data_list,first,last):
   if first < last:

       splitpoint = partition(data_list,first,last)

       quickSortHlp(data_list,first,splitpoint-1)
       quickSortHlp(data_list,splitpoint+1,last)


def partition(data_list,first,last):
   pivotvalue = data_list[first]

   leftmark = first+1
   rightmark = last

   done = False
   while not done:

       while leftmark <= rightmark and data_list[leftmark] <= pivotvalue:
           leftmark = leftmark + 1

       while data_list[rightmark] >= pivotvalue and rightmark >= leftmark:
           rightmark = rightmark -1

       if rightmark < leftmark:
           done = True
       else:
           temp = data_list[leftmark]
           data_list[leftmark] = data_list[rightmark]
           data_list[rightmark] = temp
```

Ln: 512 Col: 0

2.39 IND 21.30



modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)

File Edit Format Run Options Window Help

```
            leftmark = leftmark + 1

       while data_list[rightmark] >= pivotvalue and rightmark >= leftmark:
           rightmark = rightmark -1

       if rightmark < leftmark:
           done = True
       else:
           temp = data_list[leftmark]
           data_list[leftmark] = data_list[rightmark]
           data_list[rightmark] = temp

   temp = data_list[first]
   data_list[first] = data_list[rightmark]
   data_list[rightmark] = temp

   return rightmark

data_list = urut
quickSort(data_list)
print("\n"+"Hasil QuickSort")
print(data_list)

#NO 3
print('No 3:')
from time import time as detak
from random import shuffle as kocok
import time
k = [i for i in range(1,6001)]
kocok(k)

def bubbleSort(X) :
    n = len(X)
    for i in range(n):
        for j in range(0, n-i-1):
            if X[j] > X[j+1] :
                X[j], X[j+1] = X[j+1], X[j]

def selectionSort(X) :
    for i in range(len(X)):
        min_idx = i
        for j in range(i+1, len(X)):
```

Ln: 512 Col: 0

2.38 IND 21.30

modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)

File Edit Format Run Options Window Help

```
def insertSort(X) :
    n = len (X)
    for i in range (1, n) :
        nilai = X[i]
        abc = i-1
        while abc >= 0 and nilai < X[abc-1] :
            X[abc] = X[abc+1]
            abc -=1
        X[abc+1] = nilai


def mergeSort(X):
    if len(X) >1:
        mid = len(X)//2
        L = X[:mid]
        R = X[mid:]
        mergeSort(L)
        mergeSort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                X[k] = L[i]
                i+=1
            else:
                X[k] = R[j]
                j+=1
            k+=1
        while i < len(L):
            X[k] = L[i]
            i+=1
            k+=1
        while j < len(R):
            X[k] = R[j]
            j+=1
            k+=1
def partition(X,low,high):
    i = ( low-1 )
    pivot = X[high]
    for j in range(low , high):
        if   X[j] <= pivot:
            i = i+1
```

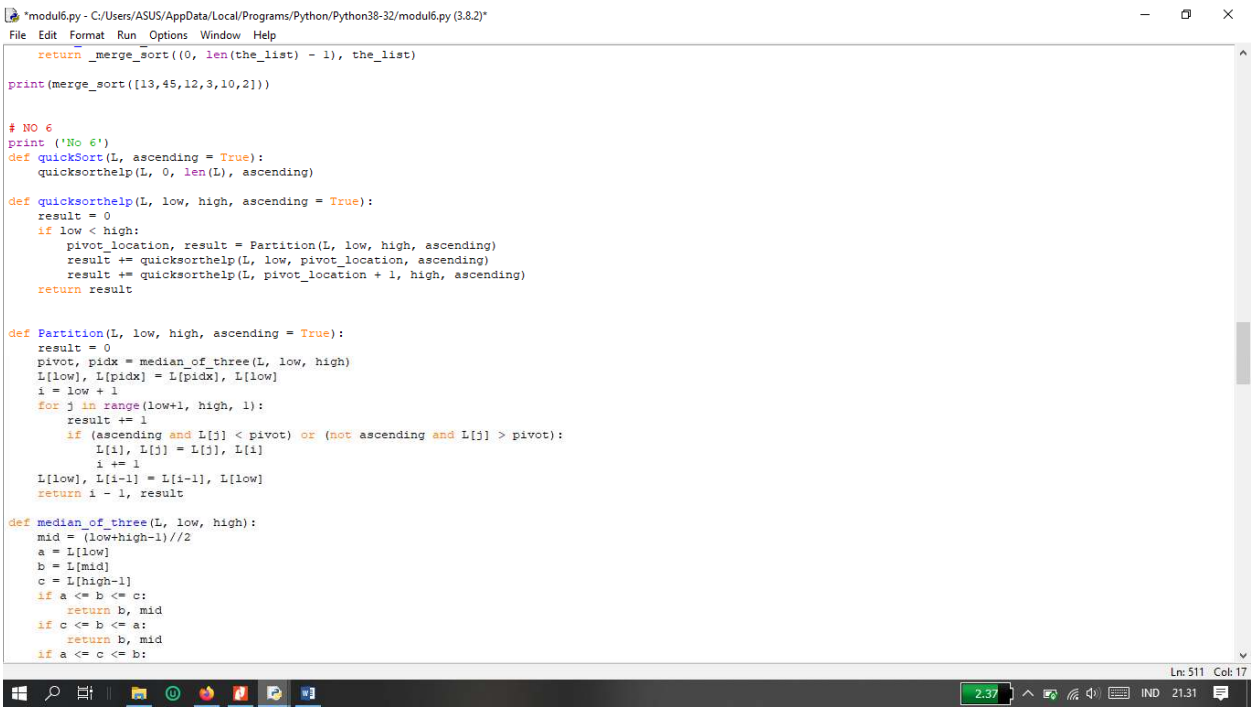
Ln: 512 Col: 0

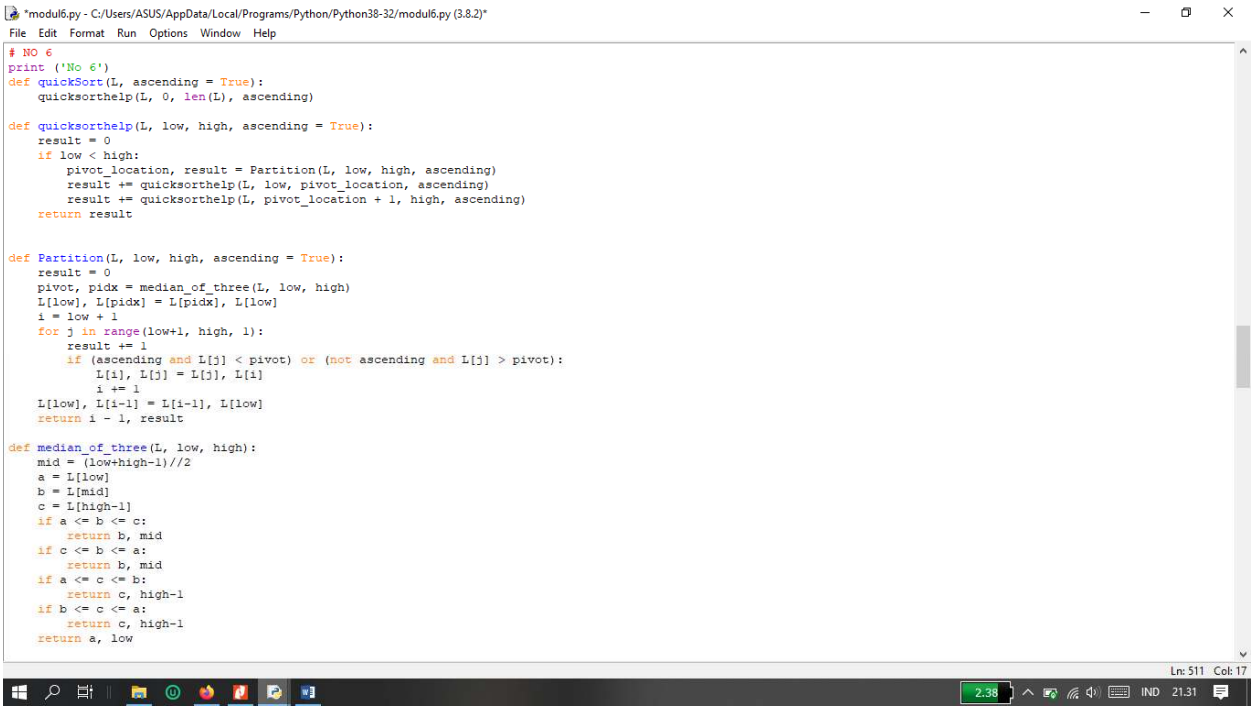
modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)

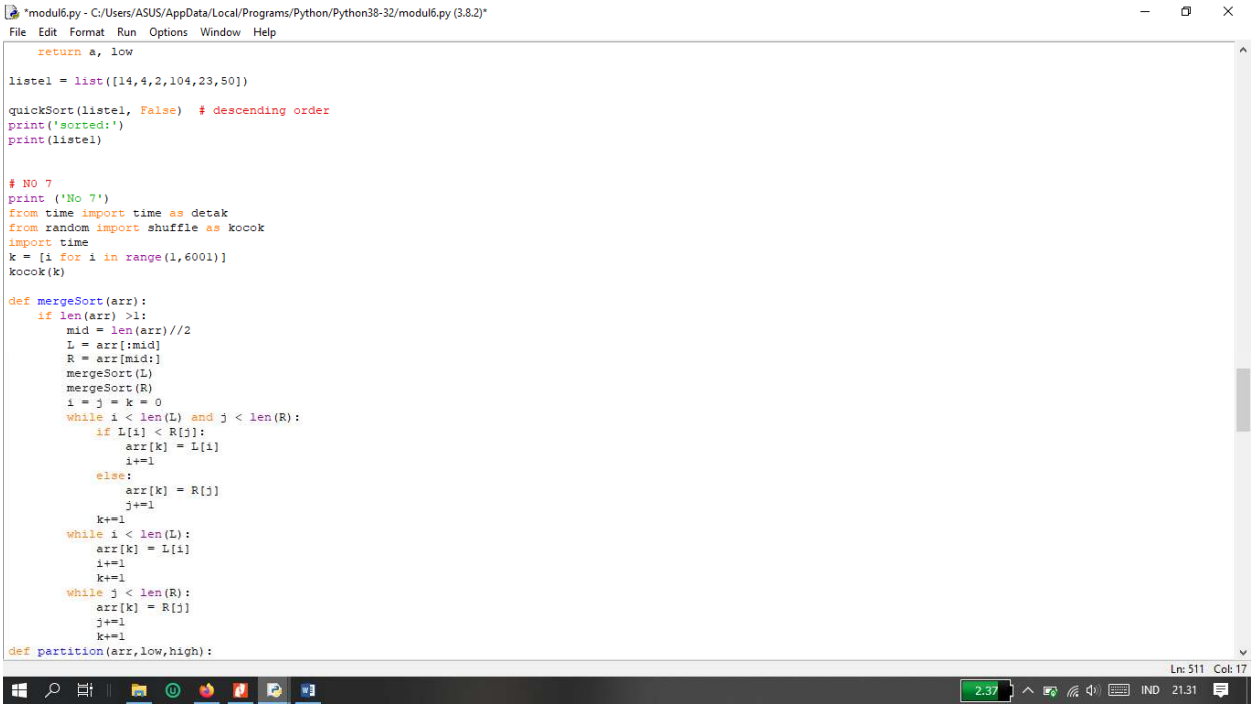
File Edit Format Run Options Window Help

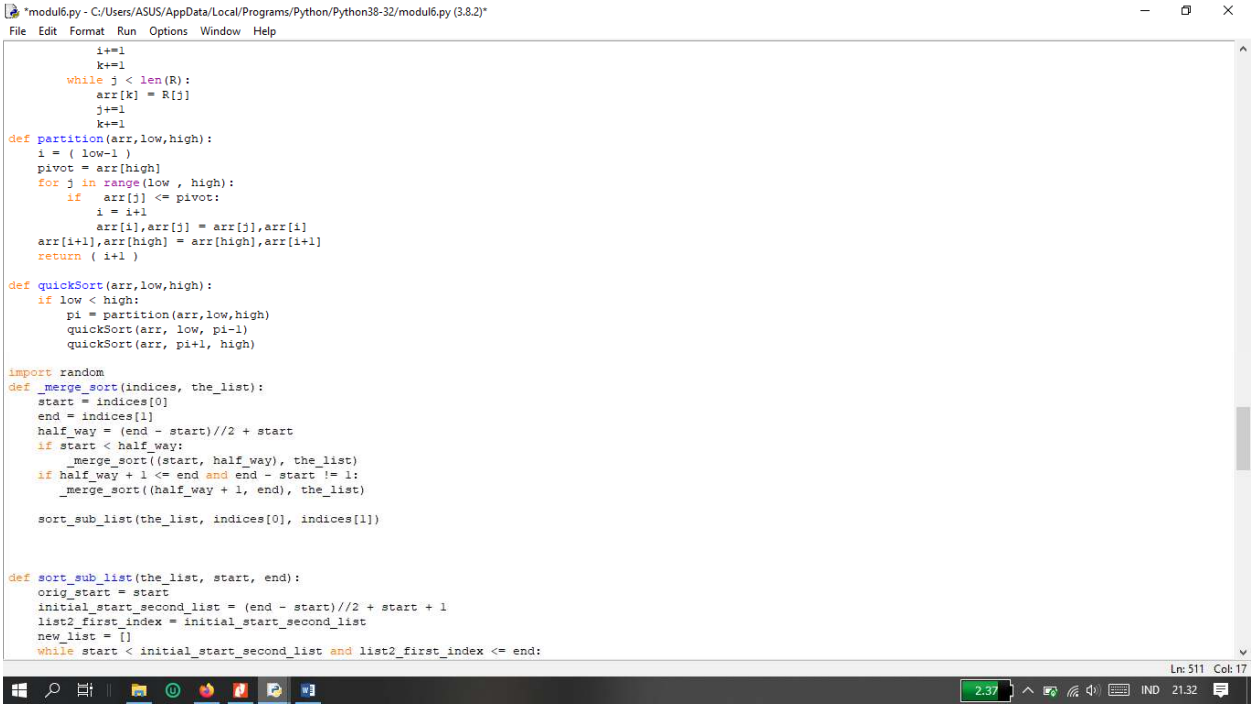
```
# NO 5
print ('No 5')
import random
def _merge_sort(indices, the_list):
    start = indices[0]
    end = indices[1]
    half_way = (end - start)//2 + start
    if start < half_way:
        _merge_sort((start, half_way), the_list)
    if half_way + 1 <= end and end - start != 1:
       _merge_sort((half_way + 1, end), the_list)

    sort_sub_list(the_list, indices[0], indices[1])
    return the_list


def sort_sub_list(the_list, start, end):
    orig_start = start
    initial_start_second_list = (end - start)//2 + start + 1
    list2_first_index = initial_start_second_list
    new_list = []
    while start < initial_start_second_list and list2_first_index <= end:
        first1 = the_list[start]
        first2 = the_list[list2_first_index]
        if first1 > first2:
            new_list.append(first2)
            list2_first_index += 1
        else:
            new_list.append(first1)
            start += 1
    while start < initial_start_second_list:
        new_list.append(the_list[start])
        start += 1

    while list2_first_index <= end:
        new_list.append(the_list[list2_first_index])
        list2_first_index += 1
    for i in new_list:
        the_list[orig_start] = i
        orig_start += 1
    return the_list
```

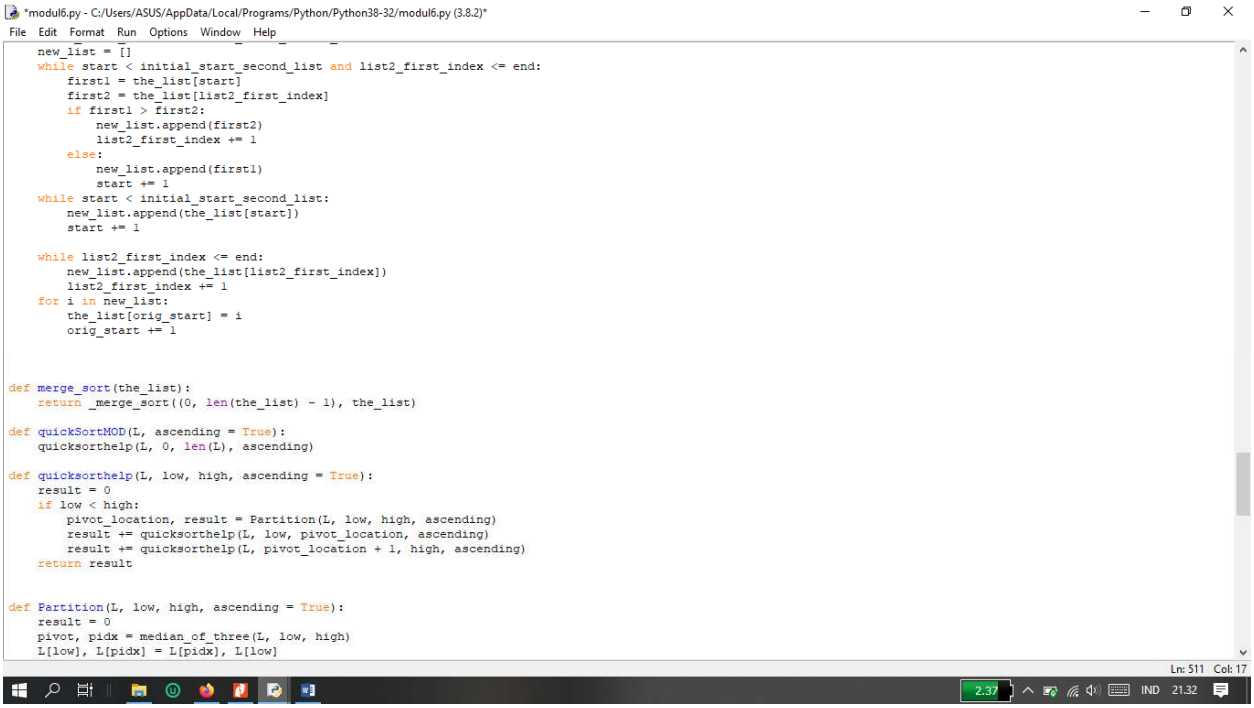
Ln: 511 Col: 17

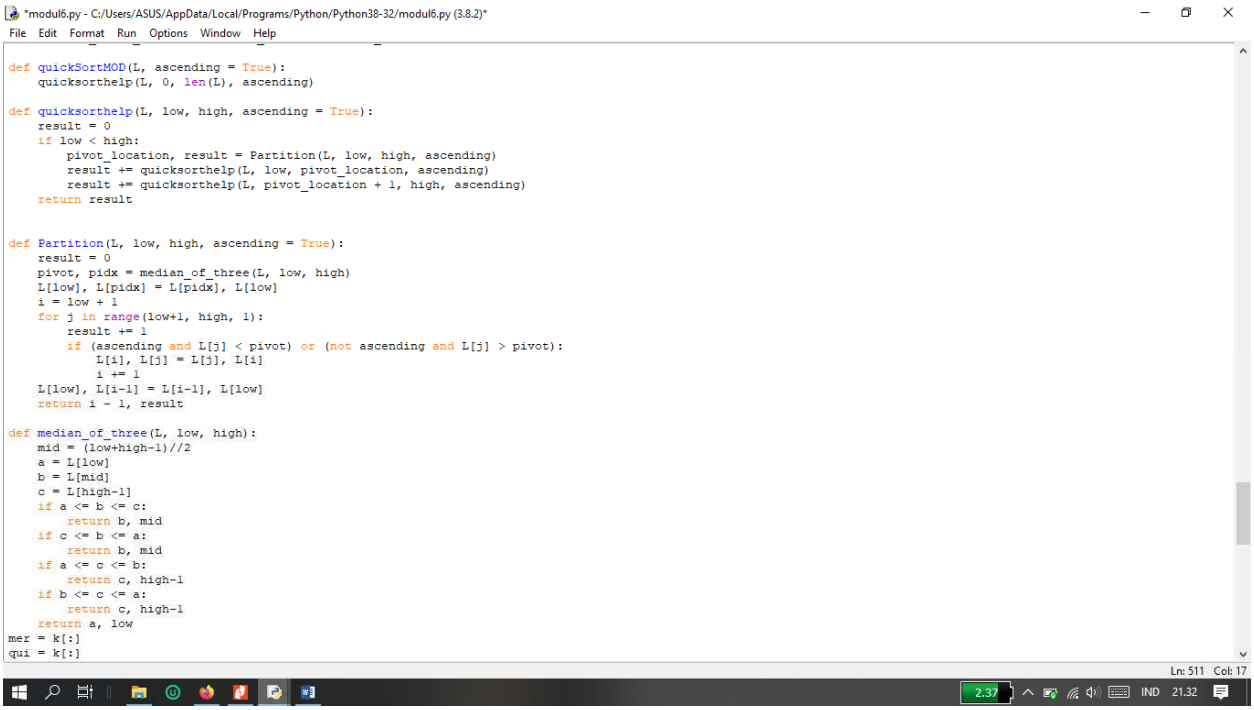

modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)
File Edit Format Run Options Window Help
return _merge_sort((0, len(the_list) - 1), the_list)
print(merge_sort([13,45,12,3,10,2]))
# NO 6
print ('No 6')
def quickSort(L, ascending = True):
quicksorthelp(L, 0, len(L), ascending)
def quicksorthelp(L, low, high, ascending = True):
result = 0
if low < high:
pivot_location, result = Partition(L, low, high, ascending)
result += quicksorthelp(L, low, pivot_location, ascending)
result += quicksorthelp(L, pivot_location + 1, high, ascending)
return result
def Partition(L, low, high, ascending = True):
result = 0
pivot, pidx = median_of_three(L, low, high)
L[low], L[pidx] = L[pidx], L[low]
i = low + 1
for j in range(low+1, high, 1):
result += 1
if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot):
L[i], L[j] = L[j], L[i]
i += 1
L[low], L[i-1] = L[i-1], L[low]
return i - 1, result
def median_of_three(L, low, high):
mid = (low+high-1)//2
a = L[low]
b = L[mid]
c = L[high-1]
if a <= b <= c:
return b, mid
if c <= b <= a:
return b, mid
if a <= c <= b:
Ln: 511 Col: 17
2.37
IND 21.31


modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)
File Edit Format Run Options Window Help
NO 6
print ('No 6')
def quickSort(L, ascending = True):
quicksorthelp(L, 0, len(L), ascending)
def quicksorthelp(L, low, high, ascending = True):
result = 0
if low < high:
pivot_location, result = Partition(L, low, high, ascending)
result += quicksorthelp(L, low, pivot_location, ascending)
result += quicksorthelp(L, pivot_location + 1, high, ascending)
return result
def Partition(L, low, high, ascending = True):
result = 0
pivot, pidx = median_of_three(L, low, high)
L[low], L[pidx] = L[pidx], L[low]
i = low + 1
for j in range(low+1, high, 1):
result += 1
if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot):
L[i], L[j] = L[j], L[i]
i += 1
L[low], L[i-1] = L[i-1], L[low]
return i - 1, result
def median_of_three(L, low, high):
mid = (low+high-1)//2
a = L[low]
b = L[mid]
c = L[high-1]
if a <= b <= c:
return b, mid
if c <= b <= a:
return b, mid
if a <= c <= b:
return c, high-1
if b <= c <= a:
return c, high-1
return a, low
Ln: 511 Col: 17
2.38
IND 21.31


modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)
File Edit Format Run Options Window Help
return a, low
listel = list([14,4,2,104,23,50])
quickSort(listel, False) # descending order
print('sorted:')
print(listel)
# NO 7
print ('No 7')
from time import time as detak
from random import shuffle as kocok
import time
k = [i for i in range(1,6001)]
kocok(k)
def mergeSort(arr):
if len(arr) >1:
mid = len(arr)//2
L = arr[:mid]
R = arr[mid:]
mergeSort(L)
mergeSort(R)
i = j = k = 0
while i < len(L) and j < len(R):
if L[i] < R[j]:
arr[k] = L[i]
i+=1
else:
arr[k] = R[j]
j+=1
k+=1
while i < len(L):
arr[k] = L[i]
i+=1
k+=1
while j < len(R):
arr[k] = R[j]
j+=1
k+=1
def partition(arr,low,high):
Ln: 511 Col: 17
2.37
IND 21.31


modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)
File Edit Format Run Options Window Help
i+=1
k+=1
while j < len(R):
arr[k] = R[j]
j+=1
k+=1
def partition(arr,low,high):
i = (low-1)
pivot = arr[high]
for j in range(low , high):
if arr[j] <= pivot:
i = i+1
arr[i],arr[j] = arr[j],arr[i]
arr[i+1],arr[high] = arr[high],arr[i+1]
return (i+1)
def quickSort(arr,low,high):
if low < high:
pi = partition(arr,low,high)
quickSort(arr, low, pi-1)
quickSort(arr, pi+1, high)
import random
def _merge_sort(indices, the_list):
start = indices[0]
end = indices[1]
half_way = (end - start)//2 + start
if start < half_way:
_merge_sort((start, half_way), the_list)
if half_way + 1 <= end and end - start != 1:
_merge_sort((half_way + 1, end), the_list)
sort_sub_list(the_list, indices[0], indices[1])
def sort_sub_list(the_list, start, end):
orig_start = start
initial_start_second_list = (end - start)//2 + start + 1
list2_first_index = initial_start_second_list
new_llst = []
while start < initial_start_second_list and list2_first_index <= end:
Ln: 511 Col: 17
2.37
IND 21.32


modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)
File Edit Format Run Options Window Help
new_list = []
while start < initial_start_second_list and list2_first_index <= end:
first1 = the_list[start]
first2 = the_list[list2_first_index]
if first1 > first2:
new_list.append(first2)
list2_first_index += 1
else:
new_list.append(first1)
start += 1
while start < initial_start_second_list:
new_list.append(the_list[start])
start += 1
while list2_first_index <= end:
new_list.append(the_list[list2_first_index])
list2_first_index += 1
for i in new_list:
the_list[orig_start] = i
orig_start += 1
def merge_sort(the_list):
return _merge_sort((0, len(the_list) - 1), the_list)
def quickSortMOD(L, ascending = True):
quicksorthelp(L, 0, len(L), ascending)
def quicksorthelp(L, low, high, ascending = True):
result = 0
if low < high:
pivot_location, result = Partition(L, low, high, ascending)
result += quicksorthelp(L, low, pivot_location, ascending)
result += quicksorthelp(L, pivot_location + 1, high, ascending)
return result
def Partition(L, low, high, ascending = True):
result = 0
pivot, pidx = median_of_three(L, low, high)
L[low], L[pidx] = L[pidx], L[low]
Ln: 511 Col: 17
2.37
IND 21.32


modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)
File Edit Format Run Options Window Help
def quickSortMOD(L, ascending = True):
quicksorthelp(L, 0, len(L), ascending)
def quicksorthelp(L, low, high, ascending = True):
result = 0
if low < high:
pivot_location, result = Partition(L, low, high, ascending)
result += quicksorthelp(L, low, pivot_location, ascending)
result += quicksorthelp(L, pivot_location + 1, high, ascending)
return result
def Partition(L, low, high, ascending = True):
result = 0
pivot, pidx = median_of_three(L, low, high)
L[low], L[pidx] = L[pidx], L[low]
i = low + 1
for j in range(low+1, high, 1):
result += 1
if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot):
L[i], L[j] = L[j], L[i]
i += 1
L[low], L[i-1] = L[i-1], L[low]
return i - 1, result
def median_of_three(L, low, high):
mid = (low+high-1)//2
a = L[low]
b = L[mid]
c = L[high-1]
if a <= b <= c:
return b, mid
if c <= b <= a:
return b, mid
if a <= c <= b:
return c, high-1
if b <= c <= a:
return c, high-1
return a, low
mer = k[:]
qui = k[:]
Ln: 511 Col: 17
2.37
IND 21.32

modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)

File Edit Format Run Options Window Help

```
def median_of_three(L, low, high):
    mid = (low+high-1)//2
    a = L[low]
    b = L[mid]
    c = L[high-1]
    if a <= b <= c:
        return b, mid
    if c <= b <= a:
        return b, mid
    if a <= c <= b:
        return c, high-1
    if b <= c <= a:
        return c, high-1
    return a, low
mer = k[:]
qui = k[:]
mer2 = k[:]
qui2 = k[:]


aw=detak();mergeSort(mer);ak=detak();print('merge : %g detik' %(ak-aw));
aw=detak();quickSort(qui,0,len(qui)-1);ak=detak();print('quick : %g detik' %(ak-aw));
aw=detak();merge_sort(mer2);print('merge mod : %g detik' %(ak-aw));
aw=detak();quickSortMOD(qui2, False);print('quick mod : %g detik' %(ak-aw));


# NO 8
print ('Nom 8')
class Node:
  def __init__(self, data):
    self.data = data
    self.next = None

class LinkedList:
  def __init__(self):
    self.head = None

  def appendList(self, data):
    node = Node(data)
    if self.head == None:
      self.head = node
```

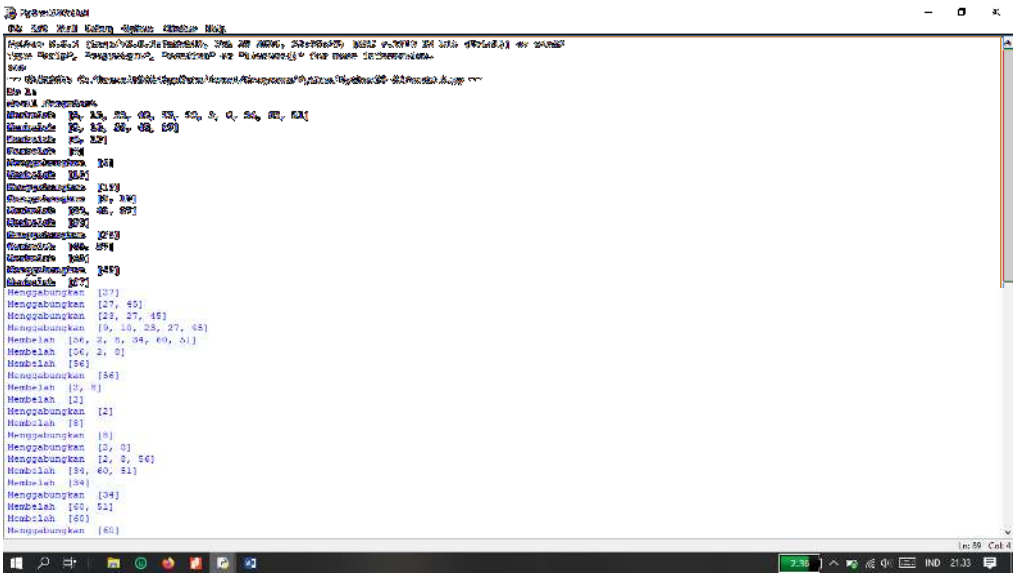
Ln: 511 Col: 17

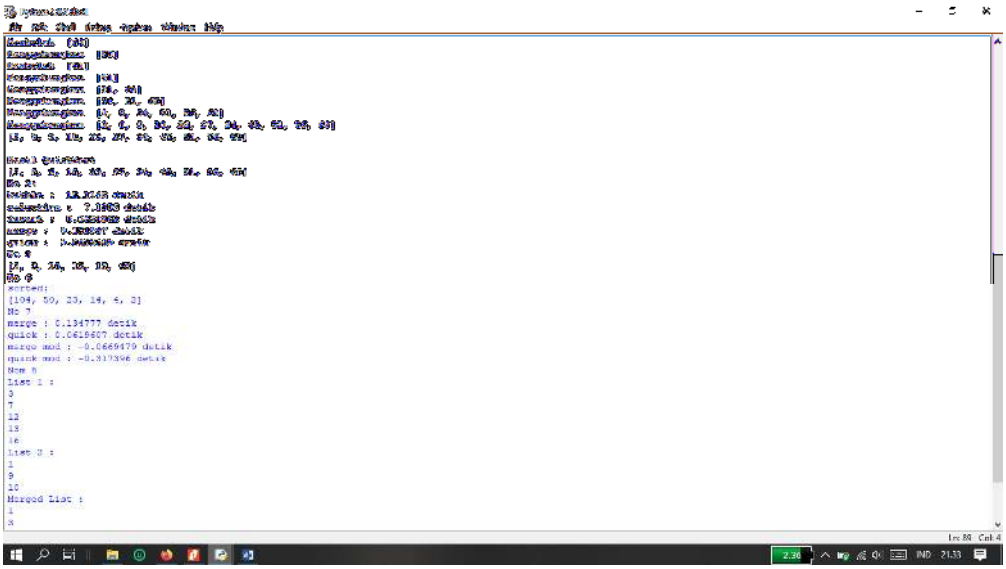
modul6.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul6.py (3.8.2)

File Edit Format Run Options Window Help

```
    curr = self.head
    while curr != None:
      print ("%d"%curr.data),
      curr = curr.next
  def mergeSorted(self, list1, list2):
    if list1 is None:
      return list2
    if list2 is None:
      return list1

    if list1.data < list2.data:
      temp = list1
      temp.next = self.mergeSorted(list1.next, list2)
    else:
      temp = list2
      temp.next = self.mergeSorted(list1, list2.next)
    return temp


list1 = LinkedList()
list1.appendSorted(13)
list1.appendSorted(12)
list1.appendSorted(3)
list1.appendSorted(16)
list1.appendSorted(7)

print("List 1 :"),
list1.printList()

list2 = LinkedList()
list2.appendSorted(9)
list2.appendSorted(10)
list2.appendSorted(1)

print("List 2 :"),
list2.printList()

list3 = LinkedList()
list3.head = list3.mergeSorted(list1.head, list2.head)

print("Merged List :"),
list3.printList()
```

Ln: 511 Col: 17





```
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>
```

Nama : Muhammad Khalif Rizaldi Wibowo

Kelas : H

NIM : L200180217

Praktikum Algoritma dan Struktur Data

modul7.py - C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul7.py (3.8.2)

File Edit Format Run Options Window Help

```
print("No 1----------------------------------------------------------------------------------------------------------------------------------------")
#Nomor 1
import re
f=open('indonesia.txt','r', encoding='latin1')
teks = f.read()
f.close()
print(re.findall(r'me\w+',teks.lower()))
print("\n")


#Nomor 2
print("No 2----------------------------------------------------------------------------------------------------------------------------------------")
import re
f=open('indonesia.txt','r', encoding='latin1')
teks2 = f.read()
f.close()
print(re.findall(r"di\w+",teks2.lower()))
print("\n")

#Nomor 3
print("No 3----------------------------------------------------------------------------------------------------------------------------------------")
import re
f=open('indonesia.txt','r', encoding='latin1')
teks3 = f.read()
f.close()
print(re.findall(r"di \w+",teks3))
print("\n")

#Nomor 4a
print("No 4a----------------------------------------------------------------------------------------------------------------------------------------")
import re
f=open('KEI.html','r', encoding='latin1')
teks4 = f.read()
f.close()
print(re.findall(r'([\w+]+)</a></td>',teks4))
print("\n")

#Nomor 4b
print("No 4b----------------------------------------------------------------------------------------------------------------------------------------")
f=open('KEI.html','r', encoding='latin1')
teks4 = f.read()
f.close()
```

```
#Nomor 4b
print("No 4b----------------------------------------------------------------------------")
f=open('KEI.html','r', encoding='latin1')
teks4 = f.read()
f.close()
string=re.findall(r'title="([\w+]+)">',teks4)
string=re.findall(r'">([\w+]+)</a></td>\n<td>([0-9.]+)</td>',teks4)
a=[]
for i in string:
    a.append((i[0], float(i[1])))
print(a)
```

Output :


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/ASUS/AppData/Local/Programs/Python/Python38-32/modul7.py ==
No 1----------
['merintahan', 'merdekaan', 'mei', 'merintahan', 'menjadi', 'menjalin', 'membawa', 'memonopoli', 'menyatakan', 'merdekaannya', 'mendapat', 'merauke', 'melanesia', 'mendiami', 'mencapai', 'membentuk', 'memiliki', 'memiliki', 'mendukung', 'merupakan', 'menjadi', 'merdekaan', 'merintahan', 'media', 'merujuk', 'merujuk', 'mengusulkan', 'melayu', 'mes', 'menggunakan', 'media', 'menggunakan', 'melayu', 'mengenai', 'menjadi', 'menggunakannya', 'memasyarakatkan', 'melalui', 'menggunakannya', 'mendirikan', 'merdekaan', 'menimbulkan', 'menyatu', 'menyebabkan', 'memusnahkan', 'mentara', 'membentuk', 'menandai', 'membentuk', 'melanesia', 'merupakan', 'membentuk', 'menyebabkan', 'melanesia', 'meskipun', 'melimpahnya', 'menyebabkan', 'melayu', 'melayu', 'melayu', 'menjadi', 'menjadi', 'menguasai', 'menanjung', 'mengendalikan', 'merupakan', 'merupakan', 'mendirikan', 'menjadi', 'menjadi', 'meliputi', 'menanjung', 'memasuki', 'memasuki', 'melalui', 'melalui', 'menyebarkan', 'mempengaruhi', 'menanjung', 'memperluas', 'menguasai', 'menanjung', 'menghadapi', 'menyebut', 'mengadopsi', 'meliputi', 'merupakan', 'menjadi', 'menjadi', 'memperluas', 'merintahannya', 'mencakup', 'merintahan', 'menjadi', 'mencakup', 'meliputi', 'memiliki', 'meliputi', 'mentara', 'meliputi', 'menggambarkan', 'merupakan', 'merupakan', 'membuat', 'menjajah', 'menguasai', 'menjajah', 'menaklukan', 'mendirikan', 'menjadi', 'meskipun', 'merintahan', 'mengalahkan', 'merintah', 'merintahan', 'mereka', 'menemukan', 'mereka', 'mendominasi', 'mendarat', 'menguasai', 'mengalahkan', 'mereka', 'menguasai', 'melalui', 'merintah', 'menghasilkan', 'merintahan', 'memperkenalkan', 'menguasai', 'mendapatkan', 'melihat', 'merupakan', 'mengerahkan', 'merdekaan', 'membentuk', 'merdekaan', 'memproklamasikan', 'merdekaan', 'merdekaan', 'menjabat', 'menteri', 'menguasai', 'mengirimkan', 'mereka', 'merdekaan', 'meredam', 'merdekaan', 'menerima', 'merdeka', 'mendapat', 'merika', 'menyerukan', 'membubarkan', 'menjadi', 'menteri', 'merintahan', 'mengikuti', 'merintis', 'menjadi', 'menjadi', 'meletus', 'menyebabkan', 'menengah', 'menyebut', 'menuduh', 'menggulingkan', 'merintahan', 'mengganti', 'menjadi', 'menggantikan', 'merintahan', 'menjadi', 'mengamankan', 'mentara', 'melemah', 'mentara', 'merintahan', 'menerapkan', 'mendatangkan', 'menghasilkan', 'meski', 'merata', 'men', 'menambah', 'melalui', 'meluas', 'memburuk', 'megawati', 'menangkan', 'menjabat', 'mengalami', 'melepaskan', 'memisahkan', 'menjadi', 'menewaskan', 'menghantam', 'meru', 'memiliki', 'menyebar', 'memberikan', 'menggunakan', 'meternya', 'mencapai', 'meter', 'memiliki', 'mega', 'menduduki', 'meskipun', 'mencatat', 'merusak', 'menurut', 'ment', 'memicu', 'menjadi', 'melindungi', 'merupakan', 'mengakibatkan', 'mencari', 'menambah', 'mereka', 'mencari', 'mengakibatkan', 'merintahan', 'menjalankan', 'merintahan', 'menjadi', 'meral', 'mendemen', 'mendemen', 'menjelma', 'menjadi', 'meral', 'merupakan', 'melalui', 'merupakan', 'melalui', 'menteri', 'mewakili', 'men', 'meskipun', 'menunjuk', 'menjaga', 'merintahan', 'mengingat', 'menteri', 'mendemen', 'meskipun', 'menteri', 'merika', 'merdeka', 'menghabiskan', 'menjaga', 'menjalin', 'melakukan', 'merintahan', 'menjadi', 'menjadi', 'menandatangani', 'menjadi', 'meskipun', 'menarik', 'mentah', 'menerima', 'merika', 'merintah', 'menimbulkan', 'menimbulkan', 'mestiknya', 'memiliki', 'mersial', 'men', 'menimbulkan', 'merdeka', 'mencolok', 'memiliki', 'mewa', 'menjadi', 'merintahan', 'memiliki', 'mentara', 'memiliki', 'memiliki', 'melalui', 'mewa', 'memiliki', 'mewa', 'membentuk', 'menetapkan', 'mendapatkan', 'mewa', 'mendukung', 'mendapat', 'menjadi', 'memisahkan', 'melalui', 'menjadi', 'medan', 'mewa', 'medan', 'menunjukkan', 'mestik', 'mencapai', 'menjadi', 'menjadi', 'merintahan', 'mengadaptasi', 'memadukannya', 'merintah', 'melut', 'mengakibatkan', 'merintahaan', 'menerapkan', 'menekan', 'menstabilkan', 'menarik', 'meningkat', 'menyebabkan', 'melonjaknya', 'memicu', 'menjelang', 'mengalirkan', 'mengalami', 'melanda', 'mei', 'melebihi', 'memengaruhi', 'mempunyai', 'mentah', 'meski', 'menjadi', 'mentah', 'mencapai', 'menyumbang', 'menyumbang', 'meskipun', 'mempekerjakan', 'mempekerjakan', 'merika', 'meski', 'menghadapi', 'merajalela', 'merintahan', 'menempatkan', 'ment', 'membaca', 'menurut', 'menurut', 'memiliki', 'merupakan', 'melanesia', 'menyatakan', 'menurut', 'mereka', 'melalui', 'menetap', 'menjadi', 'merintah', 'melakukan', 'menggolong', 'menjadikan', 'medan', 'mendagri', 'menjadi', 'memiliki', 'melalui', 'menterian', 'memiliki', 'menurut', 'melayu', 'merintah', 'meski', 'menghapus', 'menjadi', 'memiliki', 'meskipun', 'menjadi', 'merintah', 'mesti', 'mengalokasikan', 'mengalokasikan', 'menurut', 'mencapai', 'menempati', 'menurut', 'mencapai', 'menempati', 'melalui', 'ment', 'memang', 'meningkat', 'melaporkan', 'melaporkan', 'memperinci', 'menurut', 'menurut', 'mewa', 'mewa', 'mewa', 'memiliki', 'memiliki', 'melayu', 'memiliki', 'menampilkan', 'meuseukat', 'melayu', 'mengenakan', 'meliputi', 'merauke', 'merintahan', 'mencerminkan', 'membentuk', 'membawa', 'memberi', 'menjadi', 'melalui', 'menjadi', 'menampilkan', 'men', 'merah', 'medali', 'melan', 'mengesankan', 'menduduki', 'meraih', 'medali', 'melalui', 'merajai', 'merajai', 'menjadi', 'meraih', 'meraih', 'merupakan', 'merauke', 'memiliki', 'merakyat', 'melayu', 'melayu', 'melayu', 'melayu', 'melan', 'merupakan', 'memiliki', 'mematenkan', 'meliputi', 'menan', 'melan', 'memiliki', 'mendam', 'melayu', 'melati', 'mering', 'mele', 'men', 'mewa', 'meong', 'meong', 'menjadi', 'menyebabkan', 'memasak', 'melayu', 'menghasilkan', 'mendahului', 'membawa', 'merupakan', 'menggunakan', 'menyajikan', 'menyajikan', 'menyajikan', 'mes', 'mewa', 'mes', 'merdekaan', 'merupakan', 'memuncak', 'mendominasi', 'meskipun', 'meningkat', 'menjadi', 'mengkritik', 'merupakan', 'merdekaan', 'merupakan', 'memiliki', 'membantu', 'mendefinisikan', 'memelihara', 'mereka', 'media', 'meningkat', 'menyiarkan', 'mereka', 'menjadi', 'menurut', 'men_urusan_ekonomi_dan_sosial_pbb', 'ment', 'ment', 'ment', 'ment', 'merdeka', 'merican', 'menurut', 'mei', 'mes', 'merica', 'mes', 'mentary', 'ment', 'men
Ln: 63 Col: 4
3.17
IND 21.12


Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
nikasi', 'mesirtunisia', 'merika', 'merika', 'meuruguayvenezuela', 'merika', 'merika', 'meksikonikaraguapanama', 'meniaazerbaijangeorgiakazakhstankyrgystantajikistanturkmenistanuzbekistan', 'menterian', 'menterian', 'menialazerbaijanlbahrainbangladeshbhutanbruneifilipinageorgialindiaindonesiairakiranisraeljepangkambojakazakhstan3kirgizstankorea', 'mesir3mongoliamyanmarnepalomanpakistanqatarrusia3singapurasipruslsri', 'menistanuni', 'ment', 'measurestechnical', 'ment', 'measurestrade', 'ment', 'mentinformation', 'menteri', 'meniaaustraliabahrainbangladeshbarbadosbelizebeninboliviabotswanabrasilbruneiburkina', 'merunkanadatanjung', 'mesirel', 'meksikomoldovamongoliamontenegromarokomozambiknamibianepalselandia', 'meswazilandswisstaiwanâ³tanzaniathailandtiongkoktogotongatrinidad', 'merika', 'merupakan', 'men', 'mesir', 'merun', 'menistan', 'menia', 'meksiko', 'merun', 'mesir', 'menistan', 'men', 'meksiko', 'merika', 'meksiko', 'mesir', 'merika', 'meksiko', 'merintahan', 'messekretariat', 'mer', 'metersartikel']

No 2----------
['disambiguasi', 'disambiguasi', 'disambiguasi', 'dikendalikan', 'dikendalikan', 'din', 'dische', 'dia', 'dibubarkan', 'disebut', 'dilintasi', 'dia', 'diri', 'dipakai', 'dipilih', 'dia', 'dipengaruhi', 'dirinya', 'dia', 'diikuti', 'dia', 'diri', 'diri', 'diami', 'dinyatakan', 'didikan', 'dia', 'dia', 'dia', 'dia', 'dia', 'dia', 'dia', 'dia', 'diã', 'dia', 'diã', 'diperkenalkan', 'ditulis', 'die', 'dirikan', 'diri', 'diri', 'dijuluki', 'diperdebatkan', 'disebut', 'dikatakan', 'diri', 'dian', 'disi', 'ditambah', 'dimiliki', 'diketahui', 'dis', 'dis', 'dikit', 'diketahui', 'diri', 'diri', 'diri', 'dian', 'diri', 'diri', 'diikuti', 'dirinya', 'diri', 'diantaranya', 'diri', 'dian', 'diri', 'dia', 'dipengaruhi', 'dinasti', 'dirikan', 'dian', 'diri', 'diri', 'dia', 'dibawah', 'disebut', 'diketahui', 'diri', 'diri', 'diri', 'dian', 'diri', 'diri', 'diri', 'diperdebatkan', 'diri', 'dikatakan', 'diri', 'dikuasai', 'diri', 'diri', 'diri', 'diantaranya', 'dibawah', 'dikuasai', 'dian', 'dikuasai', 'dia', 'dia', 'dijajah', 'diusir', 'diusir', 'dirikan', 'dia', 'dia', 'dia', 'dileburkan', 'dia', 'dipegang', 'dikenal', 'ditandatanganinya', 'dilakukanlah', 'diusir', 'dikenal', 'dian', 'dilaksanakan', 'dihasilkan', 'dihapus', 'dia', 'dijajah', 'dia', 'diperlukan', 'diberikan', 'diri', 'dian', 'dikenal', 'dikenal', 'disebut', 'dian', 'dinya', 'dian', 'dirinya', 'dian', 'dijadikan', 'diri', 'disi', 'diri', 'dicurigai', 'dibunuh', 'dicabut', 'dinamakan', 'disebut', 'disusun', 'dipanggil', 'dia', 'dipaksa', 'disi', 'din', 'diselenggarakan', 'dimenangkan', 'diri', 'diri', 'dilanda', 'dia', 'dian', 'disusul', 'disekitar', 'dinat', 'dia', 'diri', 'diukur', 'diri', 'dii', 'diameternya', 'disebut', 'diversity', 'dikenal', 'dinya', 'didasarkan', 'dikatif', 'dipegang', 'diri', 'ditambah', 'dipilih', 'dilantik', 'ditambah', 'diketuai', 'diketuai', 'diketuai', 'diusung', 'diisi', 'dianggap', 'dikatif', 'dijalankan', 'disial', 'dipertahankan', 'didasarkan', 'diri', 'diri', 'diri', 'dilakukan', 'dikeluarkan', 'dipengaruhi', 'diri', 'dilindungi', 'dilakukan', 'dia', 'diterapkannya', 'diri', 'dibagi', 'diakomodasi', 'dian', 'dipilih', 'dibandingkan', 'diri', 'disebut', 'digabungkan', 'dian', 'diri', 'diy', 'didukung', 'diluncurkannya', 'ditambah', 'dinya', 'disiplin', 'disertai', 'diri', 'diperkirakan', 'disebabkan', 'dikeluarkannya', 'dirinya', 'dibagi', 'dia', 'dipeluk', 'dikan', 'diajarkan', 'dikuasai', 'diselenggarakan', 'didikan', 'dikit', 'digunakan', 'diperkenalkan', 'didikan', 'dipraktikkan', 'didikan', 'didikan', 'didikan', 'didikan', 'didik', 'dinasan', 'disediakan', 'dibandingkan', 'didikan', 'dilapaorkan', 'dianggap', 'dilaporkan', 'dilaporkan', 'dipengaruhi', 'dia', 'diri', 'disional', 'dian', 'ditemukan', 'dipergunakan', 'dis', 'disional', 'diklaim', 'dikenali', 'dikenakan', 'disionalnya', 'dia', 'dilihat', 'disional', 'diadakan', 'diri', 'diri', 'ditampilkan', 'diikuti', 'disional', 'disional', 'diraih', 'diraih', 'dikusuma', 'dianggap', 'diraihnya', 'dia', 'disional', 'disional', 'diri', 'disional', 'dikenal', 'dikenal', 'dipengaruhi', 'dia', 'disional', 'disional', 'disional', 'diklaim', 'diri', 'disional', 'dia', 'dirondang', 'dindin', 'ding', 'dirut', 'diunggak', 'dis', 'dis', 'dindang', 'ding', 'ding', 'dipanah', 'din', 'dindi', 'dipadhi', 'dindang', 'dindang', 'dingkoh', 'dis', 'dianku', 'diapede', 'dilale', 'diru', 'diru', 'dihidangkan', 'diri', 'dia', 'disional', 'ditemukan', 'dimakan', 'disajikan', 'dijual', 'disajikan', 'ditemui', 'dijual', 'dikenal', 'dijual', 'dijual', 'disional', 'diproduksi', 'dibuat', 'diabelanda', 'dibuat', 'diproduksi', 'dian', 'diakui', 'disutradarai', 'dihasilkannya', 'dirilis', 'diangkat', 'disi', 'dia', 'dio', 'dilaporkan', 'diarsipkan', 'diakses', 'dition', 'diakses', 'diarsipkan', 'diakses', 'diakses', 'diarsipkan', 'diakses', 'division', 'diakses', 'distribusi', 'diakses', 'diabaikan', 'dices', 'dicators', 'diarsipkan', 'diakses', 'diarsipkan', 'diakses', 'diverifikasi', 'dianut', 'diakses', 'dinata', 'dies', 'ditions', 'diakses', 'ding', 'dian', 'dian', 'dian', 'ding', 'dian', 'ditions', 'dings', 'ditions', 'ditions', 'din', 'diakses', 'diakses', 'diakses', 'directory', 'diakses', 'dies', 'diversity', 'diversity', 'diakses', 'dimuat', 'diakses', 'diakses', 'diakses', 'diakses', 'dilemmas', 'diakses', 'diakses', 'ditional', 'ding', 'diakses', 'diakses', 'diakses', 'diakses', 'director', 'diarsipkan', 'diarsipkan', 'diakses', 'diakses', 'diakses', 'ditorial', 'diakses', 'diakses', 'diakses', 'diakses', 'dikator', 'diakses', 'diakses', 'dia', 'dicators', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'didikan', 'didikan', 'diakses', 'diakses', 'diakses', 'diakses', 'dia', 'diakses', 'diakses', 'ding', 'diture', 'diakses', 'diakses', 'diakses', 'diakses', 'dia', 'diakses', 'diakses', 'die', 'dishes', 'diakses', 'diakses', 'diakses', 'disi', 'division', 'disi', 'dia', 'dia', 'dia', 'dies', 'dia', 'dia', 'dia', 'dia', 'dia', 'didikanolahragabusana', 'diatelekomunikasi', 'diotanda', 'diamaladewanepalpakistansri', 'dibahrainirakiranisraelkuwaitlebanonomanpalestinaqatarsuriahturkiueayamanyordania', 'diahongariairlandiaitaliajermankroasialatvialituanialuksemburgmaltaprancispolandiaportugalrumaniasiprusslowakiasloveniaspa
Ln: 63 Col: 4
3.17
IND 21.12

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
anku', 'diapede', 'dilale', 'diru', 'diru', 'dihidangkan', 'diri', 'dia', 'disional', 'ditemukan', 'dimakan', 'disajikan', 'dijual', 'disajikan', 'ditemui', 'dijual',
'dikenal', 'dijual', 'dijual', 'disional', 'diproduksi', 'dibuat', 'diabelanda', 'dibuat', 'diproduksi', 'dian', 'diakui', 'disutradarai', 'dihasilkannya', 'dirilis',
'diangkat', 'disi', 'dia', 'dio', 'dilaporkan', 'diarsipkan', 'diakses', 'dition', 'diakses', 'diarsipkan', 'diakses', 'diakses', 'diarsipkan', 'diakses', 'division',
'diakses', 'distribusi', 'diakses', 'diabaikan', 'dices', 'dicators', 'diarsipkan', 'diakses', 'diarsipkan', 'diakses', 'diverifikasi', 'dianut', 'diakses', 'dinata',
'dies', 'ditions', 'diakses', 'ding', 'dian', 'dian', 'dian', 'ding', 'dian', 'ditions', 'dings', 'ditions', 'ditions', 'din', 'diakses', 'diakses', 'diakses', 'direct
ory', 'diakses', 'dies', 'diversity', 'diversity', 'diakses', 'dimuat', 'diakses', 'diakses', 'diakses', 'dilemmas', 'diakses', 'diakses', 'ditional', 'ding
', 'diakses', 'diakses', 'diakses', 'diakses', 'director', 'diarsipkan', 'diarsipkan', 'diakses', 'diakses', 'diakses', 'ditorial', 'diakses', 'diakses', 'diakses', 'd
iakses', 'diakses', 'dikator', 'diakses', 'diakses', 'dia', 'dicators', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'diakses', 'diakse
s', 'diakses', 'didikan', 'didikan', 'diakses', 'diakses', 'diakses', 'diakses', 'dia', 'diakses', 'diakses', 'ding', 'diture', 'diakses', 'diakses', 'diakses', 'diaks
es', 'dia', 'diakses', 'diakses', 'diakses', 'die', 'dishes', 'diakses', 'diakses', 'diakses', 'diakses', 'disi', 'division', 'disi', 'dia', 'dia', 'dia', 'dia', 'dies
', 'dia', 'dia', 'dia', 'dia', 'dia', 'didikanolahragabusana', 'diatelekomunikasi', 'diotanda', 'diamaladewanepalpakistansri', 'dibahrainirakiranisraelkuwaitlebanonoma
npalestinaqatarsuriahturkiueayamanyordania', 'diahongariairlandiaitaliajermankroasialatvialituanialuksemburgmaltaprancispolandiaportugalrumaniasiprusslowakiasloveniasp
anyolswediayunani', 'diamoldovamontenegronorwegiarusiaserbiaswissukrainavatikan', 'dia', 'diplomatik', 'diaspora', 'diarmenialazerbaijanlbahrainbangladeshbhutanbruneif
ilipinageorgialindiaindonesiairakiranisraeljepangkambojakazakhstan3kirgizstankorea', 'dia', 'dimasukkan', 'dimasukkan', 'dipertentangkan', 'dinglembaga', 'direktur', '
direktur', 'direktur', 'direktur', 'dikambojakamerunkanadatanjung', 'diaindiaindonesiaisraeljamaikajepangyordaniakenyakorea', 'dia', 'dinessamoasaudi', 'diri', 'diapra
ncisjermanyunanihongariairlandiaitalialatvialituanialuksemburgmaltabelandapolandiaportugalrumaniaslovakiasloveniaspanyolswediabritania', 'dia', 'ding', 'dines', 'ding'
, 'dia', 'dia', 'dia', 'disertai', 'dia']


No 3----------------------------------------------------------------------------------------------------------------------------------------------------------------
['di Asia', 'di antara', 'di dunia', 'di dunia', 'di dunia', 'di Pulau', 'di Pulau', 'di Pulau', 'di India', 'di wilayah', 'di Palembang', 'di bawah', 'di akhir', 'di
bencana', 'di ujung', 'di tanah', 'di mana', 'di dunia', 'di anggota', 'di India', 'di wilayah', 'di media', 'di lebih', 'di luar', 'di Suryaningrat', 'di Belanda', 'd
i erupsi', 'di dataran', 'di Nusantara', 'di Nusantara', 'di sana', 'di timur', 'di pulau', 'di daerah', 'di Sumatra', 'di Nusantara', 'di daerah', 'di kerajaan', 'di
Kalimantan', 'di Kalimantan', 'di Sulawesi', 'di Sulawesi', 'di daerah', 'di Sumatra', 'di kemaharajaan', 'di Asia', 'di Selat', 'di dunia', 'di pulau', 'di kerajaan',
'di kemaharajaan', 'di Nusantara', 'di Indonesia', 'di pada', 'di Indonesia', 'di Aceh', 'di daerah', 'di sebelah', 'di Aceh', 'di Sumatra', 'di Sumatra', 'di pulau',
'di Jawa', 'di Jawa', 'di daerah', 'di pulau', 'di pesisir', 'di pesisir', 'di bawahan', 'di bawahan', 'di bawah', 'di beberapa', 'di Sulawesi', 'di Maluku', 'di Nusa'
, 'di Nusantara', 'di jajahan', 'di pulau', 'di Indonesia', 'di Indonesia', 'di dua', 'di antara', 'di Jawa', 'di Hindia', 'di bawah', 'di presiden', 'di lebih', 'di s
aksi', 'di belakang', 'di berdasarkan', 'di bawah', 'di Pejabat', 'di luar', 'di dunia', 'di dalam', 'di bawah', 'di negara', 'di Yogyakarta', 'di Sidoarjo', 'di Indon
esia', 'di Taman', 'di Asia', 'di antaranya', 'di antara', 'di antara', 'di mana', 'di dunia', 'di dunia', 'di Indonesia', 'di dunia', 'di di', 'di daerah', 'di sektor
', 'di kawasan', 'di musim', 'di Indonesia', 'di negara', 'di Indonesia', 'di lembaga', 'di lembaga', 'di Indonesia', 'di parlemen', 'di kabinetnya', 'di Indonesia', '
di Istana', 'di Indonesia', 'di Jakarta', 'di Asia', 'di awal', 'di anggota', 'di Organisasi', 'di anggota', 'di Kuta', 'di atas', 'di sebagian', 'di pelanggaran', 'di
pada', 'di kemajuan', 'di kekurangan', 'di antaranya', 'di 416', 'di indonesia', 'di Jakarta', 'di Jakarta', 'di provinsi', 'di Negara', 'di Indonesia', 'di Indonesia'
, 'di mata', 'di Rupiah', 'di bawah', 'di luar', 'di dunia', 'di pengimpor', 'di Pulau', 'di mana', 'di Indonesia', 'di antaranya', 'di bagian', 'di Indonesia', 'di du
nia', 'di Indonesia', 'di Indonesia', 'di seluruh', 'di negara', 'di Indonesia', 'di pusat', 'di Indonesia', 'di bawah', 'di Indonesia', 'di Papua', 'di pulau', 'di In
donesia', 'di tingkat', 'di kaum', 'di Indonesia', 'di bahasa', 'di Indonesia', 'di Indonesia', 'di luar', 'di tingkat', 'di tingkat', 'di bawah', 'di tingkat', 'di In
donesia', 'di antaranya', 'di daerah', 'di seluruh', 'di setiap', 'di Kalimantan', 'di Prambanan', 'di cukup', 'di Jakarta', 'di miniatur', 'di Institut', 'di Indonesi
a', 'di Indonesia', 'di Flores', 'di Sumba', 'di Indonesia', 'di yang', 'di Indonesia', 'di mana', 'di Hartono', 'di Hartono', 'di juara', 'di Piala', 'di Prancis', 'd
i Indonesia', 'di Indonesia', 'di daerah', 'di Indonesia', 'di Indonesia', 'di masing', 'di Mato', 'di Laut', 'di Laun', 'di Sayang', 'di tempat', 'di daerah', 'di war
ung', 'di berbagai', 'di Indonesia', 'di warung', 'di tempat', 'di atas', 'di pasar', 'di nusantara', 'di Bandung', 'di teater', 'di Indonesia', 'di film', 'di Indones
ia', 'di Indonesia', 'di Indonesia', 'di 83', 'di dunia', 'di Indonesia', 'di Nusantara', 'di Utang', 'di Indonesia', 'di Koran', 'di A', 'di DaerahPDF', 'di parameter
', 'di Piala', 'di parameter', 'di Wikimedia', 'di OpenStreetMap', 'di EncyclopÃ', 'di Timor', 'di luar', 'di IndonesiaDiaspora', 'di Asia', 'di Asia', 'di ArabiaSeneg
alSierra', 'di Â', 'di Â', 'di Â', 'di Â', 'di Wikidata', 'di Asia']
```

Ln: 63 Col: 4

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
ncisjermanyunanihongariairlandiaitalialatvialituanialuksemburgmaltabelandapolandiaportugalrumaniaslovakiasloveniaspanyolswediabritania', 'dia', 'ding', 'dines', 'ding'
, 'dia', 'dia', 'dia', 'disertai', 'dia']


No 3----------------------------------------------------------------------------------------------------------------------------------------------------------------
['di Asia', 'di antara', 'di dunia', 'di dunia', 'di dunia', 'di Pulau', 'di Pulau', 'di Pulau', 'di India', 'di wilayah', 'di Palembang', 'di bawah', 'di akhir', 'di
bencana', 'di ujung', 'di tanah', 'di mana', 'di dunia', 'di anggota', 'di India', 'di wilayah', 'di media', 'di lebih', 'di luar', 'di Suryaningrat', 'di Belanda', 'd
i erupsi', 'di dataran', 'di Nusantara', 'di Nusantara', 'di sana', 'di timur', 'di pulau', 'di daerah', 'di Sumatra', 'di Nusantara', 'di daerah', 'di kerajaan', 'di
Kalimantan', 'di Kalimantan', 'di Sulawesi', 'di Sulawesi', 'di daerah', 'di Sumatra', 'di kemaharajaan', 'di Asia', 'di Selat', 'di dunia', 'di pulau', 'di kerajaan',
'di kemaharajaan', 'di Nusantara', 'di Indonesia', 'di pada', 'di Indonesia', 'di Aceh', 'di daerah', 'di sebelah', 'di Aceh', 'di Sumatra', 'di Sumatra', 'di pulau',
'di Jawa', 'di Jawa', 'di daerah', 'di pulau', 'di pesisir', 'di pesisir', 'di bawahan', 'di bawahan', 'di bawah', 'di beberapa', 'di Sulawesi', 'di Maluku', 'di Nusa'
, 'di Nusantara', 'di jajahan', 'di pulau', 'di Indonesia', 'di Indonesia', 'di dua', 'di antara', 'di Jawa', 'di Hindia', 'di bawah', 'di presiden', 'di lebih', 'di s
aksi', 'di belakang', 'di berdasarkan', 'di bawah', 'di Pejabat', 'di luar', 'di dunia', 'di dalam', 'di bawah', 'di negara', 'di Yogyakarta', 'di Sidoarjo', 'di Indon
esia', 'di Taman', 'di Asia', 'di antaranya', 'di antara', 'di antara', 'di mana', 'di dunia', 'di dunia', 'di Indonesia', 'di dunia', 'di di', 'di daerah', 'di sektor
', 'di kawasan', 'di musim', 'di Indonesia', 'di negara', 'di Indonesia', 'di lembaga', 'di lembaga', 'di Indonesia', 'di parlemen', 'di kabinetnya', 'di Indonesia', '
di Istana', 'di Indonesia', 'di Jakarta', 'di Asia', 'di awal', 'di anggota', 'di Organisasi', 'di anggota', 'di Kuta', 'di atas', 'di sebagian', 'di pelanggaran', 'di
pada', 'di kemajuan', 'di kekurangan', 'di antaranya', 'di 416', 'di indonesia', 'di Jakarta', 'di Jakarta', 'di provinsi', 'di Negara', 'di Indonesia', 'di Indonesia'
, 'di mata', 'di Rupiah', 'di bawah', 'di luar', 'di dunia', 'di pengimpor', 'di Pulau', 'di mana', 'di Indonesia', 'di antaranya', 'di bagian', 'di Indonesia', 'di du
nia', 'di Indonesia', 'di Indonesia', 'di seluruh', 'di negara', 'di Indonesia', 'di pusat', 'di Indonesia', 'di bawah', 'di Indonesia', 'di Papua', 'di pulau', 'di In
donesia', 'di tingkat', 'di kaum', 'di Indonesia', 'di bahasa', 'di Indonesia', 'di Indonesia', 'di luar', 'di tingkat', 'di tingkat', 'di bawah', 'di tingkat', 'di In
donesia', 'di antaranya', 'di daerah', 'di seluruh', 'di setiap', 'di Kalimantan', 'di Prambanan', 'di cukup', 'di Jakarta', 'di miniatur', 'di Institut', 'di Indonesi
a', 'di Indonesia', 'di Flores', 'di Sumba', 'di Indonesia', 'di yang', 'di Indonesia', 'di mana', 'di Hartono', 'di Hartono', 'di juara', 'di Piala', 'di Prancis', 'd
i Indonesia', 'di Indonesia', 'di daerah', 'di Indonesia', 'di Indonesia', 'di masing', 'di Mato', 'di Laut', 'di Laun', 'di Sayang', 'di tempat', 'di daerah', 'di war
ung', 'di berbagai', 'di Indonesia', 'di warung', 'di tempat', 'di atas', 'di pasar', 'di nusantara', 'di Bandung', 'di teater', 'di Indonesia', 'di film', 'di Indones
ia', 'di Indonesia', 'di Indonesia', 'di 83', 'di dunia', 'di Indonesia', 'di Nusantara', 'di Utang', 'di Indonesia', 'di Koran', 'di A', 'di DaerahPDF', 'di parameter
', 'di Piala', 'di parameter', 'di Wikimedia', 'di OpenStreetMap', 'di EncyclopÃ', 'di Timor', 'di luar', 'di IndonesiaDiaspora', 'di Asia', 'di Asia', 'di ArabiaSeneg
alSierra', 'di Â', 'di Â', 'di Â', 'di Â', 'di Wikidata', 'di Asia']


No 4a---------------------------------------------------------------------------------------------------------------------------------------------------------------
[]


No 4b---------------------------------------------------------------------------------------------------------------------------------------------------------------
[]
>>>
== RESTART: C:/Users/ASUS/AppData/Local/Programs/Python/Python38-32/modul7.py ==
No 1----------------------------------------------------------------------------------------------------------------------------------------------------------------
['merintahan', 'merdekaan', 'mei', 'merintahan', 'menjadi', 'menjalin', 'membawa', 'memonopoli', 'menyatakan', 'merdekaannya', 'mendapat', 'merauke', 'melanesia', 'men
diami', 'mencapai', 'membentuk', 'memiliki', 'memiliki', 'mendukung', 'merupakan', 'menjadi', 'merdekaan', 'merintahan', 'media', 'merujuk', 'merujuk', 'mengusulkan',
'melayu', 'mes', 'menggunakan', 'media', 'menggunakan', 'melayu', 'mengenai', 'menjadi', 'menggunakannya', 'memasyarakatkan', 'melalui', 'menggunakannya', 'mendirikan'
, 'merdekaan', 'menimbulkan', 'menyatu', 'menyebabkan', 'memusnahkan', 'mentara', 'membentuk', 'menandai', 'membentuk', 'melanesia', 'merupakan', 'membentuk', 'menyeba
```

Ln: 63 Col: 4

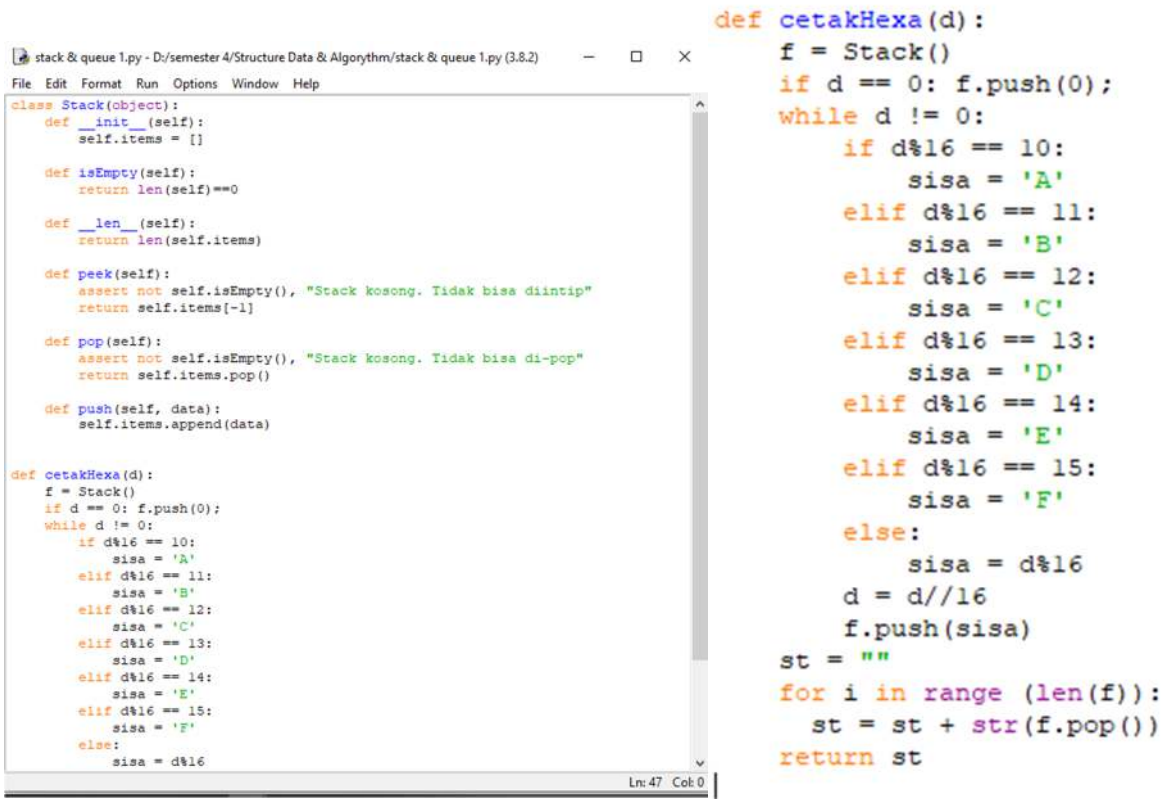
Nama : Muhammad Khalif Rizaldi W

Kelas : H

NIM : L200180217

Praktikum Algoritma dan Struktur Data

1.



```
stack & queue 1.py - D:/semester 4/Structure Data & Algorythm/stack & queue 1.py (3.8.2)
File  Edit  Format  Run  Options  Window  Help
class Stack(object):
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return len(self)==0

    def __len__(self):
        return len(self.items)

    def peek(self):
        assert not self.isEmpty(), "Stack kosong. Tidak bisa diintip"
        return self.items[-1]

    def pop(self):
        assert not self.isEmpty(), "Stack kosong. Tidak bisa di-pop"
        return self.items.pop()

    def push(self, data):
        self.items.append(data)


def cetakHexa(d):
    f = Stack()
    if d == 0: f.push(0);
    while d != 0:
        if d%16 == 10:
            sisa = 'A'
        elif d%16 == 11:
            sisa = 'B'
        elif d%16 == 12:
            sisa = 'C'
        elif d%16 == 13:
            sisa = 'D'
        elif d%16 == 14:
            sisa = 'E'
        elif d%16 == 15:
            sisa = 'F'
        else:
            sisa = d%16
Ln: 47  Col: 0
```

```
def cetakHexa(d):
    f = Stack()
    if d == 0: f.push(0);
    while d != 0:
        if d%16 == 10:
            sisa = 'A'
        elif d%16 == 11:
            sisa = 'B'
        elif d%16 == 12:
            sisa = 'C'
        elif d%16 == 13:
            sisa = 'D'
        elif d%16 == 14:
            sisa = 'E'
        elif d%16 == 15:
            sisa = 'F'
        else:
            sisa = d%16
        d = d//16
        f.push(sisa)
    st = ""
    for i in range (len(f)):
      st = st + str(f.pop())
    return st
```

```
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 (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/semester 4/Structure Data & Algorythm/stack & queue 1.py =====
>>> cetakHexa(12)
'C'
>>> cetakHexa(31)
'1F'
>>> cetakHexa(229)
'E5'
>>> cetakHexa(255)
'FF'
>>> cetakHexa(31519)
'7B1F'
>>>
```

2.

stack & queue 2.py - D:/semester 4/Structure Data & Algorythm/stack & queue 2.py (3.8.2)

File Edit Format Run Options Window Help

```
class Stack(object):
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return len(self)==0

    def __len__(self):
        return len(self.items)

    def peek(self):
        assert not self.isEmpty(), "Stack kosong. Tidak bisa diintip"
        return self.items[-1]

    def pop(self):
        assert not self.isEmpty(), "Stack kosong. Tidak bisa di-pop"
        return self.items.pop()

    def push(self, data):
        self.items.append(data)


nilai = Stack()
for i in range(16):
    if i % 3 == 0:
        nilai.push(i)
```

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 (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/semester 4/Structure Data & Algorythm/stack & queue 2.py =====
>>> print(nilai.items)
[0, 3, 6, 9, 12, 15]
>>>
```

3.

stack & queue 3.py - D:/semester 4/Structure Data & Algorythm/stack & queue 3.py (3.8.2)

File Edit Format Run Options Window Help

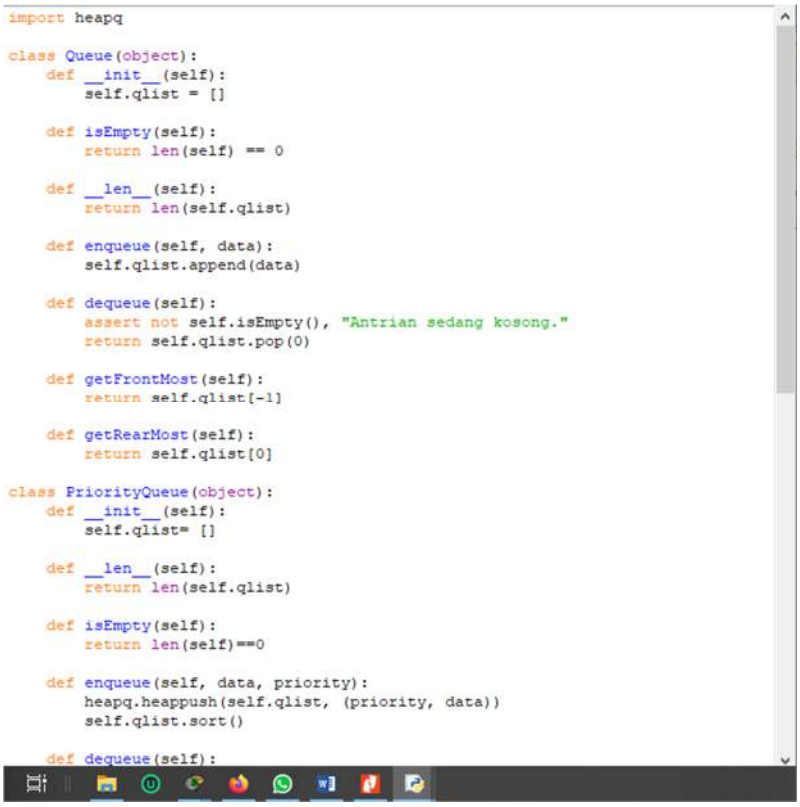
```
class Stack(object):
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return len(self)==0

    def __len__(self):
        return len(self.items)

    def peek(self):
        assert not self.isEmpty(), "Stack kosong. Tidak bisa diintip"
        return self.items[-1]

    def pop(self):
        assert not self.isEmpty(), "Stack kosong. Tidak bisa di-pop"
        return self.items.pop()

    def push(self, data):
        self.items.append(data)

nilai = Stack()
for i in range(16):
    if i % 3 == 0:
        nilai.push(i)
    elif i % 4 == 0:
        nilai.pop()
```

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) [M
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
===== RESTART: D:/semester 4/Structure Data & Algorythm/stac
>>> print(nilai.items)
[0, 9, 12, 15]
>>>
```

4.



```
import heapq

class Queue(object):
    def __init__(self):
        self.qlist = []

    def isEmpty(self):
        return len(self) == 0

    def __len__(self):
        return len(self.qlist)

    def enqueue(self, data):
        self.qlist.append(data)

    def dequeue(self):
        assert not self.isEmpty(), "Antrian sedang kosong."
        return self.qlist.pop(0)

    def getFrontMost(self):
        return self.qlist[-1]

    def getRearMost(self):
        return self.qlist[0]

class PriorityQueue(object):
    def __init__(self):
        self.qlist= []

    def __len__(self):
        return len(self.qlist)

    def isEmpty(self):
        return len(self)==0

    def enqueue(self, data, priority):
        heapq.heappush(self.qlist, (priority, data))
        self.qlist.sort()

    def dequeue(self):
```

```
class _PriorityQEntry(object):
    def __init__(self, data, priority):
        self.item = data
        self.priority = priority

Q = Queue()
Q.enqueue('qqqqq')
Q.enqueue('wwwww')
Q.enqueue('eeeee')

print("===QUEUE===")
print("enqueue: ",Q.qlist)
Q.dequeue()
print("dequeue", Q.qlist)

print("item paling depan:", Q.getFrontMost())
print("item peling belakang:", Q.getRearMost())


P = PriorityQueue()

P.enqueue('qqqqq', 5)
P.enqueue('wwwww', 1)
P.enqueue('eeeee', 2)

print("\n===PRIORITY QUEUE===")
print("enqueue: ", P.qlist)
P.dequeue()
print("dequeue: ", P.qlist)

print("item paling depan:", Q.getFrontMost())
print("item peling belakang:", Q.getRearMost())
```

5.

stack & queue 5.py - D:/semester 4/Structure Data & Algorythm/stack & queue 5.py (3.8.2)

File Edit Format Run Options Window Help

```
import heapq

class PriorityQueue(object):
    def __init__(self):
        self.qlist= []

    def __len__(self):
        return len(self.qlist)

    def isEmpty(self):
        return len(self)==0

    def enqueue(self, data, priority):
        heapq.heappush(self.qlist, (priority, data))
        self.qlist.sort()

    def dequeue(self):
        return self.qlist.pop(-1)

a = PriorityQueue()

a.enqueue('qqqqq', 5)
a.enqueue('wwwww', 1)
a.enqueue('eeeee', 2)

print(a.qlist)
a.dequeue()
print(a.qlist)
```

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 (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/semester 4/Structure Data & Algorythm/stack & queue 5.py =====
[(1, 'wwwww'), (2, 'eeeee'), (5, 'qqqqq')]
[(1, 'wwwww'), (2, 'eeeee')]
>>>
```

Nama : Muh Khalif Rizaldi W

Kelas : H

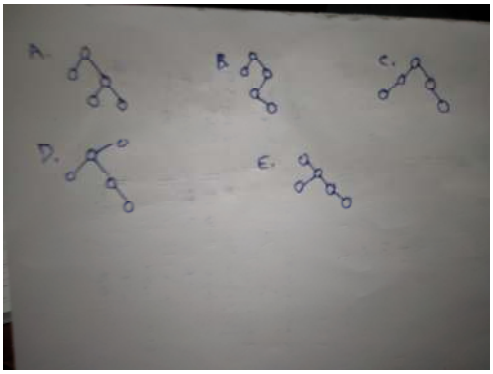
NIM : L200180217

Praktikum Algoritma dan Struktur Data

MODUL 9 BINER

1. Di berikan pohon biner dengan ukuran *n*, berapakah jumlah level minimum yang bisa di muatnya ? Berapakah jumlah level maksimumnya ? Tentukan nilai *n* berikut.
   a. n = 10, tinggi max = 10, tinggi min = $[\log_2 10] + 1 = 4$
   b. n = 35, tinggi max = 35, tinggi min = $[\log_2 35] + 1 = 6$
   c. n = 76, tinggi max = 76, tinggi min = $[\log_2 76] + 1 = 7$
   d. n = 345, tinggi max = 345, tinggi min = $[\log_2 345] + 1 = 9$

2. Gambarlah semua bentuk pohon biner berukuran 5 yang mungkin. Ada berapa kemungkinan ? Ada sekitar 5 kemungkinan dengan modifikasi lebih banyak namun konsepsi mendekati 5 kemungkinan diatas.



3. Berapakah jumlah simpul maksimum suatu pohon biner dengan jumlah level *h* ?
   Tentukan untuk nilai *h* berikut :
   a. h = 3, jumlah simpul max = $2^3 = 8$
   b. h = 4, jumlah simpul max = $2^4 = 16$
   c. h = 5, jumlah simpul max = $2^5 = 32$
   d. h = 6, jumlah simpul max = $2^6 = 64$

4. Di berikan pohon seperti berikut !
   a. Tunjukkan semua properti struktural yang berlaku pada tiap-tiap pohon di atas *penuh,sempurna dan komplit.* Ingat bahwa sebuah pohon biner bisa saja bersifat penuh sekaligus sempurna dan sebaliknya.
      A. Struktur Pohon : penuh dan komplit
      B. Struktur Pohon : sempurna, penuh dan komplit
      C. Struktur Pohon : penuh dan komplit

D. Struktur Pohon : -
E. Struktur Pohon : komplit

b. Tentukan ukuran pohon.
   A. Ukuran Pohon : 7
   B. Ukuran Pohon : 15
   C. Ukuran Pohon : 14
   D. Ukuran Pohon : 7
   E. Ukuran Pohon : 11
c. Tentukan ketinggian tiap pohon.
   A. Ketinggian Pohon : 4
   B. Ketinggian Pohon : 4
   C. Ketinggian Pohon : 8
   D. Ketinggian Pohon : 4
   E. Ketinggian Pohon : 4

d. Tentukan lebar pohon.
   A. Lebar Pohon : 2
   B. Lebar Pohon : 8
   C. Lebar Pohon : 2
   D. Lebar Pohon : 3
   E. Lebar Pohon : 5

5. Perhatikan pohon biner berikut !
   a. Tunjukkan urutan pengujungan simpul untuk :
      I. Prevorder traversal = 14, 78, 39, 52, 41, 83, 17, 9, 2, 60, 23, 4, 19
      II. Inorder traversal = 17, 39, 78, 83, 9, 52, 14, 41, 60, 2, 4, 23, 19
      III. Postorder traversal = 39, 17, 83, 78, 52, 9, 41, 14, 60, 4, 2, 23, 19
   b. Simpul man yang termasuk simpul daun ? 39, 41, 17, 9, 60, 4, 19
   c. Simpul mana saja yang merupakan simpul dalam ? 14, 78, 52, 83, 2, 23
   d. Simpul mana saja yang berada di level 4 ? 17, 9
   e. Tulis semua simpul yang berada di dalam jalur dari simpul akar menuju simpul :
      I. 83 = 14, 78, 52, 83
      II. 39 = 14, 78, 39
      III. 4 = 14, 2, 23, 4
      IV. 9 = 14, 78, 52, 83, 9
   f. Perhatikan simpul 52, tentukan :
      I. Keturunannya = 78
      II. Leluhurnya = 78, 14
      III. Saudaranya = 39
   g. Tentukan kedalaman dari tiap-tiap simpul ini :
      I. 78, memiliki kedalaman 1
      II. 41, memiliki kedalaman 3
      III. 60, memiliki kedalaman 2

IV. 19, memiliki kedalaman 3

6. Buatlah fungsi ukuran pohon(akar) yang akan mendapatkan ukuran sebuah pohin biner.

```
class simpulBiner(object):
    def __init__(self, data):
        self.data = data
        self.kiri = None
        self.kanan = None

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

A = simpulBiner('Ambarawa')
B = simpulBiner('Bantul')
C = simpulBiner('Cimahi')
D = simpulBiner('Denpasar')
E = simpulBiner('Enrekang')
F = simpulBiner('Flores')
G = simpulBiner('Garut')
H = simpulBiner('Halmahera Timur')
I = simpulBiner('Indramayu')
J = simpulBiner('Jakarta')

A.kiri = B; A.kanan = C
B.kiri = D; B.kanan = E
C.kiri = F; C.kanan = G
E.kiri = H
G.kanan = I

datalist = [A.data, B.data, C.data, D.data, E.data, F.data, G.data, H.data, I.data, J.data]
level = []

def ukuranPohon(akar):
    if akar is None:
        return 0
    else:
        return (ukuranPohon(akar.kiri)+ 1 + ukuranPohon(akar.kanan))

print('Ukuran dari Binary Tree adalah', ukuranPohon(A))
```

```
93_Modul9_G\No 6.py
Ukuran dari Binary Tree adalah 9
>>>
```

7. Buatlah sebuah fungsi tinggiPohon(akar) yang akan mendapatkan ketinggiam sebuah pohon biner.

```
class simpulBiner(object):
    def __init__(self, data):
        self.data = data
        self.kiri = None
        self.kanan = None

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

A = simpulBiner('Ambarawa')
B = simpulBiner('Bantul')
C = simpulBiner('Cimahi')
D = simpulBiner('Denpasar')
E = simpulBiner('Enrekang')
F = simpulBiner('Flores')
G = simpulBiner('Garut')
H = simpulBiner('Halmahera Timur')
I = simpulBiner('Indramayu')
J = simpulBiner('Jakarta')

A.kiri = B; A.kanan = C
B.kiri = D; B.kanan = E
C.kiri = F; C.kanan = G
E.kiri = H
G.kanan = I

datalist = [A.data, B.data, C.data, D.data, E.data, F.data, G.data, H.data, I.data, J.data]
level = []

def tinggiPohon(akar):
    if akar is None:
        return 0 ;

    else :
        lDepth = tinggiPohon(akar.kiri)
        rDepth = tinggiPohon(akar.kanan)

        if (lDepth > rDepth):
            return lDepth+1
        else:
            return rDepth+1
```

```
Tinggi maksimal dari Binary Tree adalah 4
>>>
```

8. Buatlah sebuah fungsi yang mencetak data tiap simpul sekaligus level di mana simpul itu berbeda. Silahkan memilih akan memakai *preorder transversal, inorder transversal atau postorder transversal.*

```
class simpulBiner(object):
    def __init__(self, data):
        self.data = data
        self.kiri = None
        self.kanan = None

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

A = simpulBiner('Ambarawa')
B = simpulBiner('Bantul')
C = simpulBiner('Cimahi')
D = simpulBiner('Denpasar')
E = simpulBiner('Enrekang')
F = simpulBiner('Flores')
G = simpulBiner('Garut')
H = simpulBiner('Halmahera Timur')
I = simpulBiner('Indramayu')
J = simpulBiner('Jakarta')

A.kiri = B; A.kanan = C
B.kiri = D; B.kanan = E
C.kiri = F; C.kanan = G
E.kiri = H
G.kanan = I

datalist = [A.data, B.data, C.data, D.data, E.data, F.data, G.data, H.data, I.data, J.data]
level = []

def preorder(sub):
    if sub is not None:
        print(sub.data)
        preorder(sub.kiri)
        preorder(sub.kanan)
def inorder(sub):
    if sub is not None:
        inorder(sub.kiri)
        print(sub.data)
        inorder(sub.kanan)

def postord(sub):
    if sub is not None:
```

```
    if sub is not None:
        print(sub.data)
        preorder(sub.kiri)
        preorder(sub.kanan)
def inorder(sub):
    if sub is not None:
        inorder(sub.kiri)
        print(sub.data)
        inorder(sub.kanan)

def postord(sub):
    if sub is not None:
        postorder(sub.kiri)
        postorder(sub.kanan)
        print(sub.data)

def traverse(root):
    lvlist = []
    current_level = [root]
    lv = 0
    while current_level:
        #print(' '.join(str(node) for node in current_level))
        next_level = list()
        for n in current_level:
            if n.kiri:
                next_level.append(n.kiri)
                level.append(lv+1)
            if n.kanan:
                next_level.append(n.kanan)
                level.append(lv+1)
            current_level = next_level

        lv += 1
        lvlist.append(lv)
    return lvlist

def cetakDataDanLevel(root):
    traverse(A)
    print(root.data, ', Level 0')
    for i in range(len(level)):
          print(datalist[i+1], ', Level', level[i])
```

```
>>> cetakDataDanLevel(A)
Ambarawa , Level 0
Bantul , Level 1
Cimahi , Level 1
Denpasar , Level 2
Enrekang , Level 2
Flores , Level 2
Garut , Level 2
Halmahera Timur , Level 3
Indramayu , Level 3
>>>
```

Nama : Muhammad Khalif Rizaldi W

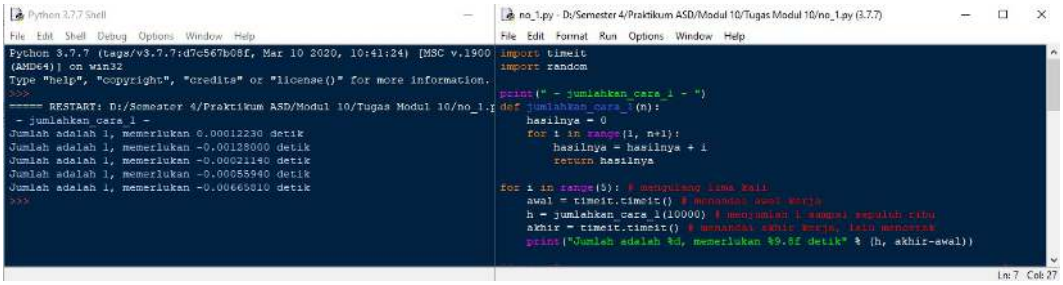
NIM : L200180217

Kelas : H

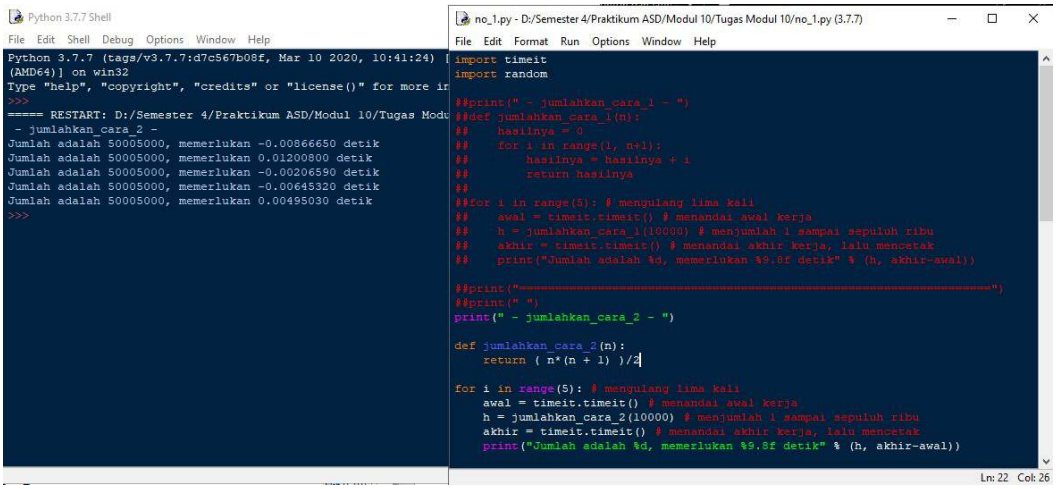
TUGAS PRAKTIKUM ASD

MODUL 10

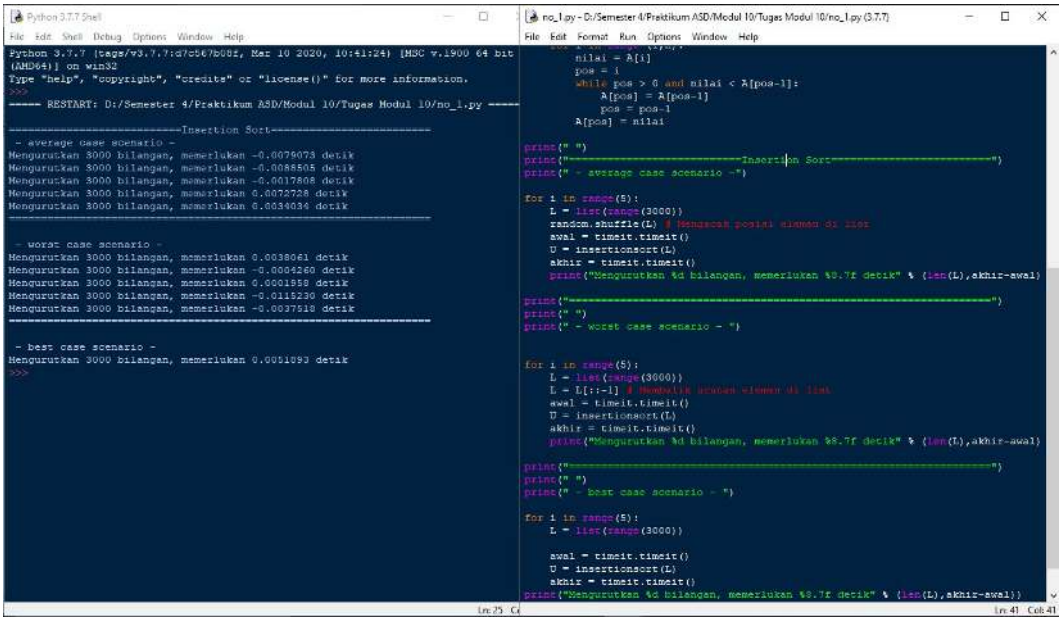
1. Kerjakan ulang contoh dan latihan di modul ini menggunakan modul timeit, yakni
   a Jumlahkan_cara_1



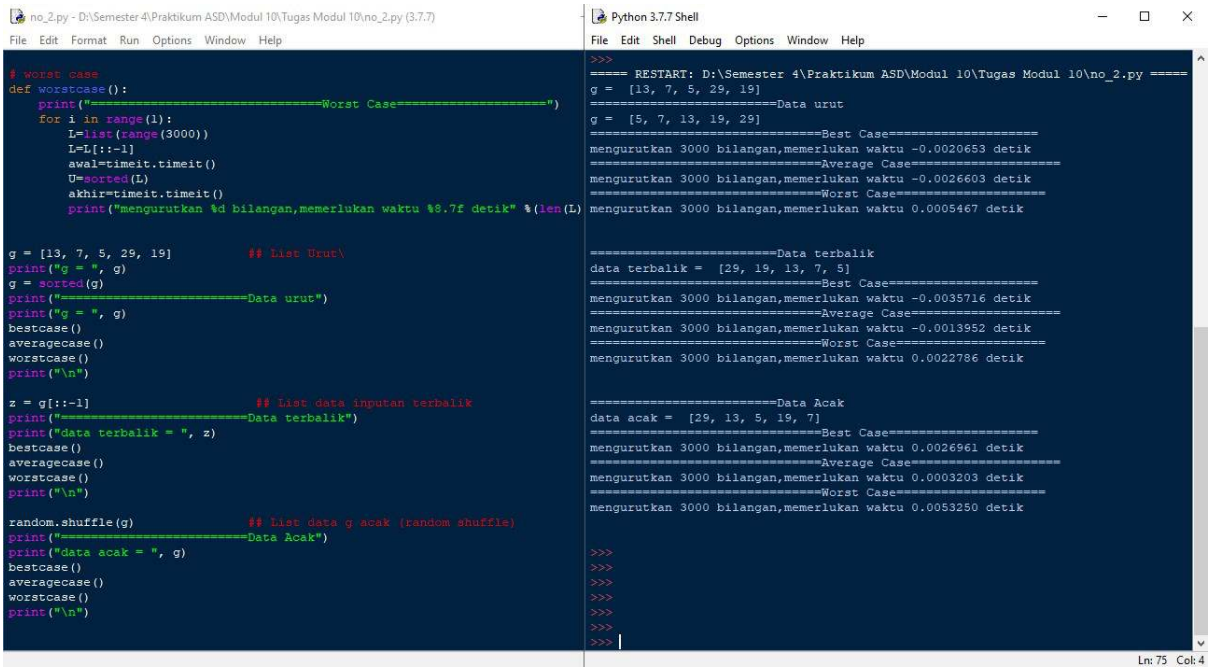
   b Jumlahkan_cara_2



   c insertionSort

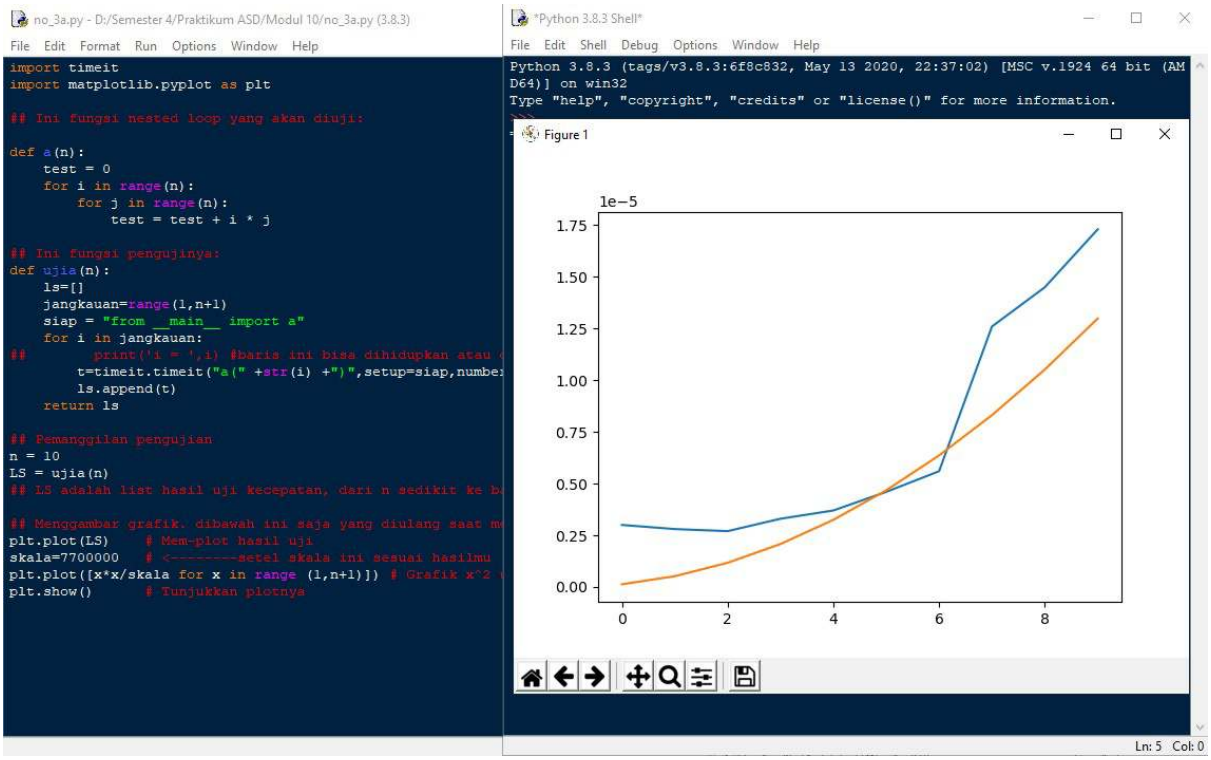


2. Python mempunyai perintah untuk mengurutkan suatu list yang memanfaatkan algoritma Timsort. Jika g adalah suatu list berisi bilangan, maka g.sort() kan mengurutkannya. Perintah yang lain sorted() mengurutkan list dan mengembalikan sebuah list baru yang sudah urut. Selidikilah fungsi sorted() ini menggunakan timeit:
   - Apakah yang merupakan best case dan average case bagi sorted( ) ?
   - confirm bahwa data input urutan terbalik bukan kasus terburuk bagi sorted( ). Bahkan dia lebih cepat dalam mengurutkannya daripada data input random



   - Dapat dibuktikan bahwa data dengan inputan terbalik bukan kasus buruk bagi sorted( ). Bahkan dia lebih cepat dalam mengurutkannya daripada data random

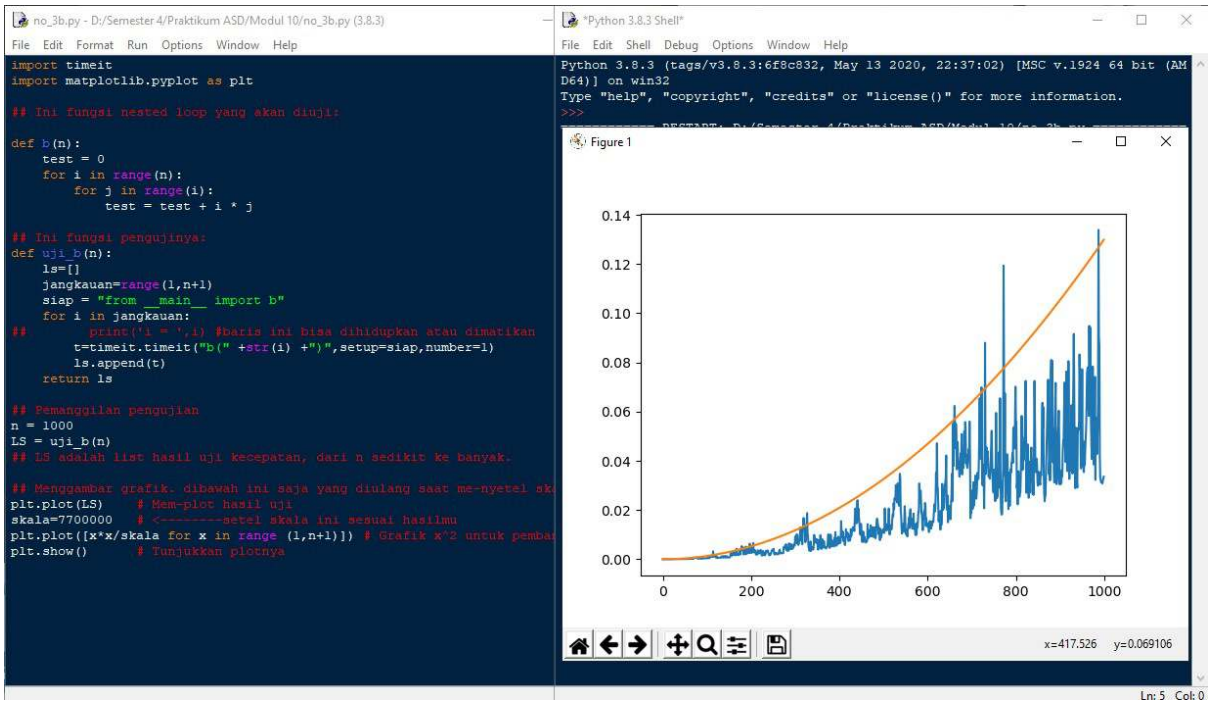
3. Untuk tiap kode berikut, tentukan running time-nya O(1), O(log n), O(n), O(n log n), $O(n^2)$ atau $O(n^3)$ atau yang lain. Untuk memulai analisis, ambil suatu nilai n tertentu lalu ikuti apa yang terjadi di kode itu
   a Loop di dalam loop keduanya sebanyak n:



   T(n) = c1+n*(n)

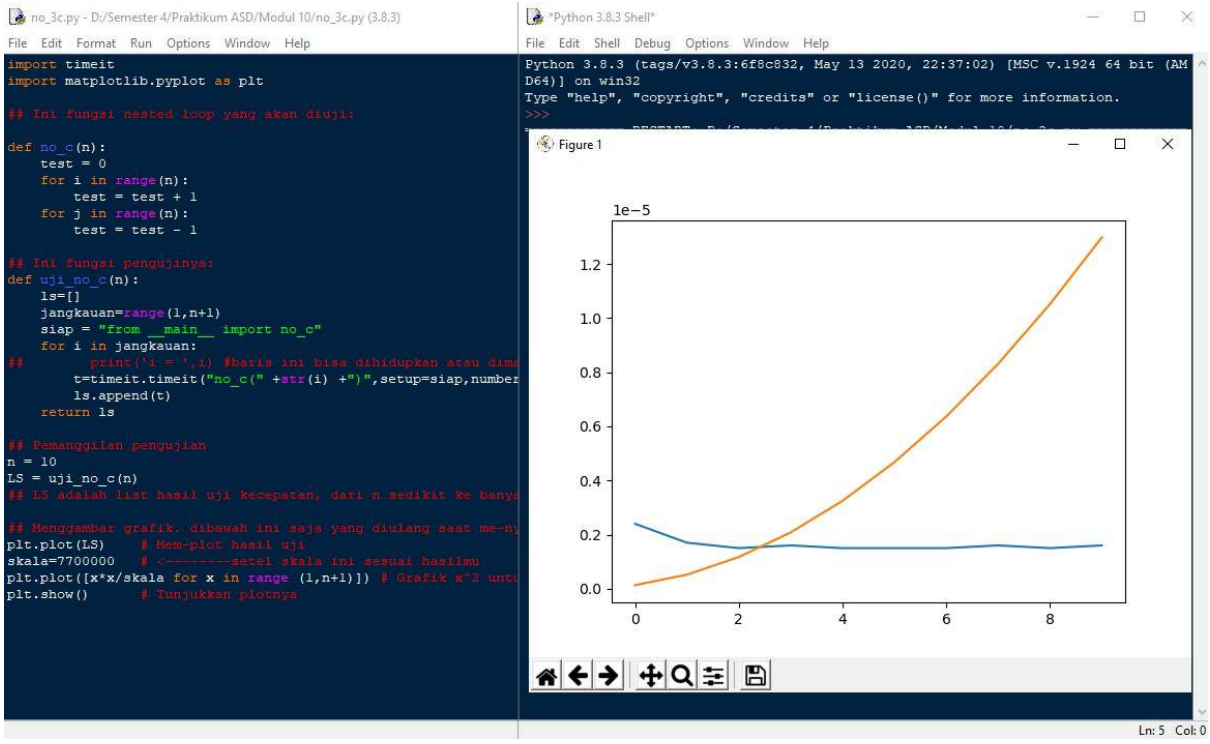
T(n) = c1+n^2
O(n) ≈ n^2
O(n^2)

b Loop di dalam loop yang dalam bergantung nilai i loop luar:



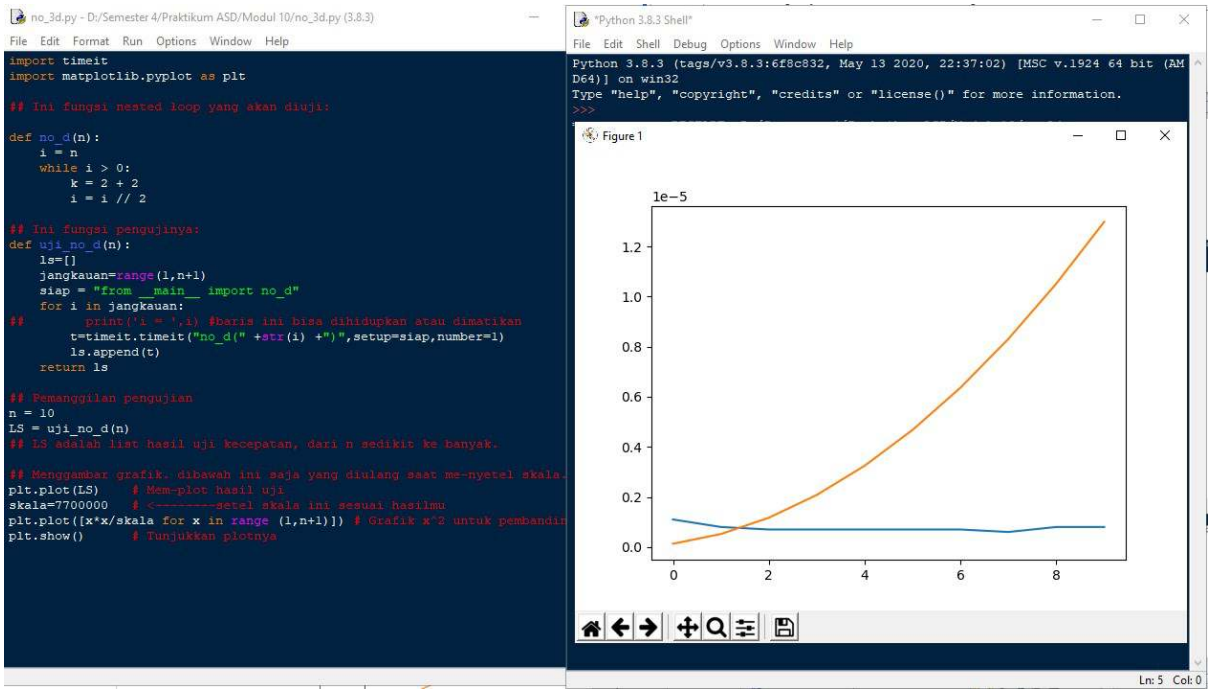
T(n) = c1+log n
T(n) = c1+log n
O(n) ≈ log n
O(log n)

c Dua loop terpisah



T(n) = c1(n) + c2(n)
T(n) = n + n
O(n) ≈ n
O(n)

d While loop yang dipangkas sepatuh tiap putaran

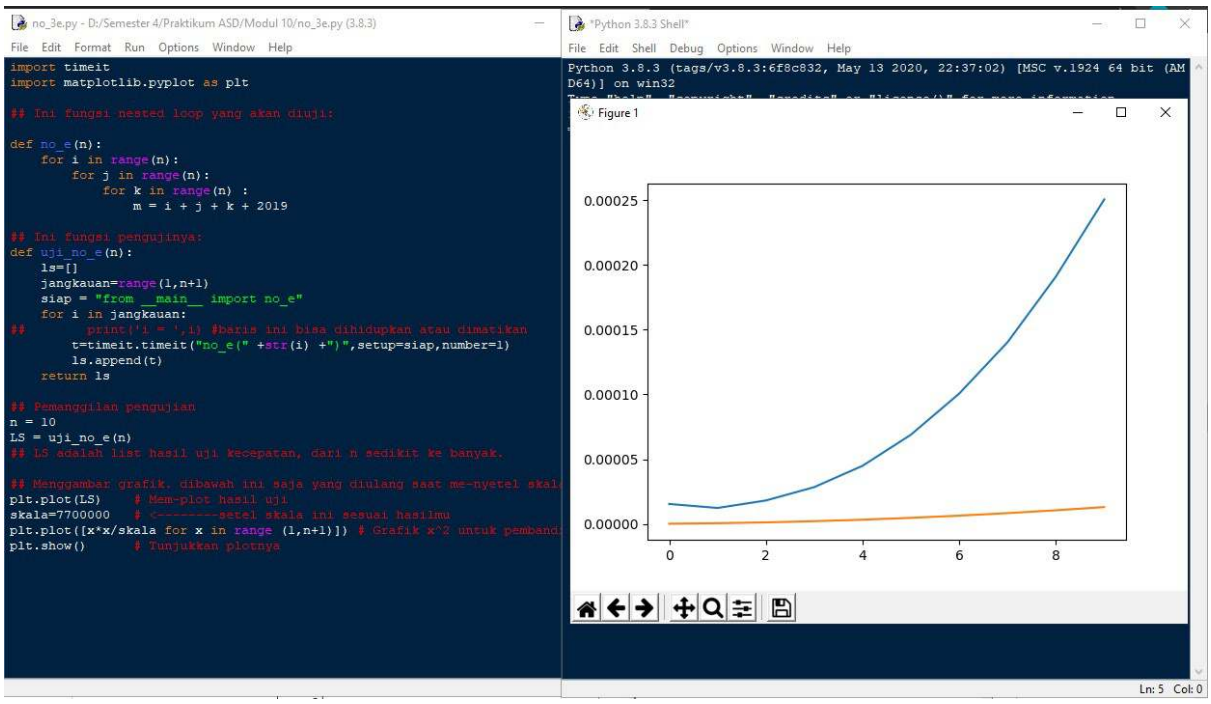


T(n) = c1(1) + c2(1)

O(n) ≈ 1

O(1)

e Loop in a loop in a loop, ketigaya sebanyak n



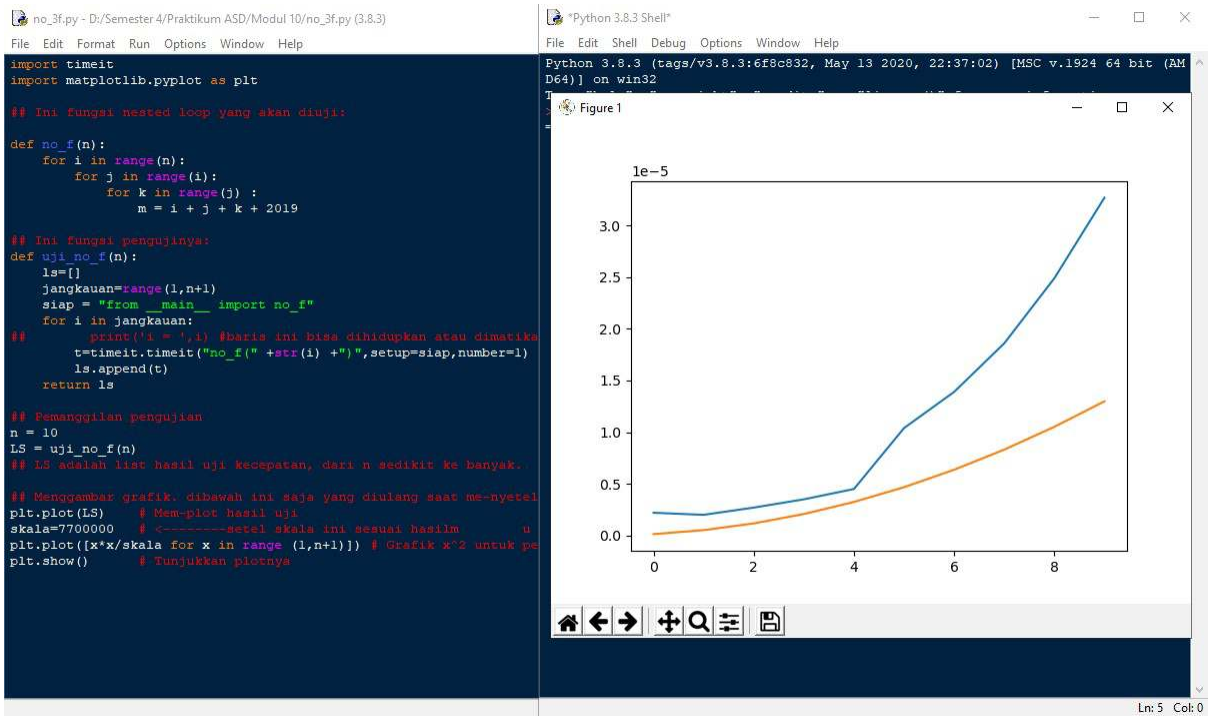
T(n) = n*(n*(n))

T(n) = n^3

O(n) ≈ n^3

O(n^3)

f Loop in a loop, dengan loop dalam sebanyak nilai loop luar terdekat



T(n) = c1(n) + c2(n) + c3(n)
T(n) = n + n + n
O(n) ≈ n
O(n)

g Fungsi ini

O(n log n)

4. Urutkan dari yang pertumbuhan kompleksitasnya lambat ke yang cepat
   $\log_4 n < 10\log_2 n < n \log_2 n < 2^{\log 2n} < 5n^2 < n^3 < 12n^6 < 4^n$

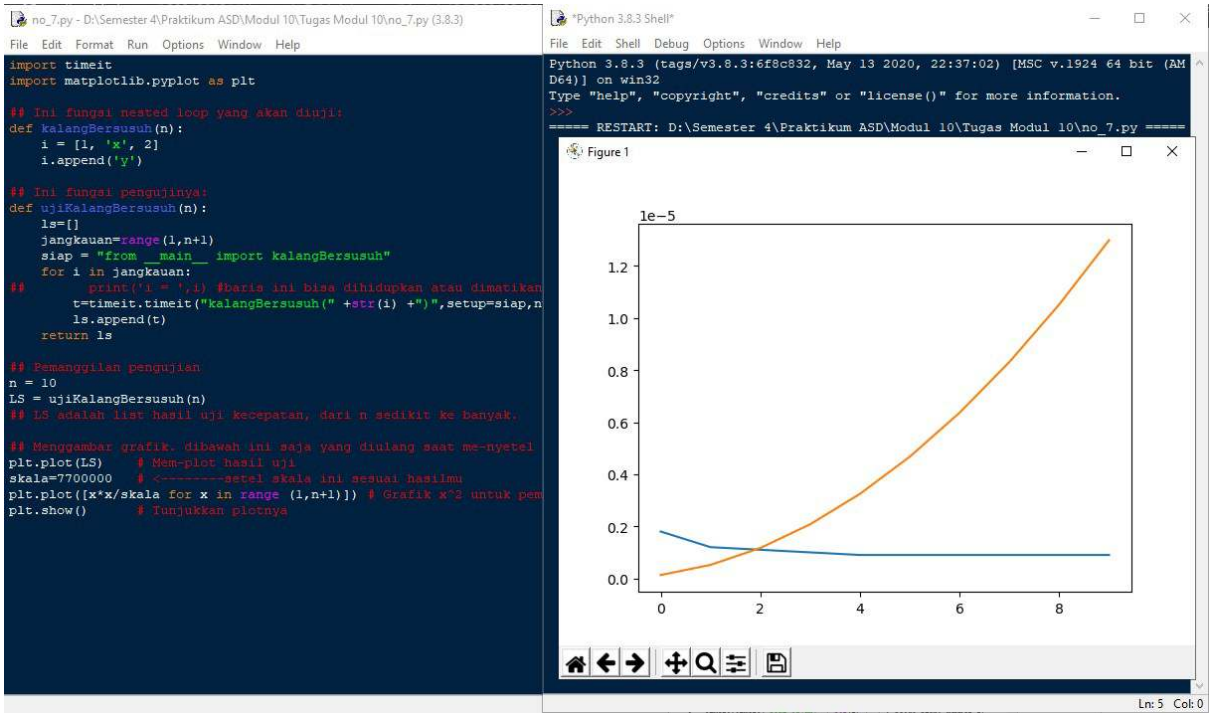
5. Tentukan O(.) dari fungi-fungsi berikut yang mewakili banyaknya langkah yang diperlakukan untuk beberapa algoritma
   a T(n) = n^2 + 32n + 8 = O(n^2)
   b T(n) = 87n + 8n = O(n)

c	T(n)	= 4n + 5n log n + 102	= O(n log n)
d	T(n)	= log n + 3n^2 + 88	= O(n^2)
e	T(n)	= 3(2^n) + n^2 + 647	= O(2^n)
f	T(n, k)	= kn + log k	= O(kn)
g	T(n, k)	= 8n + k log n + 800	= O(n)
h	T(n, k)	= 100kn + n	= O(kn)

6. Carilah di internet, kompelsitas metode pada object list di python.
   - Google python list method complexity. Lihat juga bagian “Images” –nya
   - Kunjungi https://wiki.python.org/moin/TimeComplexity

Operation	Average Case	Amortized Worst Case
Copy	O(n)	O(n)
Append[1]	O(1)	O(1)
Pop last	O(1)	O(1)
Pop intermediate	O(k)	O(k)
Insert	O(n)	O(n)
Get Item	O(1)	O(1)
Set Item	O(1)	O(1)
Delete Item	O(n)	O(n)
Iteration	O(n)	O(n)
Get Slice	O(k)	O(k)
Del Slice	O(n)	O(n)
Set Slice	O(k+n)	O(k+n)
Extend[1]	O(k)	O(k)
Sort	O(n log n)	O(n log n)
Multiply	O(nk)	O(nk)
x in s	O(n)	
min(s), max(s)	O(n)	
Get Length	O(1)	O(1)

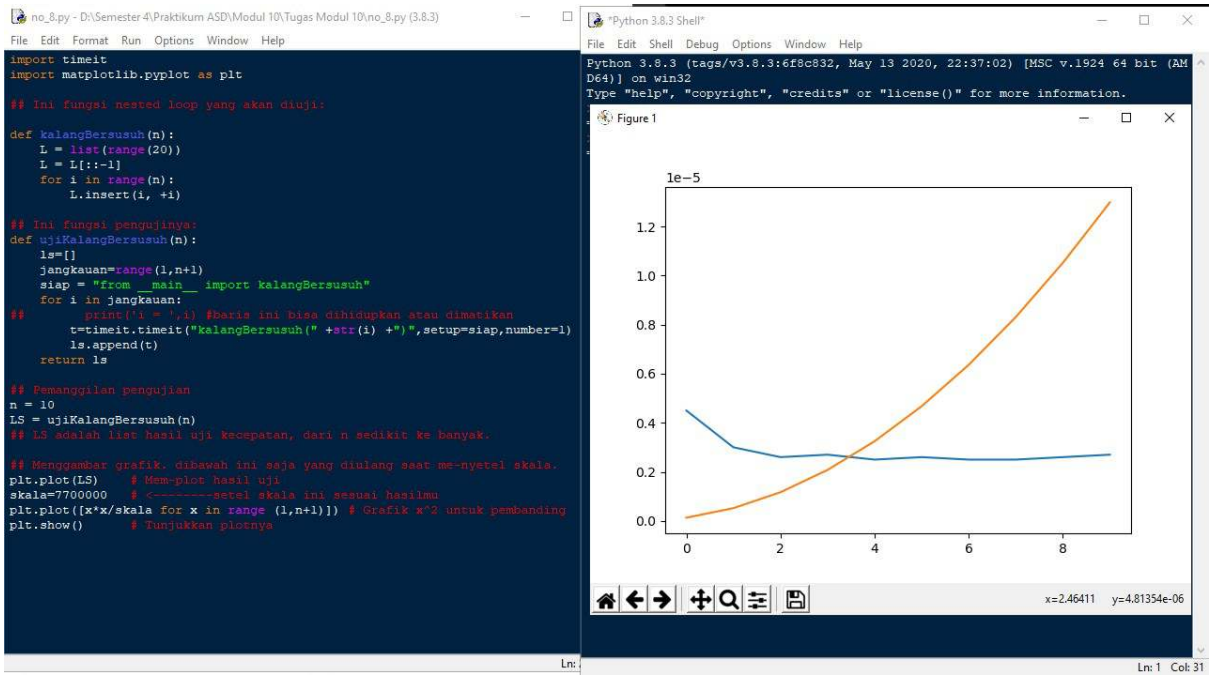
7. Buatlah suatu ujicoba untuk mengkonfirmasi bahwa metode append() adalah O(1). Gunakan timeit dan matplotib seperti sebelumnya.



```
import timeit
import matplotlib.pyplot as plt

## Ini fungsi nested loop yang akan diuji:
def kalangBersusuh(n):
    i = [1, 'x', 2]
    i.append('y')

## Ini fungsi pengujinya:
def ujiKalangBersusuh(n):
    ls=[]
    jangkauan=range(1,n+1)
    siap = "from __main__ import kalangBersusuh"
    for i in jangkauan:
##        print('i = ',i) #baris ini bisa dihidupkan atau dimatikan
        t=timeit.timeit("kalangBersusuh(" +str(i) +")",setup=siap,n
        ls.append(t)
    return ls

## Pemanggilan pengujian
n = 10
LS = ujiKalangBersusuh(n)
## LS adalah list hasil uji kecepatan, dari n sedikit ke banyak.

## Menggambar grafik. dibawah ini saja yang diulang saat me-nyetel
plt.plot(LS)     # Mem-plot hasil uji
skala=7700000    # <--------setel skala ini sesuai hasilmu
plt.plot([x*x/skala for x in range (1,n+1)]) # Grafik x^2 untuk pem
plt.show()       # Tunjukkan plotnya
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Semester 4\Praktikum ASD\Modul 10\Tugas Modul 10\no_7.py =====
```

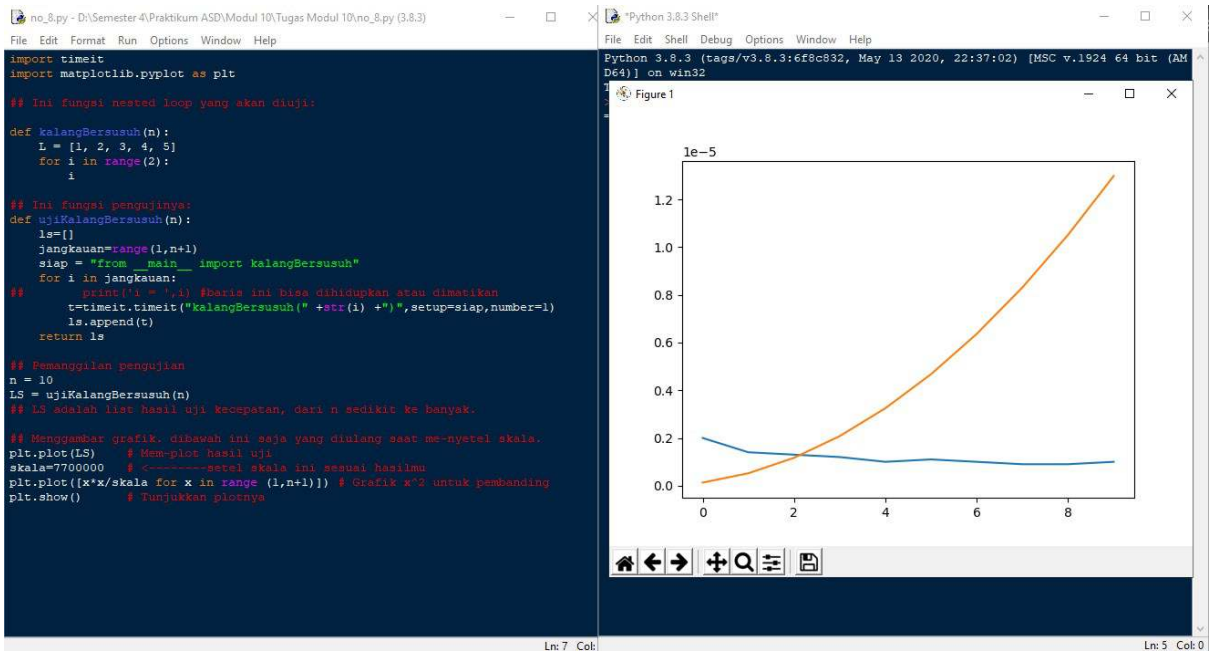
8. Buatlah suatu ujicoba untuk mengkonfirmasi bahwa metode insert() adalah O(n). Gunakan timeit dan matplotib seperti sebelumnya.



```
import timeit
import matplotlib.pyplot as plt

## Ini fungsi nested loop yang akan diuji:

def kalangBersusuh(n):
    L = list(range(20))
    L = L[::-1]
    for i in range(n):
        L.insert(i, +i)

## Ini fungsi pengujinya:
def ujiKalangBersusuh(n):
    ls=[]
    jangkauan=range(1,n+1)
    siap = "from __main__ import kalangBersusuh"
    for i in jangkauan:
##        print('i = ',i) #baris ini bisa dihidupkan atau dimatikan
        t=timeit.timeit("kalangBersusuh(" +str(i) +")",setup=siap,number=1)
        ls.append(t)
    return ls

## Pemanggilan pengujian
n = 10
LS = ujiKalangBersusuh(n)
## LS adalah list hasil uji kecepatan, dari n sedikit ke banyak.

## Menggambar grafik. dibawah ini saja yang diulang saat me-nyetel skala.
plt.plot(LS)     # Mem-plot hasil uji
skala=7700000    # <--------setel skala ini sesuai hasilmu
plt.plot([x*x/skala for x in range (1,n+1)]) # Grafik x^2 untuk pembanding
plt.show()       # Tunjukkan plotnya
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
```

9. Buatlah suatu ujicoba untuk mengkonfirmasi bahwa untuk memeriksa apakah suatu nilai berada di suatu list mempunya kompleksitas O(n). Gunakan timeit dan matplotib seperti sebelumnya.



10. Carilah di internet, kompelsitas metode pada object dict di python.

Operation	Average Case	Amortized Worst Case
k in d	O(1)	O(n)
Copy[2]	O(n)	O(n)
Get Item	O(1)	O(n)
Set Item[1]	O(1)	O(n)
Delete Item	O(1)	O(n)
Iteration[2]	O(n)	O(n)

11. Selain notasi big-O O(.) ada pula notasi big-Theta Θ(.) dan notasi big-Omega Ω (.)
Apakah beda diantara ketiganya?
    - Big O dilambangkan dengan notasi **O(...)** merupakan keadaan terburuk (worst case). Kinerja seubuah algoritma biasanya diukur menggunakan patokan keadaan Big-O ini. Merupakan notasi asymptotic untuk batas fungsi dari atas dan bawah dengan Berperilaku mirip dengan ≤ operator untuk tingkat pertumbuhan.
    - Big Theta dilmbangkan dengan notasi **Θ(...)** merupakan notasi asymptotic untuk batas atas dan bawah dengan keadaan terbaik (best case). Menyatakan persamaan pada pertumbuhan f (n) hingga faktor konstan (lebih lanjut tentang ini nanti). Berperilaku mirip dengan = operator untuk tingkat pertumbuhan
    - Big Omega dilambangkan dengan notasi **Ω(...)** merupakan notasi asymptotic untuk batas bawah dengan keadaan rata-rata(average case) yang berperilaku mirip dengan ≥operator untuk tingkat pertumbuhan.

12. Apa yang dimaksud dengan amortized analysis dalam analisis algoritma?
Jawab: Amortized analysis adalah metode untuk menganalisis kompleksitas algoritma yang diberikan, atau berapa banyak resource nya terutama waktu atau memori yang diperlukan untuk mengeksekusi. Dapat ditunjukkan dengan waktu rata-rata yang diperlukan unyuk melakukan satu urutan operasi pada struktur data terhadap keseluruhan operasi yang dilakukan.