

**PRAKTIKUM ALGORITMA DAN  
STRUKTUR DATA**

**Rekapan Laporan**



**Disusun oleh:**

**DONI WAHYU SAPUTRO**

**L200200169**

**G**

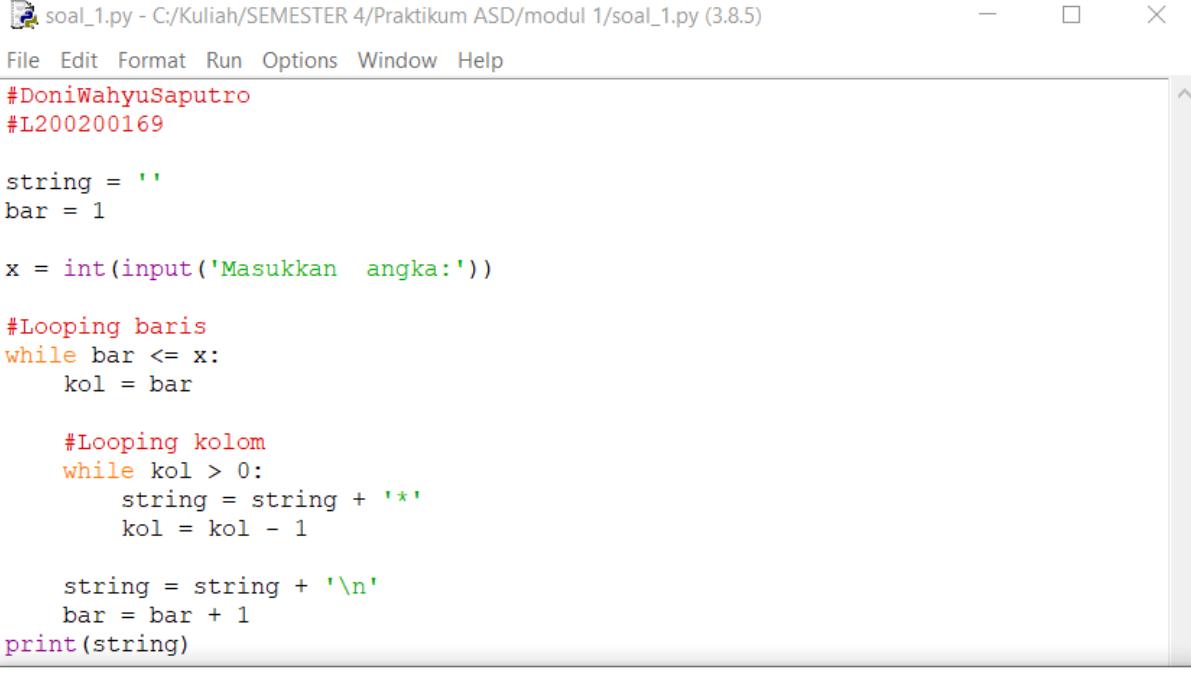
**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

# **MODUL 1**

## Soal 1



soal\_1.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal\_1.py (3.8.5)

File Edit Format Run Options Window Help

```
#DoniWahyuSaputro
#L200200169

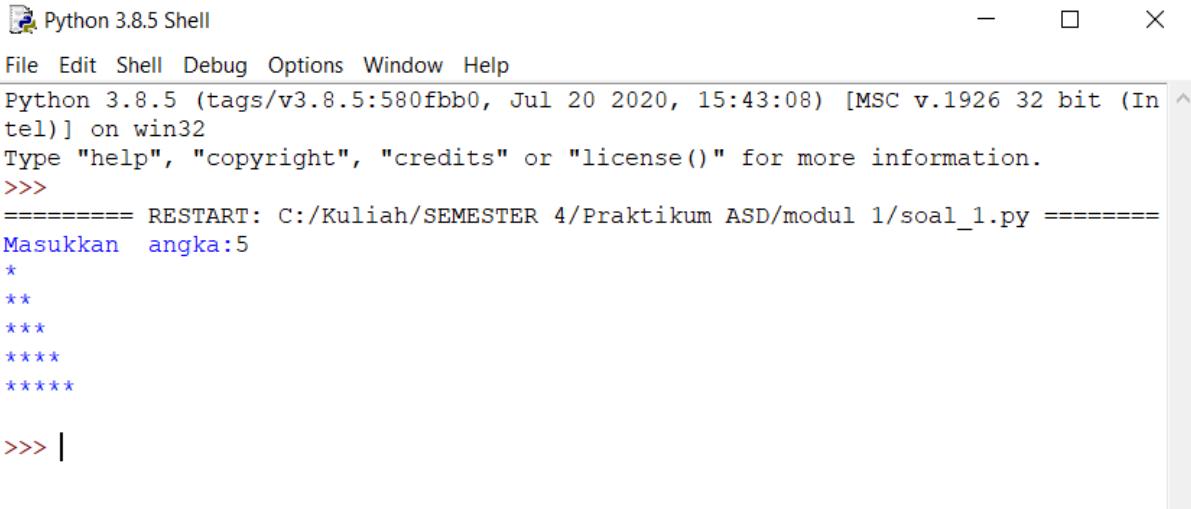
string = ''
bar = 1

x = int(input('Masukkan angka:'))

#Looping baris
while bar <= x:
    kol = bar

    #Looping kolom
    while kol > 0:
        string = string + '*'
        kol = kol - 1

    string = string + '\n'
    bar = bar + 1
print(string)
```

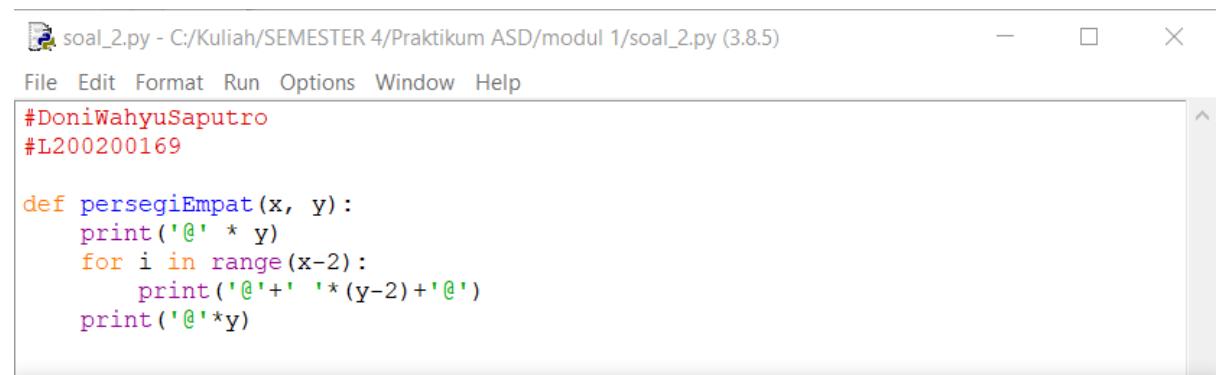


Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

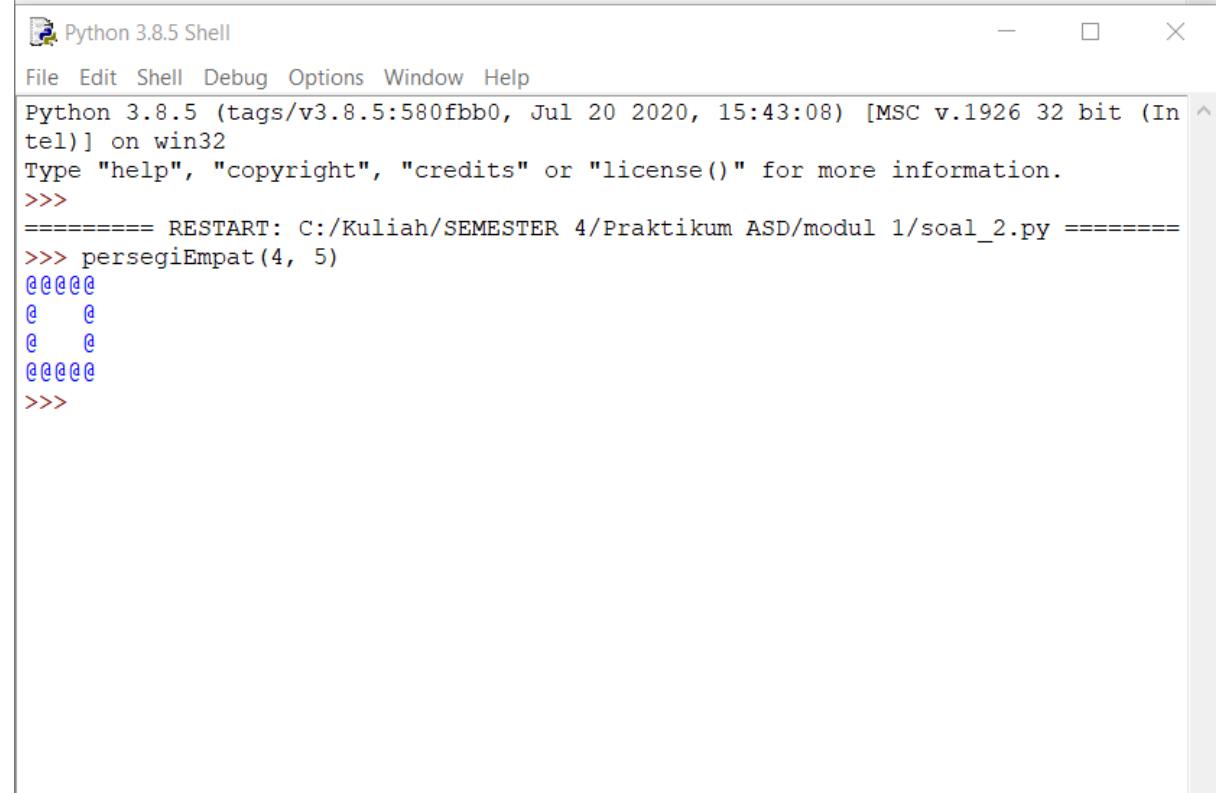
```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_1.py ======
Masukkan angka:5
*
**
***
****
*****
>>> |
```

## Soal 2



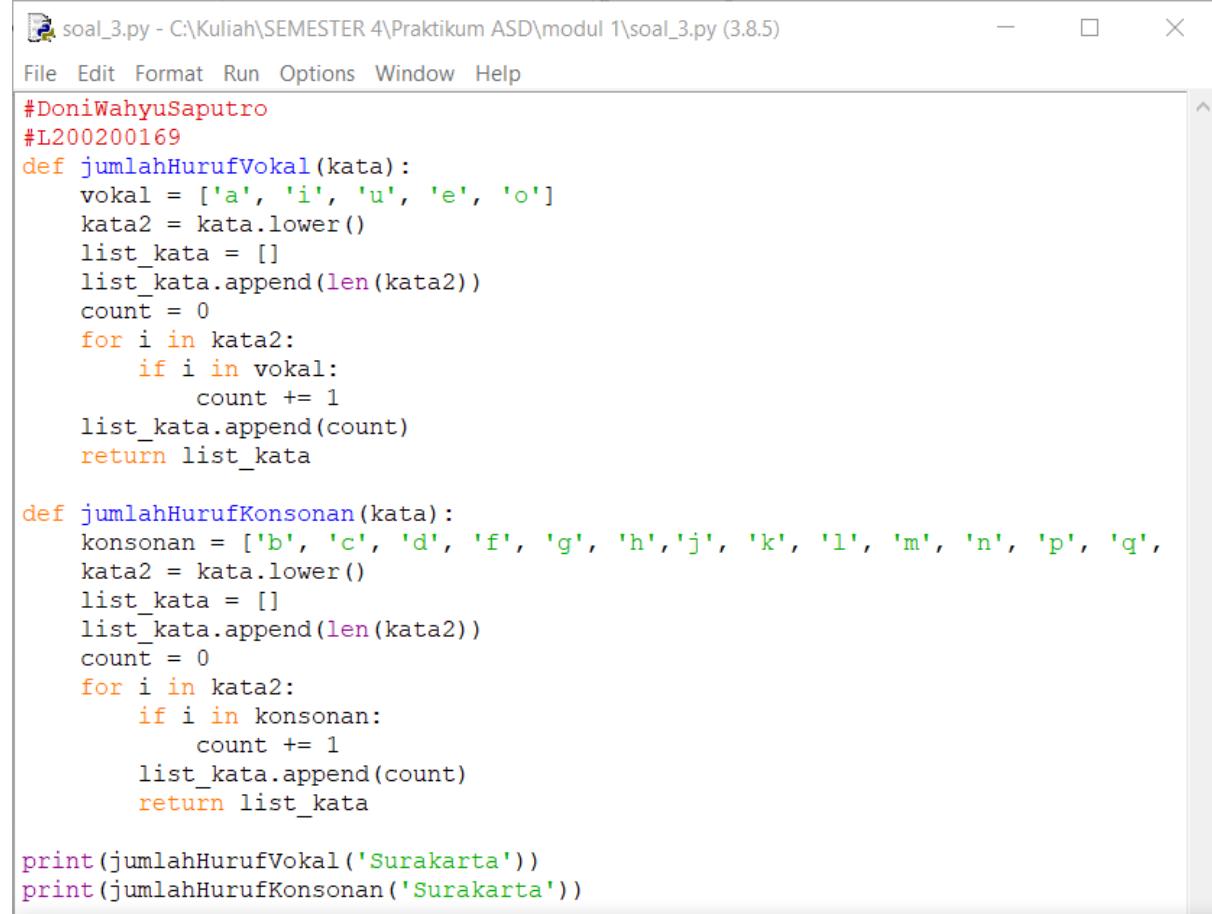
```
#DoniWahyuSaputro
#L200200169

def persegiEmpat(x, y):
    print('@' * y)
    for i in range(x-2):
        print('@' + ' ' * (y-2) + '@')
    print('@'*y)
```



```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_2.py =====
>>> persegiEmpat(4, 5)
@#####
@   @
@   @
@#####
>>>
```

### Soal 3



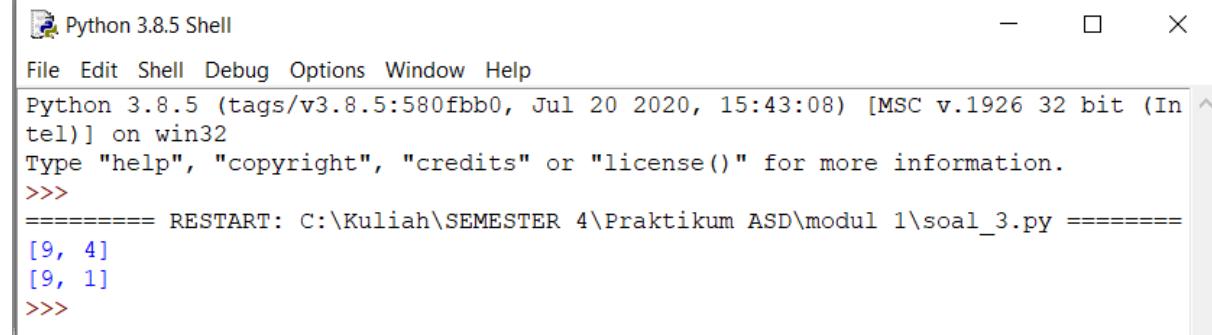
soal\_3.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 1\soal\_3.py (3.8.5)

File Edit Format Run Options Window Help

```
#DoniWahyuSaputro
#L200200169
def jumlahHurufVokal(kata):
    vokal = ['a', 'i', 'u', 'e', 'o']
    kata2 = kata.lower()
    list_kata = []
    list_kata.append(len(kata2))
    count = 0
    for i in kata2:
        if i in vokal:
            count += 1
    list_kata.append(count)
    return list_kata

def jumlahHurufKonsonan(kata):
    konsonan = ['b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm', 'n', 'p', 'q',
    kata2 = kata.lower()
    list_kata = []
    list_kata.append(len(kata2))
    count = 0
    for i in kata2:
        if i in konsonan:
            count += 1
    list_kata.append(count)
    return list_kata

print(jumlahHurufVokal('Surakarta'))
print(jumlahHurufKonsonan('Surakarta'))
```

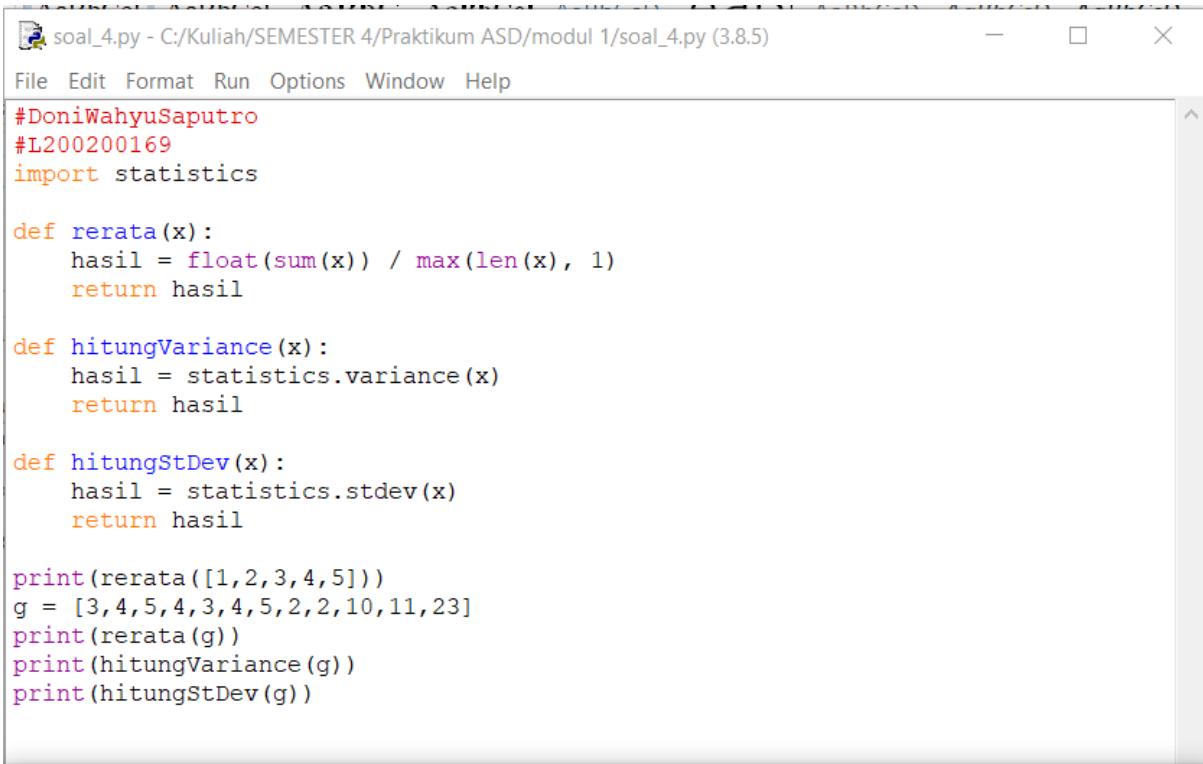


Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 1\soal_3.py ======
[9, 4]
[9, 1]
>>>
```

#### Soal 4



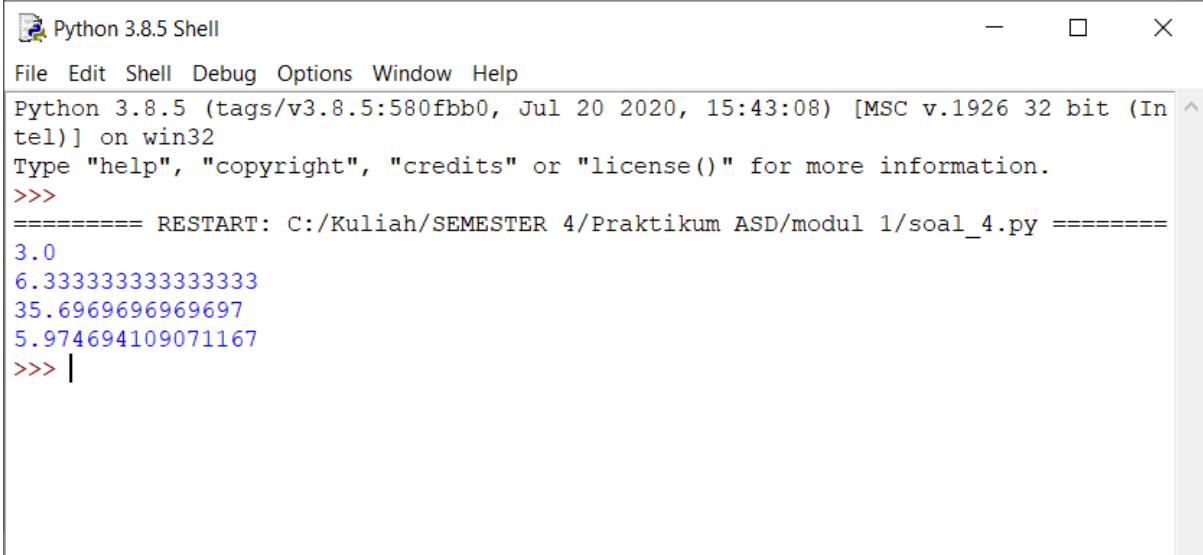
```
#DoniWahyuSaputro
#L200200169
import statistics

def rerata(x):
    hasil = float(sum(x)) / max(len(x), 1)
    return hasil

def hitungVariance(x):
    hasil = statistics.variance(x)
    return hasil

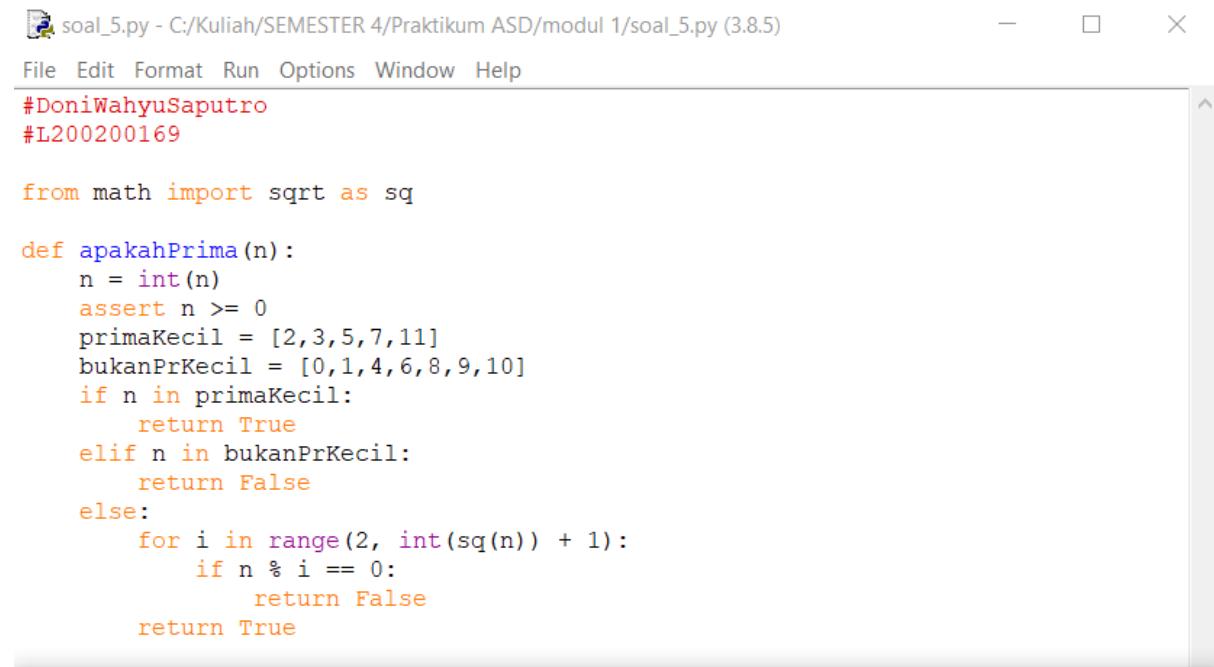
def hitungStDev(x):
    hasil = statistics.stdev(x)
    return hasil

print(rerata([1,2,3,4,5]))
g = [3,4,5,4,3,4,5,2,2,10,11,23]
print(rerata(g))
print(hitungVariance(g))
print(hitungStDev(g))
```



```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_4.py =====
3.0
6.333333333333333
35.6969696969697
5.974694109071167
>>> |
```

## Soal 5



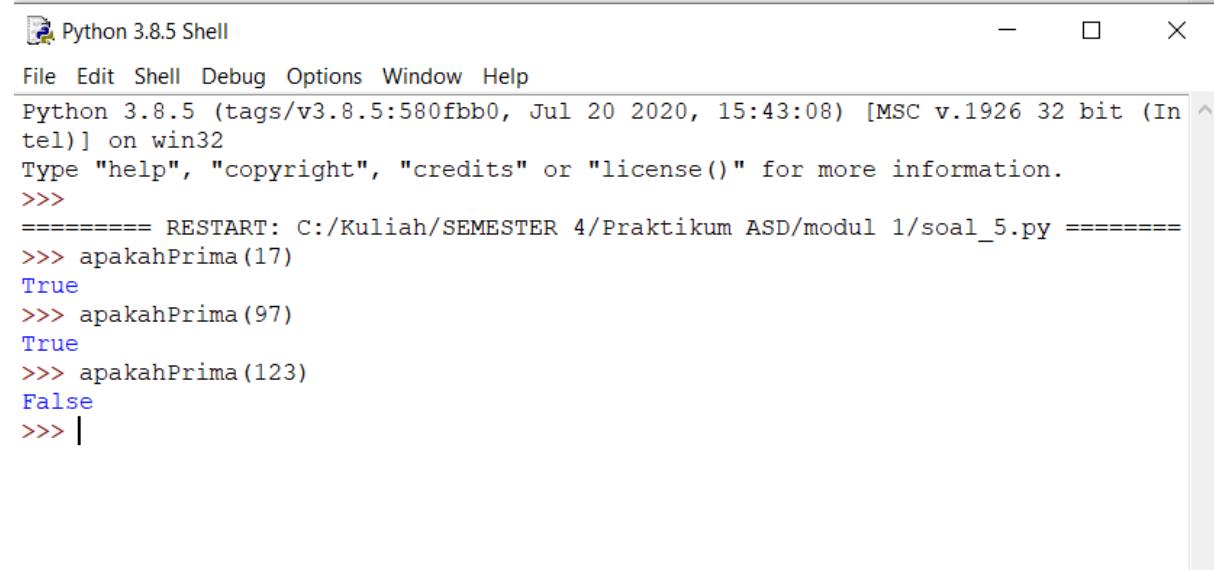
The screenshot shows a Python code editor window titled "soal\_5.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal\_5.py (3.8.5)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```
#DoniWahyuSaputro
#L200200169

from math import sqrt as sq

def apakahPrima(n):
    n = int(n)
    assert n >= 0
    primaKecil = [2,3,5,7,11]
    bukanPrKecil = [0,1,4,6,8,9,10]
    if n in primaKecil:
        return True
    elif n in bukanPrKecil:
        return False
    else:
        for i in range(2, int(sq(n)) + 1):
            if n % i == 0:
                return False
        return True
```

---



The screenshot shows a Python shell window titled "Python 3.8.5 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell output is as follows:

```
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_5.py =====
>>> apakahPrima(17)
True
>>> apakahPrima(97)
True
>>> apakahPrima(123)
False
>>> |
```

## Soal 6

The screenshot shows a Python code editor window titled "soal\_6.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal\_6.py (3.8.5)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code itself defines a function to check if a number is prime and another to generate a list of prime numbers between two given values.

```
#DoniWahyuSaputro
#L200200169

def is_prima(x):
    for i in range(2, x):
        if x % i == 0:
            return False

    return True

def bilangan_prima (awal, akhir):
    list_bilangan_prima = []

    for x in range(awal, akhir + 1):
        if is_prima(x):
            list_bilangan_prima.append(x)

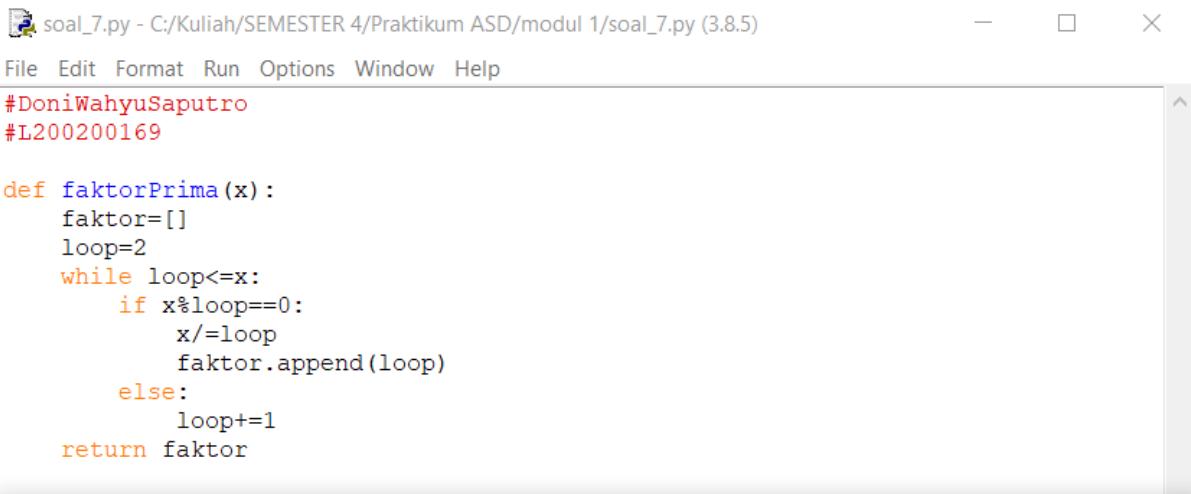
    return list_bilangan_prima

print(bilangan_prima(2, 1000))
```

The screenshot shows a Python shell window titled "Python 3.8.5 Shell". It displays the output of running the "soal\_6.py" script, which prints a list of prime numbers from 2 to 1000.

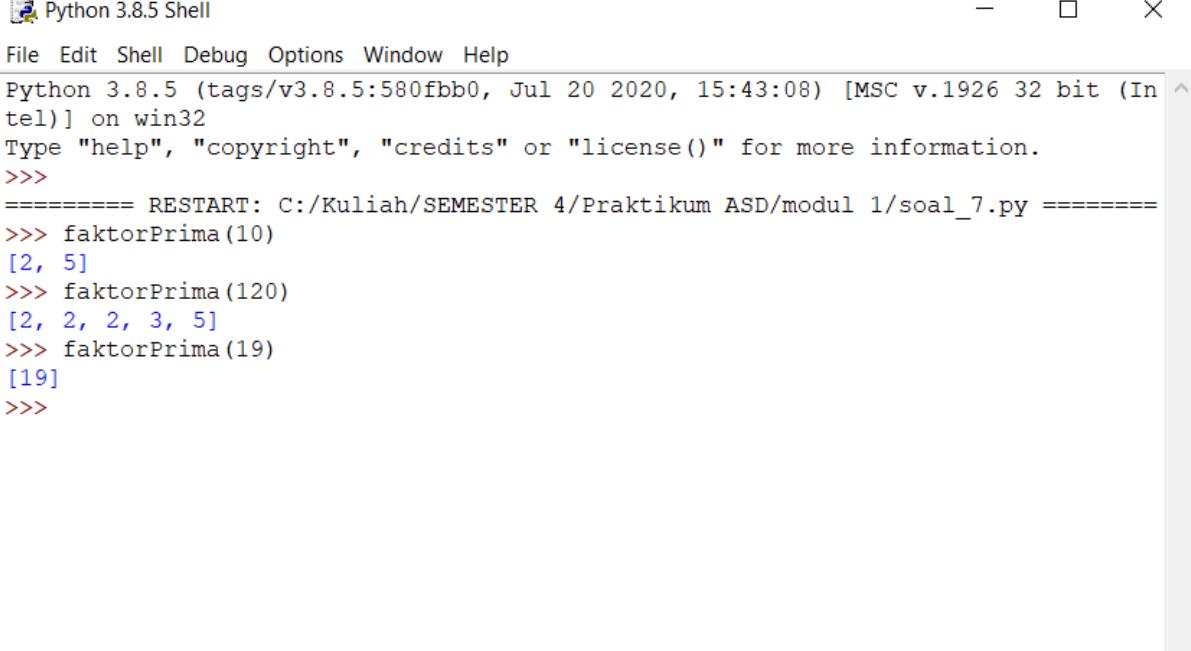
```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_6.py =====
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163
, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251
, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349
, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443
, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557
, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647
, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757
, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863
, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983
, 991, 997]
>>>
```

## Soal 7



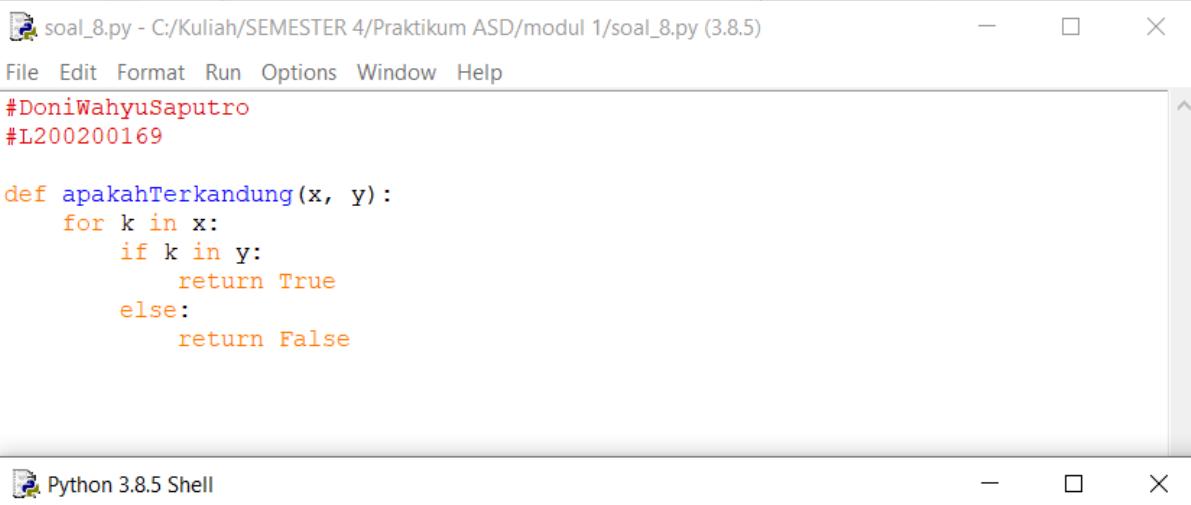
```
#DoniWahyuSaputro
#L200200169

def faktorPrima(x):
    faktor=[]
    loop=2
    while loop<=x:
        if x%loop==0:
            x/=loop
            faktor.append(loop)
        else:
            loop+=1
    return faktor
```



```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_7.py =====
>>> faktorPrima(10)
[2, 5]
>>> faktorPrima(120)
[2, 2, 2, 3, 5]
>>> faktorPrima(19)
[19]
>>>
```

## Soal 8

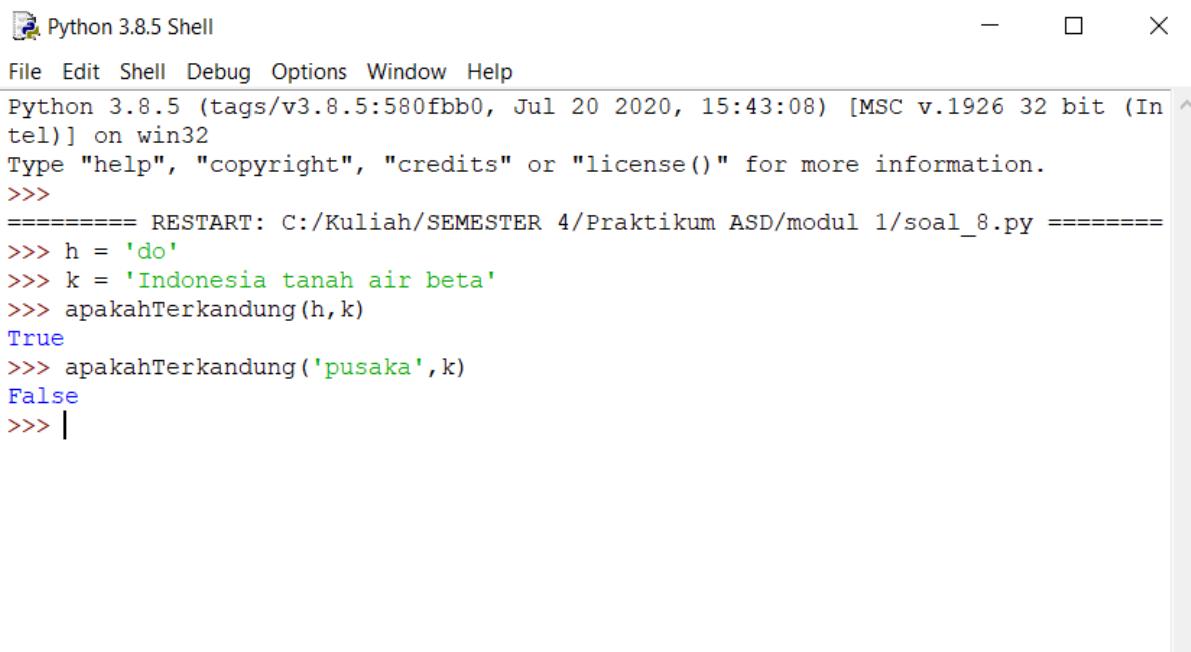


soal\_8.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal\_8.py (3.8.5)

File Edit Format Run Options Window Help

```
#DoniWahyuSaputro
#L200200169

def apakahTerkandung(x, y):
    for k in x:
        if k in y:
            return True
        else:
            return False
```



Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_8.py =====
>>> h = 'do'
>>> k = 'Indonesia tanah air beta'
>>> apakahTerkandung(h, k)
True
>>> apakahTerkandung('pusaka', k)
False
>>> |
```

## Soal 9

soal\_9.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal\_9.py (3.8.5)

- □ >

File Edit Format Run Options Window Help

#DoniWahyuSaputro

#L200200169

```
for i in range(1, 100):
    if (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)
```

Python 3.8.5 Shell

- □ ;

File Edit Shell Debug Options Window Help

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32

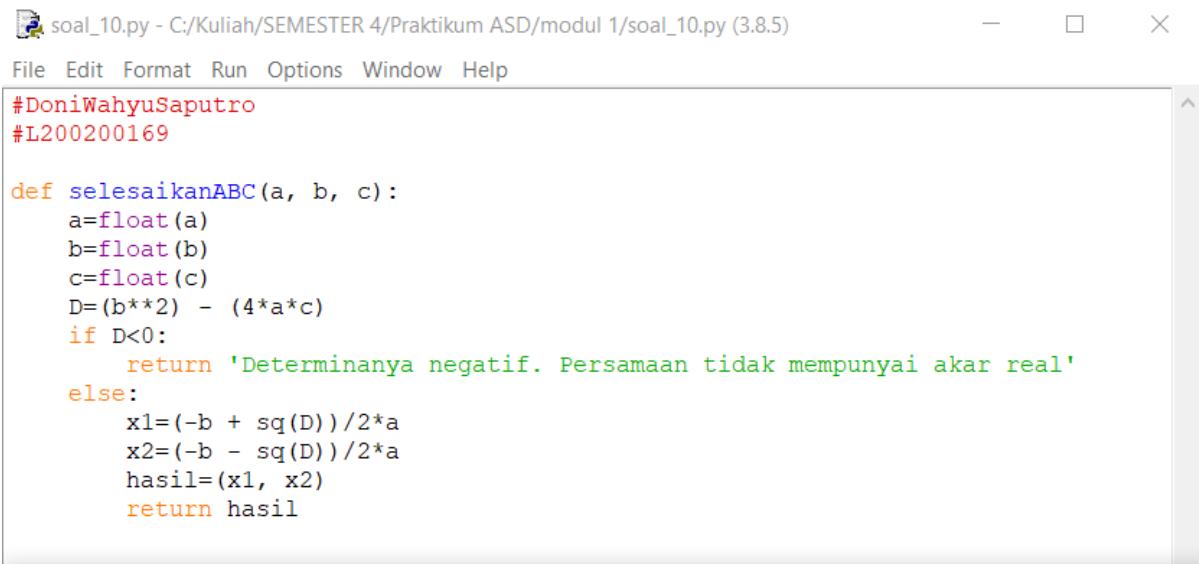
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal\_9.py =====

```
1
2
Python
4
UMS
Python
7
8
Python
UMS
11
Python
13
14
Python UMS
16
17
Python
19
UMS
Python
22
23
Python
UMS
26
```

## Soal 10

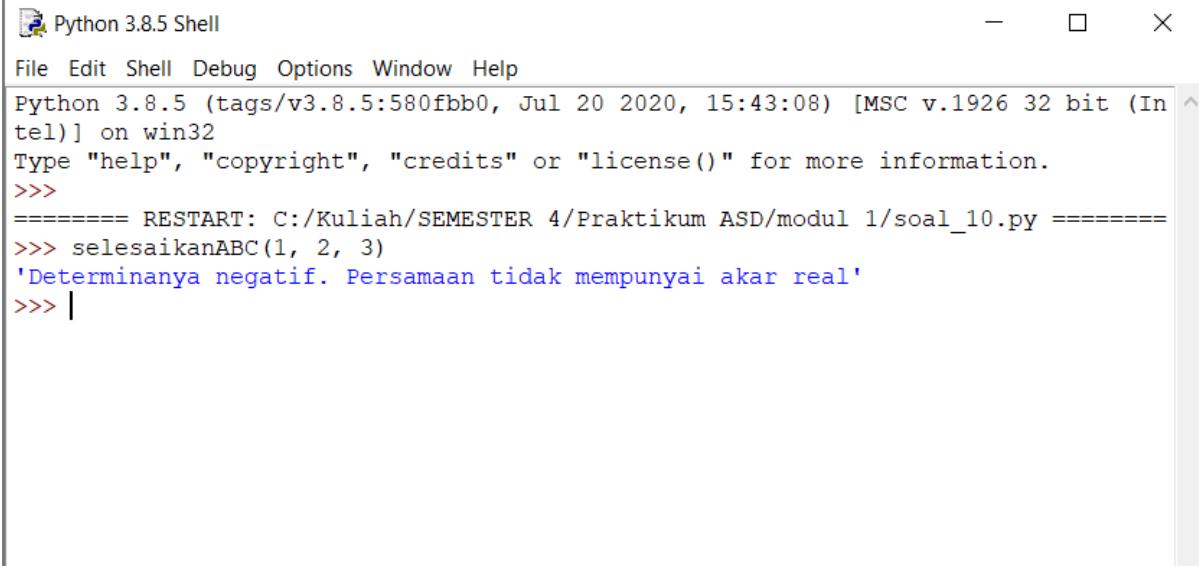


soal\_10.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal\_10.py (3.8.5)

File Edit Format Run Options Window Help

```
#DoniWahyuSaputro
#L200200169

def selesaikanABC(a, b, c):
    a=float(a)
    b=float(b)
    c=float(c)
    D=(b**2) - (4*a*c)
    if D<0:
        return 'Determinanya negatif. Persamaan tidak mempunyai akar real'
    else:
        x1=(-b + sq(D))/2*a
        x2=(-b - sq(D))/2*a
        hasil=(x1, x2)
        return hasil
```

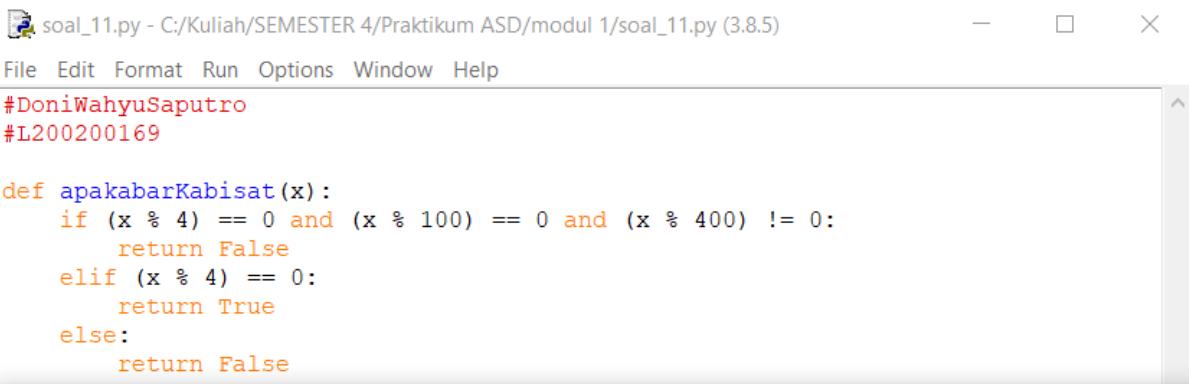


Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

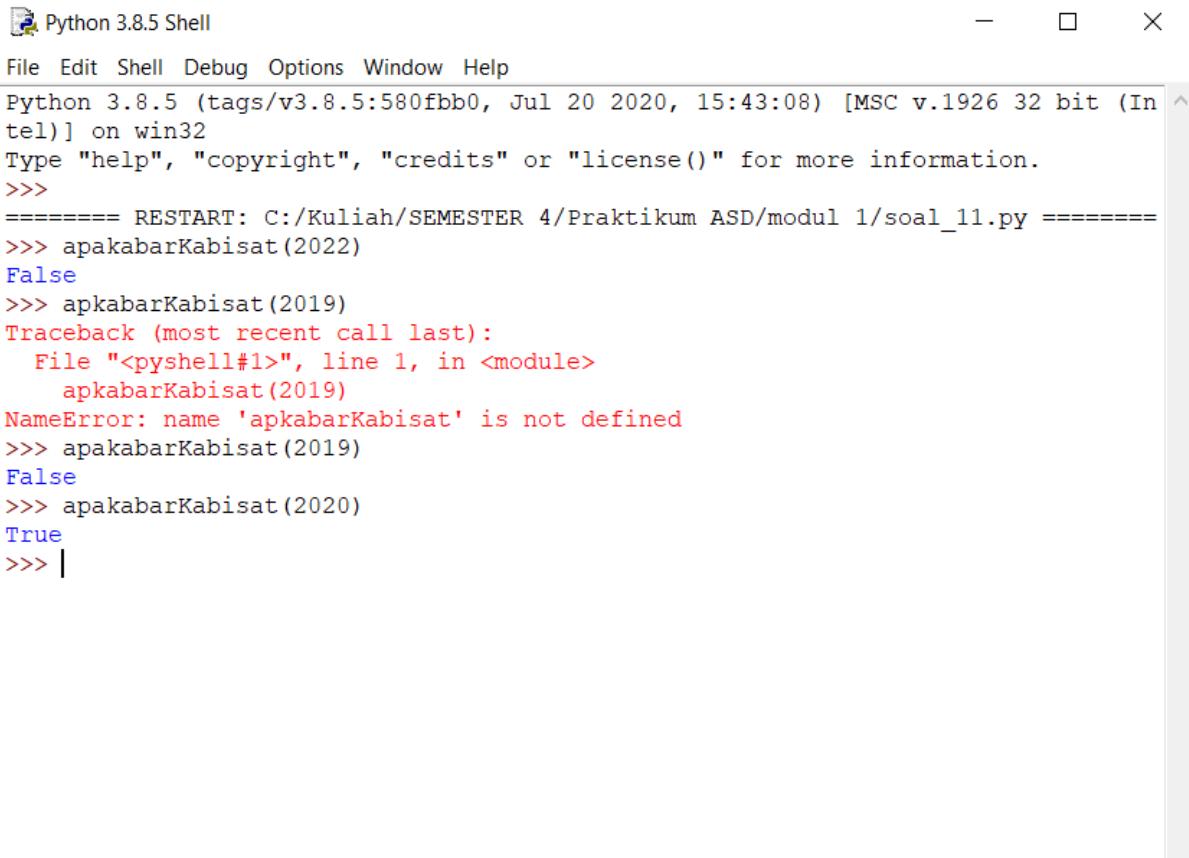
```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_10.py ======
>>> selesaikanABC(1, 2, 3)
'Determinanya negatif. Persamaan tidak mempunyai akar real'
>>> |
```

## Soal 11



```
soal_11.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_11.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

def apakabarKabisat(x):
    if (x % 4) == 0 and (x % 100) == 0 and (x % 400) != 0:
        return False
    elif (x % 4) == 0:
        return True
    else:
        return False
```



```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_11.py =====
>>> apakabarKabisat(2022)
False
>>> apakabarKabisat(2019)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    apakabarKabisat(2019)
NameError: name 'apakabarKabisat' is not defined
>>> apakabarKabisat(2019)
False
>>> apakabarKabisat(2020)
True
>>> |
```

## Soal 12

The screenshot shows a code editor on the left containing a Python script named 'soal\_12.py' and a Python shell window on the right. The script generates a random number between 1 and 100 and asks the user to guess it. The shell shows the interaction where the user guesses numbers and receives feedback (too low or too high) until they guess correctly.

```
#doniWahyuSaputro
#L200200169

from random import randint
quiz = randint(1, 100)

print('Saya menyimpan angka bulat antara 1 sampai 100. coba tebak')

jawab = 0
count = 1
while jawab != quiz:
    jawab = input('Masukkan tebakan ke-{}:'.format(count))
    jawab = int(jawab)
    if jawab == quiz:
        print('Ya. Anda Benar')
    elif jawab < quiz:
        print('Itu terlalu kecil. Coba lagi')
    else:
        print('Itu terlalu besar. Coba lagi')
    count += 1

>>>
```

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_12.py =====
Saya menyimpan angka bulat antara 1 sampai 100. coba tebak
Masukkan tebakan ke-1:>50
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-2:>75
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-3:>58
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-4:>60
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-5:>65
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-6:>70
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-7:>69
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-8:>67
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-9:>66
Ya. Anda Benar
>>>
```

## Soal 13

The screenshot shows a code editor on the left containing a Python script named 'soal\_13.py' and a Python shell window on the right. The script defines a function 'katakan' that converts integers into their Indonesian word representations. The shell demonstrates how to use this function for various numbers.

```
#DoniWahyuSaputro
#L200200169

def katakan(angka):
    di_bawah20 = ['Nol', 'Satu', 'Dua', 'Tiga', 'Empat', 'Lima', 'Enam', 'Tujuh',
                 'Sembilan', 'Sepuluh', 'Sebelas', 'Dua belas', 'Tiga belas',
                 'Enam belas', 'Tujuh belas', 'Delapan belas', 'Sembilan belas']
    puluhan = ['Dua puluh', 'Tiga puluh', 'Empat puluh', 'Lima puluh', 'Enam puluh',
               'Delapan puluh', 'Sembilan puluh']

    di_atas100 = {
        100 : 'ratus',
        1000 : 'ribu',
        1000000 : 'juta',
    }

    assert angka >= 0
    assert angka < 1000000000

    if angka < 20:
        return di_bawah20[angka]

    if angka < 100:
        return puluhan[(int)(angka/10)-2] + ('' if angka % 10==0 else '') + di_bawah20[angka % 10]

    pivot = max([key for key in di_atas100.keys() if key <= angka])

    #Recursion
    hasil = katakan((int)(angka/pivot)) + ' ' + di_atas100[pivot] + ('' if angka % pivot==0 else '') + katakan(angka % pivot)

    if 'Satu ratus' in hasil:
        hasil = hasil.replace('Satu ', 'Se')

    return hasil.capitalize()
```

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_13.py =====
>>> katakan(5901234)
'Lima juta sembilan ratus satu ribu dua ratus tiga puluh empat'
>>>
```

## Soal 14

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal_14.py =====
>>> formatRupiah(20000)
'Rp 20.000'
>>> formatRupiah(4500000)
'Rp 4.500.000'
>>>
```

soal\_14.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 1/soal\_14.py (3.8.5)

File Edit Format Run Options Window Help

```
#DoniWahyuSaputro
#L200200169

def formatRupiah(n):
    x = '{:,}'.format(n).replace(',', '.')
    return 'Rp ' + x
```

## MODUL 2

### SOAL

1.

The screenshot shows a Python IDE interface with two main windows. The left window is titled "saol1.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/saol1.py (3.8.5)" and contains the following Python code:

```
#DoniWahyuSaputro
#L200200169

class Pesan(object):
    def __init__(self, sebuahString):
        self.teks = sebuahString
    def cetakIni(self):
        print(self.teks)
    def cetakPakaiHurufKapital(self):
        print(str.upper(self.teks))
    def cetakPakaiHurufKecil(self):
        print(str.lower(self.teks))
    def jumlahKar(self):
        return len(self.teks)
    def cetakJumlahKarakterku(self):
        print('Kaliangku mempunyai', len(self.teks), 'karakter')
    def perbarui(self, sebuahString):
        self.teks = sebuahString
    def apakahTerkandung(self, word):
        if word in self.teks:
            print(True)
        else:
            print(False)
    def hitungKonsonan(self):
        count = 0
        konsonan = 'qwrtypsdfghjkllzxcvbnm'
        word = str.lower(self.teks)
        for i in word:
            if i in konsonan:
                count += 1
        print(count)
    def hitungVokal(self):
        count = 0
        vokal = 'aieu'
        word = str.lower(self.teks)
        for i in word:
            if i in vokal:
                count += 1
        print(count)
```

The right window is titled "Python 3.8.5 Shell" and shows the output of running the script:

```
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/saol1.py =====
>>> p9 = Pesan("Indonesia adalah negeri yang indah")
>>> p9.apakahTerkandung('ege')
True
>>> p9.apakahTerkandung('eka')
False
>>> p10 = Pesan('Surakarta')
>>> p10.hitungKonsonan()
5
>>> P10.HITUNGVOKAL()
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    P10.HITUNGVOKAL()
NameError: name 'P10' is not defined
>>> P10.hitungVokal()
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    P10.hitungVokal()
NameError: name 'P10' is not defined
>>> p10.hitungVokal()
4
>>> |
```

2.

The screenshot shows a Python IDE interface with two main windows. The left window is titled "soal2.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal2.py (3.8.5)" and contains the following Python code:

```
#DoniWahyuSaputro
#L200200169

from LatOOP4 import Manusia
class Mahasiswa(Manusia):
    def __init__(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', NIM ' + str(self.NIM) +
           ' Tinggal di ' + self.kotaTinggal +
           ' Uang saku Rp ' + str(self.uangSaku) +
           ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.uangSaku
    def makan(self, s):
        print("Saya baru saja makan", s, "sambil belajar")
        self.keadaan = 'kenyang'
    def perbaruiKotaTinggal(self, changeword):
        self.kotaTinggal = changeword
    def ambilUangSaku(self):
        return self.uangSaku
    def tambahUangSaku(self, upmoney):
        self.uangSaku = self.uangSaku + upmoney
```

The right window is titled "Python 3.8.5 Shell" and shows the output of running the script:

```
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal2.py =====
Salaam, namaku Fatimah
>>> m9 = Mahasiswa("Doni",169,'Solo',50000)
>>> m9.ambilKotaTinggal()
'Solo'
>>> m9.perbaruiKotaTinggal('Karanganyar')
>>> m9.ambilKotaTinggal()
'Karanganyar'
>>> m9.ambilUangSaku()
50000
>>> m9.tambahUangSaku(90000)
>>> m9.ambilUangSaku()
140000
>>> |
```

### 3.

The screenshot shows two windows side-by-side. The left window is a Python 3.8.5 Shell with the following content:

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal3.py =====
Salaam, namaku Fatimah
Program Input Data Mahasiswa
Masukkan Nama : Doni Wahyu Saputro
Masukkan NIM : L200200169
Masukkan Kota : Solo
Masukkan Uang Saku : 50000

Anda telah memasukkan data sebagai berikut :
Doni Wahyu Saputro, NIM L200200169. Tinggal di Solo. Uang saku Rp 50000 tiap bulannya.
>>>
```

The right window is titled "soal3.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal3.py (3.8.5)" and contains the following code:

```
#DoniWahyuSaputro
#L200200169

from soal2 import Mahasiswa

print("Program Input Data Mahasiswa")
nama = input('Masukkan Nama : ')
nim = str(input('Masukkan NIM : '))
kota = input('Masukkan Kota : ')
uang = int(input('Masukkan Uang Saku : '))

mhs = Mahasiswa(nama,nim,kota,uang)
print('\nAnda telah memasukkan data sebagai berikut : \n')
print(mhs)
```

### 4.

The screenshot shows two windows side-by-side. The left window is titled "soal4.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal4.py (3.8.5)" and contains the following code:

```
#DoniWahyuSaputro
#L200200169
from LatOOP4 import Manusia
class Mahasiswa(Manusia):
    listKuliah = []
    def __init__(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', NIM ' + str(self.NIM) \
            + '. Tinggal di ' + self.kotaTinggal \
            + '. Uang saku Rp ' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.uangSaku
    def makan(self, s):
        print("Saya baru saja makan", s, "sambil belajar")
        self.keadaan = 'kenyang'
    def ambilKotaTinggal(self):
        return self.kotaTinggal
    def perbaruiKotaTinggal(self, changeword):
        self.kotaTinggal = changeword
    def ambilUangSaku(self):
        return self.uangSaku
    def tambahUangSaku(self, upmoney):
        self.uangSaku = self.uangSaku + upmoney
    def ambilKuliah(self, choice):
        self.listKuliah.append(choice)

m234 = Mahasiswa('Doni',1169,'Solo',900000)
```

The right window is a Python 3.8.5 Shell with the following content:

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal4.py =====
Salaam, namaku Fatimah
>>> m234.listKuliah
[]
>>> m234.ambilKuliah('matematika Diskrit')
>>> m234.listKuliah
['matematika Diskrit']
>>> m234.ambilKuliah('Algoritma dan Struktur Data')
>>> m234.listKuliah
['matematika Diskrit', 'Algoritma dan Struktur Data']
>>> |
```

## 5.

The screenshot shows two windows side-by-side. The left window is a code editor titled 'soal5.py' with the following content:

```
#soal5.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal5.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

from Lat00P4 import Manusia
class Mahasiswa(Manusia):
    listKuliah = []
    def __init__(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', NIM ' + str(self.NIM) \
            + '. Tinggal di ' + self.kotaTinggal \
            + '. Uang saku Rp ' + str(self.uangSaku) \
            + ' tiap bulannya.'
        return s
    def ambilNama(self):
        return self.nama
    def ambilNIM(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.uangSaku
    def makan(self, s):
        print("Saya baru saja makan", s, "sambil belajar")
        self.keadaan = 'kenyang'
    def ambilKotaTinggal(self):
        return self.kotaTinggal
    def perbaruiKotaTinggal(self, changeword):
        self.kotaTinggal = changeword
    def ambilUangSaku(self):
        return self.uangSaku
    def tambahUangSaku(self, upmoney):
        self.uangSaku = self.uangSaku + upmoney
    def ambilKuliah(self, choice):
        self.listKuliah.append(choice)
    def hapusKuliah(self, choice):
        if choice not in self.listKuliah:
            print('Tidak ada mata kuliah {} di khs anda'.format(choice))
        else:
            self.listKuliah.remove(choice)
```

The right window is a Python shell titled 'Python 3.8.5 Shell' with the following session:

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal5.py ======
Salaam, namaku Fatimah
>>> m234 = Mahasiswa('Doni',169,'solo',50000)
>>> m234.listKuliah
[]
>>> m234.ambilKuliah('Matematika Diskrit')
>>> m234.listKuliah
['Matematika Diskrit']
>>> m234.ambilKuliah('Algoritma dan Struktur Data')
>>> m234.listKuliah
['Matematika Diskrit', 'Algoritma dan Struktur Data']
>>> m234.hapusKuliah('Kamu')
Tidak ada mata kuliah Kamu di khs anda
>>> m234.hapusKuliah('Matematika Diskrit')
>>> m234.listKuliah
['Algoritma dan Struktur Data']
>>>
```

## 6.

The screenshot shows two windows side-by-side. The left window is a code editor titled 'soal6.py' with the following content:

```
#soal6.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal6.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

from Lat00P4 import Manusia
class SiswaSMA(Manusia):
    def __init__(self, nama, kota, jurusan):
        Manusia.__init__(self, nama)
        self.kota = kota
        self.jurusan = jurusan
    def __str__(self):
        point = 'Nama : {}, Tempat tinggal di {}. Masuk Jurusan {}'.format(self.nama, self.kota, self.jurusan)
        return point
    def showNama(self):
        return self.nama
    def showKota(self):
        return self.kota
    def showJurusan(self):
        return self.jurusan
    def makanSiang(self, eating):
        print('Siang ini {} sedang kelaparan lalu dia makan {}'.format(self.nama, eating))
        self.keadaan = 'kenyang'
    def jamPelajaran(self, study):
        print('Saat ini {} sedang belajar {}'.format(self.nama, study))
        self.keadaan = 'lapar'
```

The right window is a Python shell titled 'Python 3.8.5 Shell' with the following session:

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal6.py ======
Salaam, namaku Fatimah
>>> sma = SiswaSMA('Ngatiyem','Stagen','IPA')
>>> print(sma)
Nama : Ngatiyem, Tempat tinggal di Stagen. Masuk Jurusan IPA
>>> sma.makanSiang('Ramen')
Siang ini Ngatiyem sedang kelaparan lalu dia makan Ramen
>>> sma.jamPelajaran('Psikologi')
Saat ini Ngatiyem sedang belajar Psikologi
>>> sma.keadaan
'kenyang'
>>> sma.makanSiang('Mie Goreng')
Siang ini Ngatiyem sedang kelaparan lalu dia makan Mie Goreng
>>> sma.keadaan
'kenyang'
>>>
```

## 7.

The screenshot shows a window titled "soal7.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 2/soal7.py (3.8.5)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor displays the following Python script:

```
#DoniWahyuSaputro
#L200200169

from mhstif import MhsTIF

a = MhsTIF('Doni', 169, 'Karanganyar', 500000)

a.ambilKotaTinggal() #Berasal dari class Mahasiswa
a.ambilKuliah('Algopro') #Berasal dari class Mahasiswa
a.ambilNama() #Berasal dari class Mahasiswa
a.ambilNIM() #Berasal dari class Mahasiswa
a.ambilUangSaku() #Berasal dari class Mahasiswa
a.hapusKuliah('Algopro') #Berasal dari class Mahasiswa
a.katakanPy() #Berasal dari class MhsTIF
a.keadaan #Berasal dari class Manusia
a.kota #Berasal dari class Mahasiswa
a.listKuliah #Berasal dari class Mahasiswa
a.makan() #Berasal dari class Manusia
a.mengalikanDenganDua() #Berasal dari class Manusia
a.nama #Berasal dari class Mahasiswa
a.NIM #Berasal dari class Mahasiswa
a.olahraga('Push Up') #Berasal dari class Manusia
a.perbaruiKotaTinggal('Jakarta') #Berasal dari class Mahasiswa
a.tambahUangSaku(2131233) #Berasal dari class Mahasiswa
a.uangSaku #Berasal dari class Mahasiswa
a.ucapkanSalam() #Berasal dari class Manusia
```

The status bar at the bottom right indicates "Ln: 27 Col: 0".

# **MODUL 3**

## **SOAL**

### **1. Kode Program**

```
nomor1.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 3/nomor1.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

A = [[1,2],[3,4],[5,6]]
B = [[7,8],[9,10]]
C = [[3,6],[5,2]]

#Nomor 1A
class matriks (object):
    def cetakmatriks(self, matriks):
        for i in matriks:
            print(i)
    def cekkonsisten(self, matriks):
        if len(matriks[0]) == len(matriks):
            print ("matriks konsisten")
        else:
            print ("matriks tidak konsisten")

x = matriks()
x.cetakmatriks(A)
print(x.cekkonsisten(A))

y = matriks()
y.cetakmatriks(B)
print(y.cekkonsisten(B))

#Nomor 1B
def ordo(matriks):
    return ("Ordo matriks =" +str(len(matriks))+" x "+str(len(matriks[0])))

#Nomor 1C
def Jumlah(matriks1, matriks2):
    if ordo(matriks1) == ordo(matriks2):
        for x in range(0, len(matriks1)):
            for y in range(0, len(matriks1[0])):
                print (matriks1[x][y] + matriks2[x][y], ' '),
                print()
    else:
        print("Matriks tidak sesuai")

#Nomor 1D
def kali(m,n):
    a = 0
    x,y = 0,0
    for i in range(len(m)):
        x += 1
        y = len(m[i])
```

```

v,w = 0,0
for i in range(len(n)):
    v += 1
    w = len(n[i])

if (y == v):
    print ("Bisa Dikalikan")
    vwxy = [[0 for j in range(w)] for i in range(x)]
    for i in range(len(m)):
        for j in range(len(n[0])):
            for k in range(len(n)):
                vwxy[i][j] += m[i][k] * n[k][j]
    print (vwxy)
else:
    print("Tidak memenuhi syarat")

kali(A,B)
kali(B,C)

#Nomor 1E
def determinan(p, total = 0):
    x = len(p[0])
    z = 0
    for i in range(len(p)):
        if (len(p[i]) == x):
            z += 1
    if (z == len(p)):
        if (x == len(p)):
            indices = list(range(len(p)))
            if len(p) == 2 and len(p[0]) == 2:
                val = p[0][0] * p[1][1] - p[1][0] * p[0][1]
                return val
            for fc in indices:
                pq = p
                pq = pq[1:]
                height = len(pq)
                for i in range(height):
                    pq[i] = pq[i][0:fc] + pq[i][fc+1:]
                sign = (-1) ** (fc % 2)
                sub_det = determinanHitung(pq)
                total += sign * A[0][fc] * sub_det
        else:
            return "Tidak bisa dihitung, bukan matriks bujur sangkar"
    else:
        return "Tidak bisa dihitung, bukan matriks bujur sangkar"
return total

```

## Output

The screenshot shows a Python 3.8.5 Shell window with the following content:

```

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
[1, 2]
[3, 4]
[5, 6]
matriks tidak konsisten
None
[7, 8]
[9, 10]
matriks konsisten
None
Bisa Dikalikan
[[25, 28], [57, 64], [89, 100]]
Bisa Dikalikan
[[61, 58], [77, 74]]
>>> ordo(A)
'Ordo matriks = 3 x 2'
>>> ordo(B)
>>> jumlah(A, B)
Traceback (most recent call last):
  File "<ghashell#1>", line 1, in <module>
    ordo(b)
NameError: name 'b' is not defined
>>> ordo(B)
'Ordo matriks = 2 x 2'
>>> ordo(C)
'Ordo matriks = 2 x 2'
>>> jumlah(A, B)
Matriks tidak sesuai
>>> jumlah(B, C)
10
14
12
>>> jumlah(A, C)
Matriks tidak sesuai
>>> jumlah(A, B)
Bisa Dikalikan
[[25, 28], [57, 64], [89, 100]]
>>> kali(B, C)
Bisa Dikalikan
[[61, 58], [77, 74]]
>>> determinan(A)
'Tidak bisa dihitung, bukan matriks bujur sangkar'
>>> determinan(B)
2
>>> determinan(C)
-24

```

The shell shows various matrix operations like multiplication, addition, and determinant calculation, along with error messages for non-square matrices and incompatible dimensions.

## 2. Kode Program dan Output

```

nomor2.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 3/nomor2.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyusaputro
$L200200169

#Nomor 2A
def buatMatriks(n, m=None):
    if (m == None):
        n = m
    print ("Membuat matriks 0 dengan ordo "+str(n)+" x "+str(m))
    print ([[0 for j in range(m)] for i in range(n)])

#Nomor 2B
def buatIdentitas(m):
    n = m
    print("Membuat matriks identitas dengan ordo "+str(n)+" x "+str(n))
    matriks = [[1 if j == i else 0 for j in range(m)] for i in range(n)]
    print(matriks)

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 3/nomor2.py ======
>>> buatMatriks(2,4)
Membuat matriks 0 dengan ordo 2 x 4
[[0, 0, 0, 0], [0, 0, 0, 0]]
>>> buatMatriks(3)
Membuat matriks 0 dengan ordo 3 x 3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>> buatIdentitas(4)
Membuat matriks identitas dengan ordo 4 x 4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
>>> buatIdentitas(2)
Membuat matriks identitas dengan ordo 2 x 2
[[1, 0], [0, 1]]
>>>

```

### 3.Kode Program

```

nomor3.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 3/nomor3.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyusaputro
$L200200169

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

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

    def tambahDepan(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node

    def tambahAkhir(self, data):
        if (self.head == None):
            self.head = Node(data)
        else:
            current = self.head
            while (current.next != None):
                current = current.next
            current.next = Node(data)

    def tambah(self, data, pos):
        node = Node(data)
        if not self.head:
            self.head = node
        elif pos == 0:
            node.next = self.head
            self.head = node
        else:
            prev = None
            current = self.head
            current_pos = 0
            while (current_pos < pos) and current.next:
                prev = current
                current = current.next
                current_pos += 1
            prev.next = node
            node.next = current

    def hapus(self, posisi):
        if self.head == None:
            return
        temp = self.head
        if posisi == 0:
            self.head = temp.next

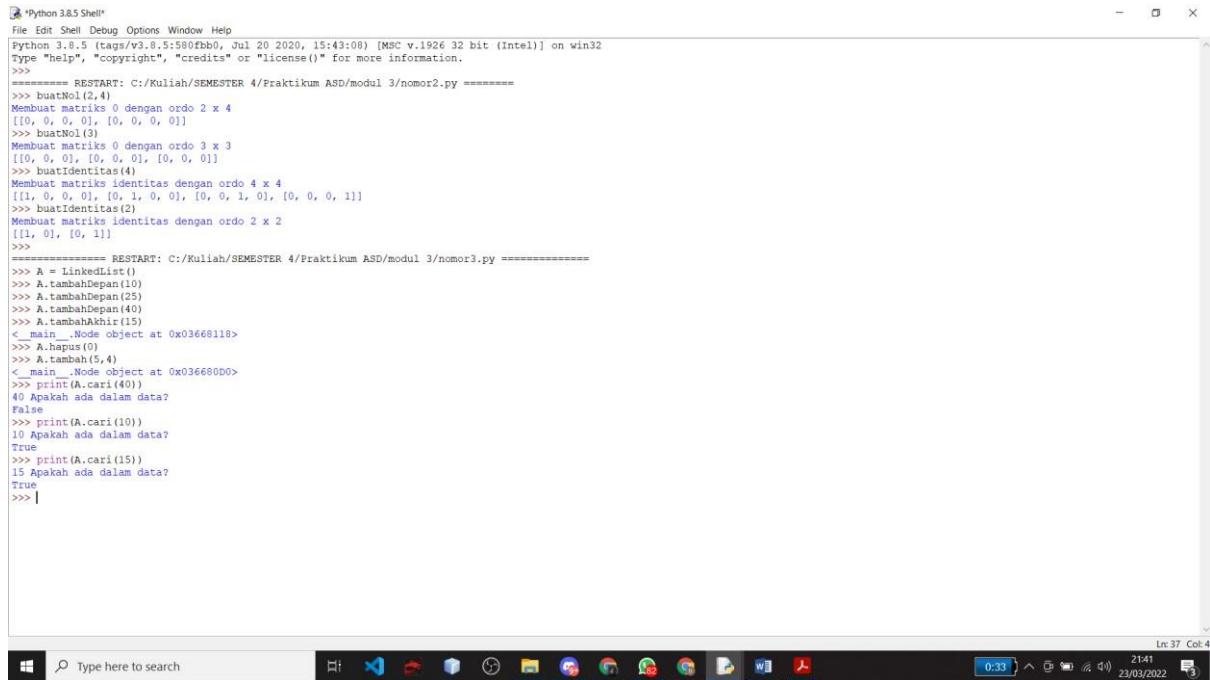
```

```

        temp = None
    return
for i in range(posisi - 1):
    temp = temp.next
    if temp is None:
        break
if temp is None:
    return
if temp.next is None:
    return
next = temp.next.next
temp.next = None
temp.next = next
def cari(self,x):
    current = self.head
    while current != None:
        if current.data == x:
            print(x, "Apakah ada dalam data?")
            return True
        current = current.next
    print(x,"Apakah ada dalam data?")
    return False
def display(self):
    current = self.head
    while current is not None:
        print(current.data, end = ' ')
        current = current.next

```

## Output



```

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fcb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 3/nomor2.py ======
>>> buatMol(2,4)
Membuat matriks 0 dengan ordo 2 x 4
[[0, 0, 0, 0], [0, 0, 0, 0]]
>>> buatMol(3)
Membuat matriks 0 dengan ordo 3 x 3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>> buatIdentitas(4)
Membuat matriks identitas dengan ordo 4 x 4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
>>> buatIdentitas(2)
Membuat matriks identitas dengan ordo 2 x 2
[[1, 0], [0, 1]]
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 3/nomor3.py ======
>>> A = LinkedList()
>>> A.tambahDepan(10)
>>> A.tambahDepan(25)
>>> A.tambahDepan(40)
>>> A.tambahAkhir(15)
<__main__._Node object at 0x03668118>
>>> A.hapus(0)
>>> A.tambah(5,4)
<__main__._Node object at 0x036680D0>
>>> print(A.cari(40))
40 Apakah ada dalam data?
False
>>> print(A.cari(10))
10 Apakah ada dalam data?
True
>>> print(A.cari(15))
15 Apakah ada dalam data?
True
>>>

```

## 4. Kode Program dan Output

```
nomor4.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 3\nomor4.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyakaputro
#L200200169

class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
class DoublyLinkedList:
    def __init__(self):
        self.head = None
    def addFirst(self, new_data):
        print("Menambah pada awal ",new_data)
        new_node = Node(new_data)
        new_node.next = self.head
        if self.head is not None:
            self.head.prev = new_node
        self.head = new_node
    def akhir(self,new_data):
        print("Menambah pada akhir ",new_data)
        new_node = Node(new_data)
        new_node.next = None
        if self.head is None:
            new_node.prev = None
            self.head = new_node
            return
        last = self.head
        while(last.next is not None):
            last = last.next
        last.next = new_node
        new_node.prev = last
    def printList(self,node):
        print("\nDari depan :")
        while (node is not None):
            print (" * "+str(node.data))
            node = node.next
        print ("\nDari belakang :")
        while (last is not None):
            print (" * "+str(last.data))
            last = last.prev
        print()

Python 3.8.5 (tags/v3.8.5:580fbdb, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 3/nomor4.py =====
>>> d = DoublyLinkedList()
>>> d.addFirst(4)
Menambah pada awal 4
>>> d.addFirst(9)
Menambah pada awal 9
>>> d.akhir(15)
Menambah pada akhir 15
>>> d.akhir(7)
Menambah pada akhir 7
>>> d.printList(d.head)

Dari depan :
9
4
15
7

Dari belakang :
7
15
4
9
>>>
```

# MODUL 4

## SOAL

### Source Code:

```
Tugas.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 4/Tugas.py (3.8.5)
File Edit Format Run Options Window Help
##Doni Wahyu Saputro
##120020016
class Mahasiswa(object):
    """Mahasiswa yang dibangun dari class Manusia."""
    def __init__(self, nama, NIM, kota, us):
        """Metode inisiasi ini menutupi metode inisiasi di class Manusia"""
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us

c0 = Mahasiswa('Ika',10,'Sukoharjo',240000)
c1 = Mahasiswa('Budi',51,'Sragen',230000)
c2 = Mahasiswa('Andi',12,'Malang',250000)
c3 = Mahasiswa('Chandra',18,'Surakarta',235000)
c4 = Mahasiswa('Eka',4,'Boyolali',240000)
c5 = Mahasiswa('Fandi',31,'Salatiga',250000)
c6 = Mahasiswa('Deni',13,'Klaten',245000)
c7 = Mahasiswa('Galuh',5,'Wonogiri',245000)
c8 = Mahasiswa('Janto',23,'Klaten',245000)
c9 = Mahasiswa('Hasan',64,'Karanganyar',270000)
c10 = Mahasiswa('Khalid',29,'Purwodadi',265000)

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

target = 'Klaten'
def cari(target, Daftar):
    o = []
    o = 0
    for i in range (len(Daftar)):
        if Daftar[i].kotaTinggal == target:
            o.append(i)
            o += 1
        else:
            o += 1
    return o

print ("-----NO. 1-----")
print(cari(target, Daftar))

#-----NO. 2-----
def cariUangSakuTerkecil(kumpulan):
    terkecil = kumpulan[0].uangSaku
    for i in kumpulan:
        if i.uangSaku < terkecil:
            terkecil = i.uangSaku
    return terkecil #kembalikan yang terkecil

Ln: 169 Col: 0
02/04/2022 02:21
```

```
Tugas.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 4/Tugas.py (3.8.5)
File Edit Format Run Options Window Help
print ("-----NO. 2-----")
print(cariUangSakuTerkecil(Daftar))

#-----NO. 3-----
def cariYangTerkecil(kumpulan):
    n = []
    terkecil = kumpulan[0].uangSaku
    for i in kumpulan:
        if i.uangSaku < terkecil:
            terkecil = i.uangSaku
            n.append(kumpulan.index(i))
    return n

print ("-----NO. 3-----")
print(cariYangTerkecil(Daftar))

#-----NO. 4-----
def cariKurangDari(kumpulan):
    b = []
    for i in kumpulan:
        if i.uangSaku < 250000:
            terkecil = i.uangSaku
            b.append(kumpulan.index(i))
    return b

print ("-----NO. 4-----")
print(cariKurangDari(Daftar))

#-----NO. 5-----
class node (object):
    def __init__(self, data, next = None):
        self.data = data
        self.next = next
    def cari (self, cari):
        curNode = self
        while curNode is not None :
            if curNode.next != None :
                if curNode.data != cari :
                    curNode = curNode.next
                else:
                    print ("Item", cari, "ada dalam Linked List")
                    break
            elif curNode.next == None :
                print ("Item", cari, "tidak ada Linked list")
                break
a = node (12)
menu = a
while menu != None :
```

```
Tugaspy - C/Kuliah/SEMESTER 4/PRAKTIKUM ASD/modul 4/Tugaspy (3.8.5)
File Edit Format Run Options Window Help
a.next = node (34)
a = a.next
a.next = node (10)
a = a.next
a.next = node (45)

print ("n-----NO. 5-----")
menu.cari(10)
menu.cari(110)

#-----NO. 6-----
def binse(kumpulan, target):
    low = 0
    high = len(kumpulan) -1
    data = []
    #Secara berulang belah runtutan itu menjadi separuhnya
    #sampai targetnya ditemukan
    while low <= high:
        #Temukan pertengahan runtut itu
        mid = (high + low) //2
        #Apakah pertengahannya memuat target?
        if kumpulan[mid] == target:
            data.append(kumpulan.index(target))
            return True
        #ataukah targetnya di sebelah kirinya?
        elif target < kumpulan[mid]:
            high = mid -1
        #ataukah targetnya di sebelah kanannya?
        else :
            low = mid +1
        #Jika runtutnya tidak bisa dibelah lagi, berarti targetnya tidak ada
    return False

a = [2,3,5,6,8,9,10,11,12,13,14]
index_a = 12
index_b = 17

print ("n-----NO. 6-----")
print ("Index : ", index_a)
print (binse(a, index_a))

print ("\nIndex : ", index_b)
print (binse(a, index_b))

##-----NO. 7-----
def binSearch(kumpulan, target):
    #mulai dari seluruh runtutan elemen
    low = 0
    high = len(kumpulan) -1
    data = []
    #Secara berulang belah runtutan itu menjadi separuhnya
    #sampai targetnya ditemukan
    while low != high:
        #Temukan pertengahan runtut itu
        mid = (high + low) //2
        #Apakah pertengahannya memuat target?
        if kumpulan[mid] == target:
            data.append(mid)
            return True
        #ataukah targetnya di sebelah kirinya?
        elif target < kumpulan[mid]:
            high = mid -1
        #ataukah targetnya di sebelah kanannya?
        else :
            low = mid +1
        for i in range (low, high):
            if target == kumpulan[i]:
                data.append(i)
    return data

List = [2, 3, 5, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
elemen = 6
|]

print ("n-----NO. 7-----")
print ("Indeks lokasi elemen ", elemen, "yang terdapat pada list ", List, "yaitu ")
print (binSearch(List, elemen))

##-----NO. 8-----
print("'''n-----NO. 8-----")
\Untuk membuat permainan tebak angka, kalau angka yang ditebak di antara 1 dan 100 maksimal jumlah tebakan adalah 7.
\Kalau antara 1 dan 1000 maksimal jumlah tebakan adalah 10.
\Nah ini terjadi karena jumlah tebakan nya bila dipangkatkan 2 tidak boleh lebih dari 100 atau 1000.
\Npolo yang digunakan pada tebakan adalah 2^n.
'''")
```

```
Tugaspy - C/Kuliah/SEMESTER 4/PRAKTIKUM ASD/modul 4/Tugaspy (3.8.5)
File Edit Format Run Options Window Help
#-----NO. 8-----
print("'''n-----NO. 8-----")
\Untuk membuat permainan tebak angka, kalau angka yang ditebak di antara 1 dan 100 maksimal jumlah tebakan adalah 7.
\Kalau antara 1 dan 1000 maksimal jumlah tebakan adalah 10.
\Nah ini terjadi karena jumlah tebakan nya bila dipangkatkan 2 tidak boleh lebih dari 100 atau 1000.
\Npolo yang digunakan pada tebakan adalah 2^n.
'''")
```

## Output

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fcb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 4/Tugas.py =====
-----NO. 1-----
(6, 8)
-----NO. 2-----
230000
-----NO. 3-----
[1]
-----NO. 4-----
[0, 1, 3, 4, 6, 7, 8]
-----NO. 5-----
Item 10 ada dalam Linked List
Item 110 tidak ada Linked list
-----NO. 6-----
Index : 12
True
Index : 17
False
-----NO. 7-----
Indeks lokasi elemen 6 yang terdapat pada list [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14] yaitu
[3, 4, 5]
-----NO. 8-----
Untuk membuat permainan tebak angka, kalau angka yang ditebak di antara 1 dan 100 maksimal jumlah tebakan adalah 7.
Kalau antara 1 dan 1000 maksimal jumlah tebakan adalah 10.
Hal ini terjadi karena jumlah tebakan nya bila dipangkatkan 2 tidak boleh lebih dari 100 atau 1000.
Pola yang digunakan pada tebakan adalah 2^n.
>>>
```

Ln: 43 Col: 4

## MODUL 5

### TUGAS

1.

tugas1.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 5/tugas1.py (3.8.5)

```
#DoniWahyusaputro
#L20200169

class MhsTF(object):
    listKuliah = []
    def __init__(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangsaku = us

c0 = MhsTF('Ika',153,'Sukoharjo', 240000)
c1 = MhsTF('Eka',122,'Surakarta', 230000)
c2 = MhsTF('Ali',22,'Surakarta', 250000)
c3 = MhsTF('Caca',180,'Surakarta', 235000)
c4 = MhsTF('Eka',47,'Boyolali', 240000)
c5 = MhsTF('Kamidi',131,'Salatiga', 250000)
c6 = MhsTF('Deni',132,'Klaten', 245000)
c7 = MhsTF('Ngatiye',50,'Wonogiri', 245000)
c8 = MhsTF('Sumarto',23,'Klaten', 245000)
c9 = MhsTF('Hamid',64,'Karanganyar', 270000)
c10 = MhsTF('Yetno',70,'Purwodadi', 265000)

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

#Nomor 1
def swap(A, p, q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp

def sortNIM(daftar):
    n = len(daftar)
    for i in range(n-1):
        for j in range(n-i-1):
            if daftar[j].NIM > daftar[j+1].NIM:
                swap(daftar, j, j+1)

def checkNIM(a):
    n = len(a)
    for i in a:
        print('NIM : {} \nNama : {} \nKota Tinggal : {}'.format(i.NIM, i.nama, i.kotaTinggal))

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 5/tugas1.py ======
>>> sortNIM(Daftar)
>>> checkNIM(Daftar)
NIM : 153
Nama : Ika
Kota Tinggal : Surakarta

NIM : 23
Nama : Sumarto
Kota Tinggal : Klaten

NIM : 47
Nama : Eka
Kota Tinggal : Boyolali

NIM : 50
Nama : Ngatiye
Kota Tinggal : Wonogiri

NIM : 64
Nama : Hamid
Kota Tinggal : Karanganyar

NIM : 70
Nama : Yetno
Kota Tinggal : Purwodadi

NIM : 120
Nama : Budi
Kota Tinggal : Slragen

NIM : 131
Nama : Kamidi
Kota Tinggal : Salatiga

NIM : 132
Nama : Deni
Kota Tinggal : Klaten

NIM : 153
Nama : Ika
Kota Tinggal : Sukoharjo

NIM : 180
Nama : Caca
Kota Tinggal : Surakarta
```

2.

tugas2.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 5/tugas2.py (3.8.5)

```
#DoniWahyusaputro
#L20200169

A = [2,9,78,65,5,70]
B = [1,4,12,43,22,11,120]

def sortTOC(a, b):
    c = a+b
    for i in range(1, len(c)):
        nilai = c[i]
        pos = i
        while pos > 0 and nilai < c[pos - 1]:
            c[pos] = c[pos-1]
            pos -=1
        c[pos] = nilai
    return c

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 5/tugas2.py ======
>>> C = sortTOC(A, B)
>>> C
[1, 2, 4, 5, 9, 11, 12, 22, 43, 65, 70, 78, 120]
```

### 3.

```
*tugas3.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 5/tugas3.py (3.8.5)*
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

from time import time as detak
from random import shuffle as acak

def swap(A, p, q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp

def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1, sampaiSini):
        if A[i] < A[posisiYangTerkecil]:
            posisiYangTerkecil = i
    return posisiYangTerkecil

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

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

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

x = [i for i in range(1, 6001)]
acak(x)
u_bub = x[:]
u_sel = x[:]
u_ins = x[:]

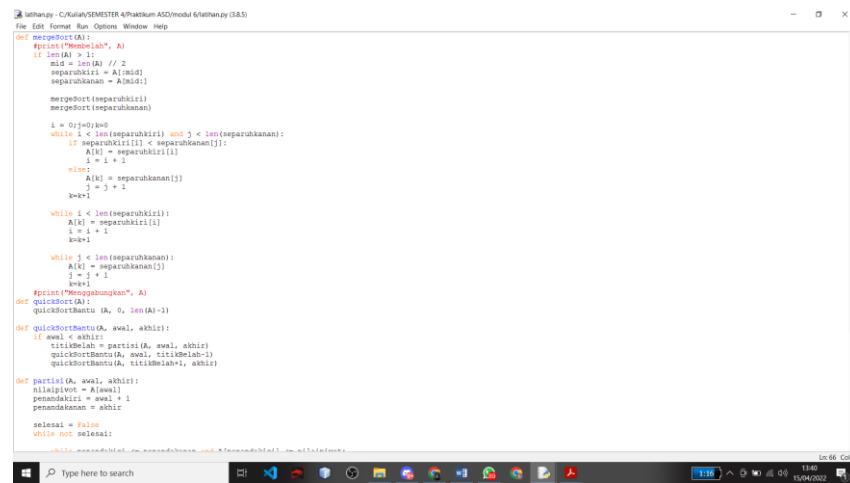
aw = detak();bubbleSort(u_bub);ak=detak();print('bubble: %g detik' %(ak-aw));
aw = detak();selectionSort(u_sel);ak=detak();print('selection: %g detik' %(ak-aw));
aw = detak();insertionSort(u_ins);ak=detak();print('insertion: %g detik' %(ak-aw));
```

### Output:

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 5/tugas3.py ======
bubble: 4.57141 detik
selection: 1.6838 detik
insertion: 2.01285 detik
>>> |
```

## MODUL 6

### Latihan.py

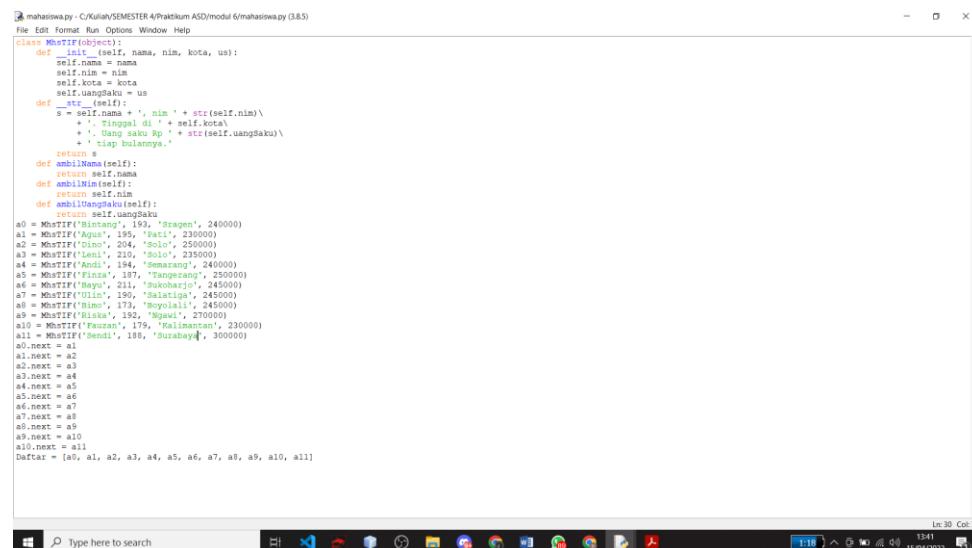


```
Latihan.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 6\latihan.py (3.8.5)
File Edit Formulas Run Options Window Help
def bubbleSort(A):
    #print("Membalik", A)
    if len(A) > 1:
        i = 0
        while i < len(A) - 1:
            j = i + 1
            separuhkiri = A[:mid]
            separuhkanan = A[mid:]
            mergeSort(separuhkiri)
            mergeSort(separuhkanan)

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

            while j < len(separuhkanan):
                A[i] = separuhkanan[j]
                j += 1
                k+=1
            #print("Setelah penghubungan", A)
        quickSort(A, 0, len(A)-1)
def quickSort(A, awl, akhir):
    if awl < akhir:
        pivot = partisi(A, awl, akhir)
        quickSortBantu(A, awl, pivot-1)
        quickSortBantu(A, pivot+1, akhir)
    else:
        selesai = True
        while not selesai:
            #print("Setelah penghubungan", A)
            quickSort(A, awl, akhir);
            if awl < akhir:
                pivot = partisi(A, awl, akhir)
                quickSortBantu(A, awl, pivot-1)
                quickSortBantu(A, pivot+1, akhir)
            else:
                selesai = True
            while not selesai:
                #print("Setelah penghubungan", A)
                quickSort(A, awl, akhir);
                if awl < akhir:
                    pivot = partisi(A, awl, akhir)
                    quickSortBantu(A, awl, pivot-1)
                    quickSortBantu(A, pivot+1, akhir)
                else:
                    selesai = True
            selesai = False
while not selesai:
    #print("Setelah penghubungan", A)
    quickSort(A, 0, len(A)-1)
def partisi(A, awl, akhir):
    nilaipivot = A[awl]
    penandakiri = awl + 1
    penandakanan = akhir
    selesai = False
    while not selesai:
        while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
            penandakiri = penandakiri + 1
        while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
            penandakanan = penandakanan - 1
        if penandakanan < penandakiri:
            selesai = True
        else:
            temp = A[penandakiri]
            A[penandakiri] = A[penandakanan]
            A[penandakanan] = temp
    temp = A[awl]
    A[awl] = A[penandakanan]
    A[penandakanan] = temp
return penandakanan
```

### Mahasiswa.py



```
mahasiswa.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 6\mahasiswa.py (3.8.5)
File Edit Formulas Run Options Window Help
class Mahasiswa(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', nim ' + str(self.nim) +
           ' tinggal di ' + self.kota +
           ', uang saku Rp ' + str(self.uangSaku) +
           ' tiap bulannya'
        return s
    def ambilImbalan(self):
        return self.uangSaku
    def tambahImbalan(self):
        return self.uangSaku
a0 = Mahasiswa('Wintang', 193, 'Stagen', 240000)
a1 = Mahasiswa('Dwi', 194, 'Surabaya', 250000)
a2 = Mahasiswa('Cindi', 200, 'Solo', 250000)
a3 = Mahasiswa('Lend', 210, 'Solo', 235000)
a4 = Mahasiswa('Andi', 194, 'Semarang', 240000)
a5 = Mahasiswa('Titi', 197, 'Malang', 250000)
a6 = Mahasiswa('Bayu', 211, 'Sukoharjo', 245000)
a7 = Mahasiswa('Ulin', 192, 'Salatiga', 245000)
a8 = Mahasiswa('Tina', 178, 'Magelang', 250000)
a9 = Mahasiswa('Rizka', 192, 'Ngawi', 270000)
a10 = Mahasiswa('Fauzan', 179, 'Kalinantan', 230000)
all = Mahasiswa('Sendi', 186, 'Surabaya', 300000)
a0.next = a1
a1.next = a2
a2.next = a3
a3.next = a4
a4.next = a5
a5.next = a6
a6.next = a7
a7.next = a8
a8.next = a9
a9.next = a10
a10.next = all
Daftar = [a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, all]
```

## No1

```
#DoniWahyuSaputro
#L200200169

from latihan import *
from mahasiswa import *

def convert(arr, obj):
    hasil=[]
    for x in range (len(arr)):
        for i in range (len(arr)):
            if arr[x] == obj[i].nim:
                hasil.append(obj[i])
    return hasil

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

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

urutkanQuick()
urutkanMerge()

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win ^_32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no1.py =====
Quick Sort
173
179
187
188
190
192
193
193
194
195
204
210
211
Merge Sort
173
179
187
188
190
192
193
194
195
204
210
211
>>>
```

## No2

```
if A[i] < A[posisiYangTerkecil]:
    posisiYangTerkecil = i
return posisiYangTerkecil

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

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

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

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

        mergeSort(separuhkiri)
        mergeSort(separuhkanan)

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

## No3

The screenshot displays two windows side-by-side. The left window is titled 'no3.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 6\no3.py (3.8.5)' and contains Python code for a merge sort algorithm. The right window is titled 'Python 3.8.5 Shell' and shows the output of a quicksort execution with various performance metrics.

```
File Edit Format Run Options Window Help
penandakanan = akhir
selesai = False
while not selesai:
    while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
        penandakiri = penandakiri + 1
    while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
        penandakanan = penandakanan - 1
    if penandakanan < penandakiri:
        selesai = True
    else:
        temp = A[penandakiri]
        A[penandakiri] = A[penandakanan]
        A[penandakanan] = temp

    temp = A[awal]
    A[awal] = A[penandakanan]
    A[penandakanan] = temp

return penandakanan

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

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

k = list(range(6000))
cocok(k)
u_bub = k[:]
u_sel = k[:]
u_ins = k[:]
u_mrg = k[:]
u_qck = k[:]

awdetak();bubbleSort(u_bub);ak=detak();print('bubble: %g detik' %(ak-aw));
awdetak();selectionSort(u_sel);ak=detak();print('selection: %g detik' %(ak-aw));
awdetak();insertionSort(u_ins);ak=detak();print('insertion: %g detik' %(ak-aw));
awdetak();mergeSort(u_mrg);ak=detak();print('merge: %g detik' %(ak-aw));
awdetak();quicksort(u_qck);ak=detak();print('quicksort: %g detik' %(ak-aw));
>>> ===== RESTART: C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 6\no3.py =====
bubble: 4.19728 detik
selection: 1.67987 detik
insertion: 2.07987 detik
merge: 0.0263574 detik
quicksort: 0.0162489 detik
```

## No4

The screenshot displays two windows side-by-side. The left window is titled 'latihan62.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 6\latihan62.py (3.8.5)' and contains Python code for a merge sort algorithm. The right window is titled 'latihan63.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 6\latihan63.py (3.8.5)' and contains Python code for a quicksort algorithm.

```
File Edit Format Run Options Window Help
def mergeSort(A):
    #print("Membelah ", A)
    if len(A) > 1:
        mid = len(A) // 2      # Membelah list.
        separuhKiri = A[:mid]  # Slicing ini langkah yang expensive sebenarnya,
        separuhKanan = A[mid:] # bisaakah kamu membuatnya lebih baik?

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

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

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

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

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

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

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

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

    penandaKiri = awal + 1
    penandaKanan = akhir

    selesai = False
    while not selesai:
        while penandaKiri <= penandaKanan and A[penandaKiri] <= nilaiPivot:
            penandaKiri = penandaKiri + 1

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

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

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

    return penandaKanan
```

```

no4.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no4.py (3.8.5)
File Edit Format Run Options Window Help
from latihan62 import *
from latihan63 import *
L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72]
mergeSort(L)
print(L)
quicksort(L)
print(L)

Python 3.8.5 (tags/v3.8.5:580fbdb, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no4.py =====
[2, 7, 16, 19, 24, 35, 43, 72, 80, 91]
[2, 7, 16, 19, 24, 35, 43, 72, 80, 91]
>>>

```

## No5

```

no5.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no5.py (3.8.5)
File Edit Format Run Options Window Help
#DoniMahyusaputro
#L200200159

from mahasiswa import *

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

def mergeSort2(A, awal, akhir):
    mid = (awal+akhir)//2
    if awal < akhir:
        mergeSort2(A, awal, mid)
        mergeSort2(A, mid+1, akhir)

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

    if f <= mid:
        tmp[a:] = A[f:mid+1]

    if l <= akhir:
        tmp[a:] = A[l:akhir+1]

    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

def mergeSort(A):
    mergeSort2(A, 0, len(A)-1)

print("Sebelum diurutkan")
cetak(Daftar)
mergeSort(Daftar)
print("\nSetelah diurutkan")
cetak(Daftar)

Python 3.8.5 (tags/v3.8.5:580fbdb, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no5.py =====
Sebelum diurutkan
Bintang, nim 193. Tinggal di Sragen. Uang saku Rp 240000 tiap bulannya.
Agus, nim 195. Tinggal di Pati. Uang saku Rp 230000 tiap bulannya.
Dina, nim 196. Tinggal di Solo. Uang saku Rp 250000 tiap bulannya.
Leni, nim 210. Tinggal di Solo. Uang saku Rp 235000 tiap bulannya.
Andi, nim 194. Tinggal di Semarang. Uang saku Rp 240000 tiap bulannya.
Finza, nim 187. Tinggal di Tangerang. Uang saku Rp 250000 tiap bulannya.
Bayu, nim 211. Tinggal di Sukoharjo. Uang saku Rp 245000 tiap bulannya.
Ulin, nim 190. Tinggal di Salatiga. Uang saku Rp 245000 tiap bulannya.
Bimo, nim 173. Tinggal di Boyolali. Uang saku Rp 245000 tiap bulannya.
Riska, nim 192. Tinggal di Ngawi. Uang saku Rp 270000 tiap bulannya.
Fauzan, nim 179. Tinggal di Kalimantan. Uang saku Rp 230000 tiap bulannya.
Senoi, nim 186. Tinggal di Surabaya. Uang saku Rp 300000 tiap bulannya.

Setelah diurutkan
Fauzan, nim 179. Tinggal di Kalimantan. Uang saku Rp 230000 tiap bulannya.
Agus, nim 195. Tinggal di Pati. Uang saku Rp 230000 tiap bulannya.
Leni, nim 210. Tinggal di Solo. Uang saku Rp 235000 tiap bulannya.
Andi, nim 194. Tinggal di Semarang. Uang saku Rp 240000 tiap bulannya.
Bintang, nim 193. Tinggal di Sragen. Uang saku Rp 240000 tiap bulannya.
Bimo, nim 173. Tinggal di Boyolali. Uang saku Rp 245000 tiap bulannya.
Ulin, nim 190. Tinggal di Salatiga. Uang saku Rp 245000 tiap bulannya.
Bella, nim 182. Tinggal di Sukoharjo. Uang saku Rp 245000 tiap bulannya.
Finza, nim 187. Tinggal di Tangerang. Uang saku Rp 250000 tiap bulannya.
Dino, nim 204. Tinggal di Solo. Uang saku Rp 250000 tiap bulannya.
Riska, nim 192. Tinggal di Ngawi. Uang saku Rp 270000 tiap bulannya.
Senoi, nim 186. Tinggal di Surabaya. Uang saku Rp 300000 tiap bulannya.
>>>

```

## No6

```
#DoniWahyuSaputro
#120200159

from mahasiswa import *

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

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

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

```
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no6.py
=====
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no6.py
=====
Sebelum diurutkan
Bintang, nim 193. Tinggal di Slragen. Uang saku Rp 240000 tiap bulannya.
Agus, nim 195. Tinggal di Pati. Uang saku Rp 230000 tiap bulannya.
Dino, nim 204. Tinggal di Solo. Uang saku Rp 250000 tiap bulannya.
Leni, nim 210. Tinggal di Solo. Uang saku Rp 235000 tiap bulannya.
Andi, nim 194. Tinggal di Tangerang. Uang saku Rp 245000 tiap bulannya.
Finza, nim 187. Tinggal di Tangerang. Uang saku Rp 250000 tiap bulannya.
Bayu, nim 211. Tinggal di Sukoharjo. Uang saku Rp 245000 tiap bulannya.
Ulin, nim 190. Tinggal di Salatiga. Uang saku Rp 245000 tiap bulannya.
Bimo, nim 173. Tinggal di Boyolali. Uang saku Rp 245000 tiap bulannya.
Riska, nim 192. Tinggal di Ngawi. Uang saku Rp 270000 tiap bulannya.
Fauzan, nim 179. Tinggal di Kalimantan. Uang saku Rp 230000 tiap bulannya.
Sendi, nim 188. Tinggal di Surabaya. Uang saku Rp 300000 tiap bulannya.

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

## No7

```
#DoniWahyuSaputro
#120200169

from time import time as tak
from random import shuffle as kocok
import time

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

        mergeSort(separuhkiri)
        mergeSort(separuhkanan)

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

        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i = i + 1
            k=k+1

        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j = j + 1
            k=k+1
    #print("Menggabungkan",A)

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

    penandaKiri = awal + 1
    penandaKanan = akhir

    selesai = False
    while not selesai:
```

```
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no7.py
=====
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no7.py
=====
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
merge: 0.0376332 detik
quicksort: 0.0172144 detik
mergesort: 0.036438 detik
quick New: 0.0142506 detik
merge New: 0.0376332 detik
>>>
```

```

no7.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no7.py (3.8.5)
File Edit Format Run Options Window Help
selesai = False
while not selesai:
    while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
        penandakiri = penandakiri + 1
    while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
        penandakanan = penandakanan - 1
    if penandakanan < penandakiri:
        selesai = True
    else:
        temp = A[penandakiri]
        A[penandakiri] = A[penandakanan]
        A[penandakanan] = temp
    temp = A[awal]
    A[awal] = A[penandakanan]
    A[penandakanan] = temp
return penandakanan

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

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

def mergeSort2(A, awal, akhir):
    mid = (awal+akhir)//2
    if awal < akhir:
        mergeSort2(A, awal, mid)
        mergeSort2(A, mid+1, akhir)

a, f, l = 0, awal, mid+1
tmp = [None] * (akhir - awal + 1)
while f <= mid and l <= akhir:
    if A[f] < A[l]:
        tmp[a] = A[f]
        f += 1
    else:
        tmp[a] = A[l]
        l += 1
    a += 1
a = 0
while awal <= akhir:
    A[awal] = tmp[a]
    awal += 1
    a += 1

def mergeSortNew(A):
    mergeSort2(A, 0, len(A)-1)

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

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

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

k = [[i] for i in range(1, 6001)]

```

```

no7.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no7.py (3.8.5)
File Edit Format Run Options Window Help
k = [[i] for i in range(1, 6001)]
kocok(k)
u_mrg = k[:]
u_qck = k[:]
u_mrgNew = k[:]
u_qckNew = k[:]

aw=detak();mergeSort(u_mrg);ak=detak();print("merge: %g detik" %(ak-aw));
aw=detak();quickSort(u_qck);ak=detak();print("quick: %g detik" %(ak-aw));
aw=detak();mergeSortNew(u_mrgNew);ak=detak();print("merge New: %g detik" %(ak-aw));
aw=detak();quickSortNew(u_qckNew);ak=detak();print("quick New: %g detik" %(ak-aw));

```

```

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

aw=detak();mergeSort(u_mrg);ak=detak();print("merge: %g detik" %(ak-aw));
aw=detak();quickSort(u_qck);ak=detak();print("quick: %g detik" %(ak-aw));
aw=detak();mergeSortNew(u_mrgNew);ak=detak();print("merge New: %g detik" %(ak-aw));
aw=detak();quickSortNew(u_qckNew);ak=detak();print("quick New: %g detik" %(ak-aw));

```

## No8

The screenshot shows a Windows desktop with two windows open. On the left is a Notepad++ window titled 'no8.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no8.py (3.8.5)'. It contains Python code for a linked list and merge sort. On the right is a 'Python 3.8.5 Shell' window showing the output of running the script. The shell output includes the Python version, copyright information, and the result of the merge sort function.

```
#DoniWahyaputro
#120200169

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

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

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

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

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

    if len(A) > 1:
        mid = len(A) // 2
        separuhkiri = A[:mid]
        separuhkanan = A[mid:]

        mergeSortLL(separuhkiri)
        mergeSortLL(separuhkanan)

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

        while i < len(separuhkiri):
            A[i] = separuhkiri[i]
            i = i + 1
            k=k+1

        while j < len(separuhkanan):
            A[j] = separuhkanan[j]
            j = j + 1
            k=k+1

    for x in A:
        try:
            linked.data = x
            linked = linked.tautan
        except:
            pass

mergeSortLL(a)
cetak(a)
```

```
Python 3.8.5 (tags/v3.8.5:580fbdb, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no8.py =====
1
2
3
4
5
6
7
>>>
```

The screenshot shows a Windows desktop with two windows open. On the left is a Notepad++ window titled 'no8.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no8.py (3.8.5)'. It contains the same Python code as the previous screenshot. On the right is a 'Python 3.8.5 Shell' window showing the output of running the script. The shell output includes the Python version, copyright information, and the result of the merge sort function.

```
#DoniWahyaputro
#120200169

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

def cetak(head):
    curr = head
    while curr:
        try:
            print (curr.data)
            curr = curr.tautan
        except:
            pass

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

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

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

    if len(A) > 1:
        mid = len(A) // 2
        separuhkiri = A[:mid]
        separuhkanan = A[mid:]

        mergeSortLL(separuhkiri)
        mergeSortLL(separuhkanan)

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

        while i < len(separuhkiri):
            A[i] = separuhkiri[i]
            i = i + 1
            k=k+1

        while j < len(separuhkanan):
            A[j] = separuhkanan[j]
            j = j + 1
            k=k+1

    for x in A:
        try:
            linked.data = x
            linked = linked.tautan
        except:
            pass

mergeSortLL(a)
cetak(a)
```

```
Python 3.8.5 (tags/v3.8.5:580fbdb, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 6/no8.py =====
1
2
3
4
5
6
7
>>>
```

MODUL 7

Tugas

No 1

## Source Code



```
*no1.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\modul 7\no1.py (3.8.5)*
File Edit Format Run Options Window Help
##DoniWahyuSaputro
##L200200169|
import re

a = open('Indonesia.txt','r')
baca = a.read()
a.close
cocok= re.findall(r"\sme\w+",baca)
print(cocok)
print("( Double Klik )")
```

## Output

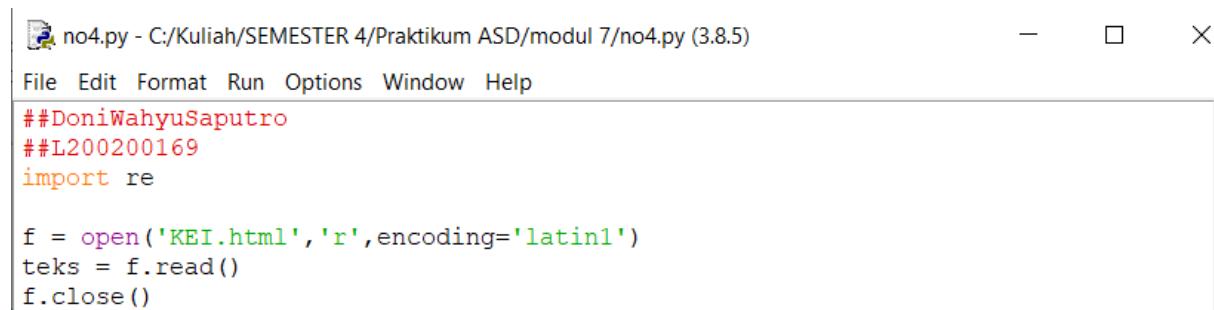
```
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 7/nol.py =====
Squeezed text (60 lines).
( Double Klik )
>>>
```

No2

No3

No4

## Source Code



```
##DoniWahyuSaputro
##L200200169
import re

f = open('KEI.html','r',encoding='latin1')
teks = f.read()
f.close()
```

## Output:

```
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 7/no4.py =====
>>> pola4 = r"(wr)/></td>"[0]
>>> num4= re.findall(pola4,teks)
>>> print(num4)
['("Denmark", "9.58"), ("Sweden", "9.52"), ("Finland", "9.37"), ('Netherlands', '9.32'), ('Norway', '9.27'), ('Canada', '9.21'), ('Switzerland', '9.15'), ('Kingdom', '9.09'), ('States', '9.08'), ('Australia', '9.05'), ('Ireland', '8.92'), ('Austria', '8.89), ('Iceland', '8.88'), ('Germany', '8.87), ('Zealand', '8.87), ('Belgium', '8.73), ('Taiwan', '8.69), ('Luxembourg', '8.65), ('Japan', '8.56), ('France', '8.47), ('Estonia', '8.34), ('Slovenia', '8.25), ('Spain', '8.24), ('Singapore', '8.24), ('Israel', '8.22), ('China', '8.20), ('Italy', '7.86), ('Hungary', '7.85), ('Romania', '7.84), ('Croatia', '7.83), ('Bulgaria', '7.82), ('Emirates', '7.81), ('Ukraine', '7.78), ('Latvia', '7.77), ('Cyprus', '7.58), ('Portugal', '7.57), ('Greece', '7.31), ('Poland', '7.38), ('Slovakia', '7.33), ('Barbados', '7.25), ('Croatia', '7.19), ('Chile', '6.92), ('Bulgaria', '6.88), ('Romania', '6.66), ('Emirates', '6.37), ('Uruguay', '6.34), ('Qatar', '6.15), ('Thailand', '7.71), ('Malta', '6.06), ('Maldives', '6.04), ('Jordan', '6.01), ('Ukraine', '6.01), ('Ukraine', '5.80), ('Lebanon', '5.49), ('Angola', '5.44), ('Macedonia', '5.37), ('Macedonia', '5.37), ('Macedonia', '5.33), ('Brazil', '5.57), ('Turkey', '5.61), ('Africa', '5.55), ('Jordan', '5.53), ('Mexico', '5.51), ('Thailand', '5.44), ('Oman', '5.37), ('Macedonia', '5.33), ('Mauritius', '5.18), ('Arabia', '5.15), ('Jamaica', '5.04), ('Kazakhstan', '5.01), ('Belarus', '4.93), ('Lebanon', '4.86), ('Panama', '4.69), ('Georgia', '4.69), ('Peru', '4.64), ('Mongolia', '4.50), ('Colombia', '4.42), ('China', '4.35), ('Guyana', '4.31), ('Philippines', '4.25), ('Venezuela', '4.23), ('Nanibia', '4.19), ('Lanka', '4.16), ('Albania', '4.04), ('Egypt', '4.03), ('Botswana', '3.96), ('Republic', '3.92), ('Salvador', '3.91), ('Azerbaijan', '3.81), ('Kyrgyzstan', '3.74), ('Paraguay', '3.62), ('Ecuador', '3.46), ('Morocco', '3.45), ('Bolivia', '3.42), ('Iran', '3.39), ('Uzbekistan', '3.28), ('Algeria', '3.25), ('Verde', '3.24), ('Indonesia', '3.23), ('Honduras', '3.21), ('India', '3.12), ('Vietnam', '3.02), ('Swaziland', '2.93), ('Republic', '2.90), ('Nicaragua', '2.87), ('Kenya', '2.82), ('Tajikistan', '2.79), ('Senegal', '2.63), ('Simbabwe', '2.51), ('Ghana', '2.46), ('Madagascar', '2.35), ('Mauritania', '2.35), ('Tanzania', '2.28), ('Pakistan', '2.24), ('Lesotho', '2.15), ('Benin', '2.10), ('Nigeria', '2.04), ('Yemen', '1.80), ('Mali', '1.78), ('Angola', '1.70), ('Cameroon', '1.69), ('Faso', '1.64), ('Nepal', '1.61), ('Malawi', '1.55), ('Laos', '1.53), ('Bangladesh', '1.49), ('Myanmar', '1.48), ('Rwanda', '1.34), ('Ethiopia', '1.18), ('Djibouti', '1.15), ('Eritrea', '1.07), ('Leone', '0.91')]
```

```
>>> tuples = [(t[0], float(t[1])) for t in tuples]
>>> print(tuples)
[('Denmark', 9.58), ('Sweden', 9.52), ('Finland', 9.37), ('Netherlands', 9.32), (9.27), ('Norway', 9.21), ('Canada', 9.15), ('Kingdom', 9.09), ('States', 9.08), ('Australia', 9.05), ('Ireland', 8.92), ('Austria', 8.89), ('Iceland', 8.88), ('Germany', 8.87), ('Zealand', 8.87), ('Belgium', 8.73), ('Taiwan', 8.69), ('Luxembourg', 8.65), ('Japan', 8.56), ('France', 8.47), ('Estonia', 8.34), ('Slovenia', 8.25), ('Spain', 8.24), ('Singapore', 8.24), ('Israel', 8.22), ('China', 8.20), ('Italy', 7.86), ('Hungary', 7.85), ('Romania', 7.84), ('Croatia', 7.83), ('Bulgaria', 7.82), ('Emirates', 7.81), ('Ukraine', 7.78), ('Latvia', 7.77), ('Cyprus', 7.58), ('Portugal', 7.57), ('Greece', 7.31), ('Poland', 7.38), ('Slovakia', 7.33), ('Barbados', 7.25), ('Croatia', 7.19), ('Chile', 6.92), ('Bulgaria', 6.88), ('Romania', 6.66), ('Emirates', 6.37), ('Uruguay', 6.34), ('Qatar', 6.15), ('Thailand', 7.71), ('Malta', 6.06), ('Maldives', 6.04), ('Jordan', 6.01), ('Ukraine', 6.01), ('Ukraine', 5.80), ('Lebanon', 5.49), ('Angola', 5.44), ('Macedonia', 5.37), ('Macedonia', 5.37), ('Macedonia', 5.33), ('Brazil', 5.57), ('Turkey', 5.61), ('Africa', 5.55), ('Jordan', 5.53), ('Mexico', 5.51), ('Thailand', 5.44), ('Oman', '5.37), ('Macedonia', '5.33), ('Mauritius', '5.18), ('Arabia', '5.15), ('Jamaica', '5.04), ('Kazakhstan', '5.01), ('Belarus', '4.93), ('Lebanon', '4.86), ('Panama', '4.69), ('Georgia', '4.69), ('Peru', '4.64), ('Mongolia', '4.50), ('Colombia', '4.42), ('China', '4.35), ('Guyana', '4.31), ('Philippines', '4.25), ('Venezuela', '4.23), ('Nanibia', '4.19), ('Lanka', '4.16), ('Albania', '4.04), ('Egypt', '4.03), ('Botswana', '3.96), ('Republic', '3.92), ('Salvador', '3.91), ('Azerbaijan', '3.81), ('Kyrgyzstan', '3.74), ('Paraguay', '3.62), ('Ecuador', '3.46), ('Morocco', '3.45), ('Bolivia', '3.42), ('Iran', '3.39), ('Uzbekistan', '3.28), ('Algeria', '3.25), ('Verde', '3.24), ('Indonesia', '3.23), ('Honduras', '3.21), ('India', '3.12), ('Vietnam', '3.02), ('Swaziland', '2.93), ('Republic', '2.90), ('Nicaragua', '2.87), ('Kenya', '2.82), ('Tajikistan', '2.79), ('Senegal', '2.63), ('Simbabwe', '2.51), ('Ghana', '2.46), ('Madagascar', '2.35), ('Mauritania', '2.35), ('Tanzania', '2.28), ('Pakistan', '2.24), ('Lesotho', '2.15), ('Benin', '2.10), ('Nigeria', '2.04), ('Yemen', '1.80), ('Mali', '1.78), ('Angola', '1.70), ('Cameroon', '1.69), ('Faso', '1.64), ('Nepal', '1.61), ('Malawi', '1.55), ('Laos', '1.53), ('Bangladesh', '1.49), ('Myanmar', '1.48), ('Rwanda', '1.34), ('Ethiopia', '1.18), ('Djibouti', '1.15), ('Eritrea', '1.07), ('Leone', '0.91')]
```

```
>>> tupp = [(t[0], float(t[1])) for t in tuples]
>>> print(tupp)
[('Denmark', 9.58), ('Sweden', 9.52), ('Finland', 9.37), ('Netherlands', 9.32), (9.27), ('Norway', 9.21), ('Canada', 9.15), ('Kingdom', 9.09), ('States', 9.08), ('Australia', 9.05), ('Ireland', 8.92), ('Austria', 8.89), ('Iceland', 8.88), ('Germany', 8.87), ('Zealand', 8.87), ('Belgium', 8.73), ('Taiwan', 8.69), ('Luxembourg', 8.65), ('Japan', 8.56), ('France', 8.47), ('Estonia', 8.34), ('Slovenia', 8.25), ('Spain', 8.24), ('Singapore', 8.24), ('Israel', 8.22), ('China', 8.20), ('Italy', 7.86), ('Hungary', 7.85), ('Romania', 7.84), ('Croatia', 7.83), ('Bulgaria', 7.82), ('Emirates', 7.81), ('Ukraine', 7.78), ('Latvia', 7.77), ('Cyprus', 7.58), ('Portugal', 7.57), ('Greece', 7.31), ('Poland', 7.38), ('Slovakia', 7.33), ('Barbados', 7.25), ('Croatia', 7.19), ('Chile', 6.92), ('Bulgaria', 6.88), ('Romania', 6.66), ('Emirates', 6.37), ('Uruguay', 6.34), ('Qatar', 6.15), ('Thailand', 7.71), ('Malta', 6.06), ('Maldives', 6.04), ('Jordan', 6.01), ('Ukraine', 6.01), ('Ukraine', 5.80), ('Lebanon', 5.49), ('Angola', 5.44), ('Macedonia', 5.37), ('Macedonia', 5.37), ('Macedonia', 5.33), ('Brazil', 5.57), ('Turkey', 5.61), ('Africa', 5.55), ('Jordan', 5.53), ('Mexico', 5.51), ('Thailand', 5.44), ('Oman', '5.37), ('Macedonia', '5.33), ('Mauritius', '5.18), ('Arabia', '5.15), ('Jamaica', '5.04), ('Kazakhstan', '5.01), ('Belarus', '4.93), ('Lebanon', '4.86), ('Georgia', '4.69), ('Peru', '4.64), ('Mongolia', '4.50), ('Colombia', '4.42), ('China', '4.35), ('Guyana', '4.31), ('Philippines', '4.25), ('Venezuela', '4.23), ('Nanibia', '4.19), ('Lanka', '4.16), ('Albania', '4.04), ('Egypt', '4.03), ('Botswana', '3.96), ('Republic', '3.92), ('Salvador', '3.91), ('Azerbaijan', '3.81), ('Kyrgyzstan', '3.74), ('Paraguay', '3.62), ('Ecuador', '3.46), ('Morocco', '3.45), ('Bolivia', '3.42), ('Iran', '3.39), ('Uzbekistan', '3.28), ('Algeria', '3.25), ('Verde', '3.24), ('Indonesia', '3.23), ('Honduras', '3.21), ('India', '3.12), ('Vietnam', '3.02), ('Swaziland', '2.93), ('Republic', '2.90), ('Nicaragua', '2.87), ('Kenya', '2.82), ('Tajikistan', '2.79), ('Senegal', '2.63), ('Simbabwe', '2.51), ('Ghana', '2.46), ('Madagascar', '2.35), ('Mauritania', '2.35), ('Tanzania', '2.28), ('Pakistan', '2.24), ('Lesotho', '2.15), ('Benin', '2.10), ('Nigeria', '2.04), ('Yemen', '1.80), ('Mali', '1.78), ('Angola', '1.70), ('Cameroon', '1.69), ('Faso', '1.64), ('Nepal', '1.61), ('Malawi', '1.55), ('Laos', '1.53), ('Bangladesh', '1.49), ('Myanmar', '1.48), ('Rwanda', '1.34), ('Ethiopia', '1.18), ('Djibouti', '1.15), ('Eritrea', '1.07), ('Leone', '0.91')]
```

## MODUL 8

### Tugas

1.

```
*no1.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no1.py (3.8.5)*
File Edit Format Run Options Window Help
#DoniMahyuSaputro
#120200169
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(), "Tidak bisa diintip. Stack kosong"
        return self.items[-1]
    def pop(self):
        assert not self.isEmpty(), "Tidak bisa dipop dari Stack kosong"
        return self.items.pop()
    def push(self, data):
        self.items.append(data)

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

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

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbbo, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no1.py =====
>>> cetakHexa(12)
12
>>> cetakHexa(31)
'31'
>>> cetakHexa(229)
'E5'
>>> cetakHexa(255)
'FF'
>>> cetakHexa(31519)
'7B1F'
>>>
```

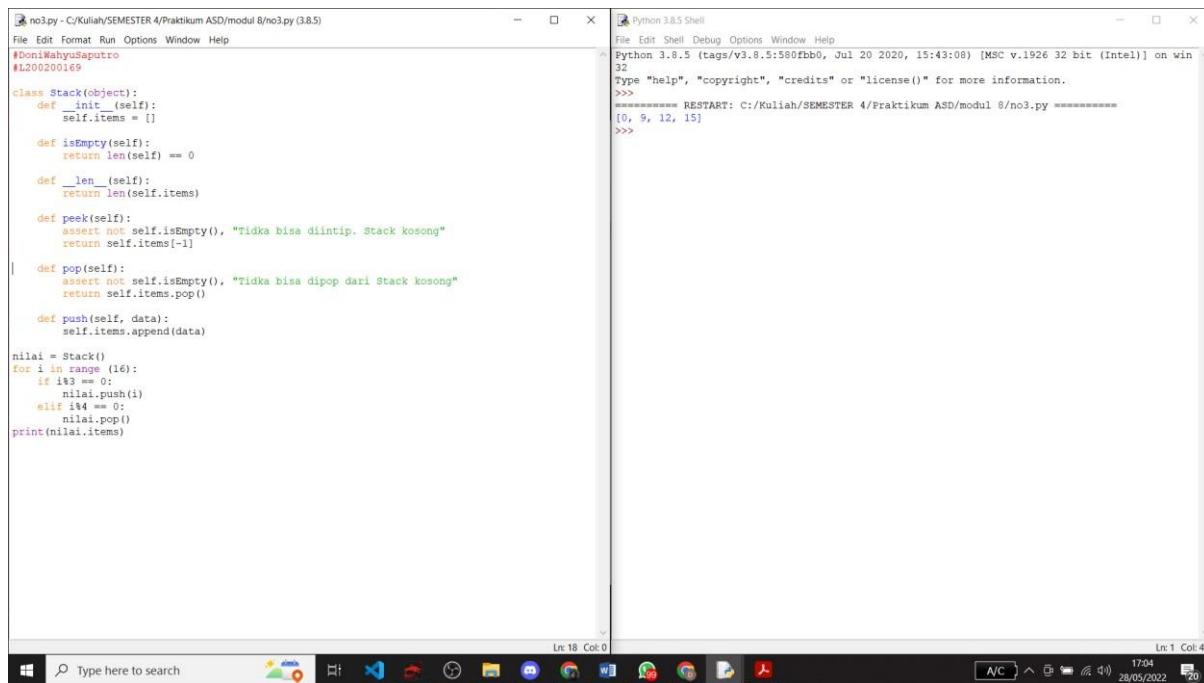
2.

```
*no2.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no2.py (3.8.5)*
File Edit Format Run Options Window Help
#DoniMahyuSaputro
#120200169
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(), "Tidak bisa diintip. Stack kosong"
        return self.items[-1]
    def pop(self):
        assert not self.isEmpty(), "Tidak bisa dipop dari Stack kosong"
        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)
print(nilai.items)

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbbo, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no2.py =====
[0, 3, 6, 9, 12, 15]
>>>
```

### 3.



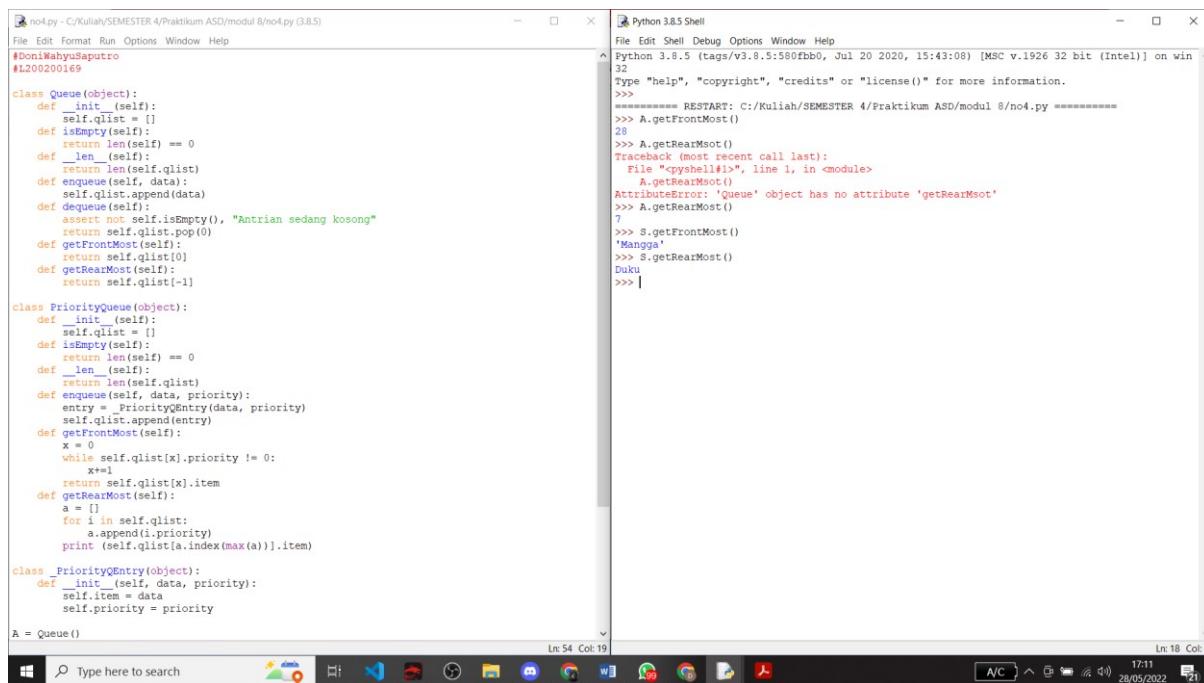
```
#no3.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no3.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

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(), "Tidka bisa diintip. Stack kosong"
        return self.items[-1]
    def pop(self):
        assert not self.isEmpty(), "Tidka bisa dipop dari Stack kosong"
        return self.items.pop()
    def push(self, data):
        self.items.append(data)

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

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no3.py =====
[0, 9, 12, 15]
>>>
```

### 4.



```
#no4.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no4.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

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[0]
    def getRearMost(self):
        return self.qlist[-1]

class PriorityQueue(object):
    def __init__(self):
        self.qlist = []
    def isEmpty(self):
        return len(self.qlist) == 0
    def __len__(self):
        return len(self.qlist)
    def enqueue(self, data, priority):
        entry = _PriorityEntry(data, priority)
        self.qlist.append(entry)
    def getFrontMost(self):
        x = 0
        while self.qlist[x].priority != 0:
            x+=1
        return self.qlist[x].item
    def getRearMost(self):
        a = []
        for i in self.qlist:
            a.append(i.priority)
        print (self.qlist[a.index(max(a))].item)

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

A = Queue()
```

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no4.py =====
28
>>> A.getFrontMost()
28
>>> A.getRearMost()
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    A.getRearMost()
AttributeError: 'Queue' object has no attribute 'getRearMost'
>>> S.getFrontMost()
>>> S.getRearMost()
'Manga'
>>> S.getRearMost()
Duku
>>> |
```

```

A = Queue()
A.enqueue(28)
A.enqueue(19)
A.enqueue(45)
A.enqueue(13)
A.enqueue(7)

S = PriorityQueue()
S.enqueue("Jeruk", 4)
S.enqueue("Tomat", 2)
S.enqueue("Mangga", 0)
S.enqueue("Duku", 5)
S.enqueue("Pepaya", 2)

```

## 5.

The screenshot shows a Windows desktop environment with two windows open. On the left is a code editor window titled "no5.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no5.py (3.8.5)". It contains Python code for implementing a queue and a priority queue. On the right is a Python shell window titled "Python 3.8.5 Shell". It shows the execution of the code and a test case where an assertion error occurs because the queue is not empty. The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 28/05/2022 17:18.

```

#DoniWahyuSupatro
#L200200169

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

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

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

    def enqueue(self, data, priority):
        entry = _PriorityQEntry(data, priority)
        self.qlist.append(entry)

    def dequeue(self):
        assert not self.isEmpty(), "Antrian sedang kosong"
        a = []
        for i in self.qlist:
            a.append(i.priority)
        print (self.qlist.pop(a.index(min(a))).item)

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

S = PriorityQueue()
S.enqueue("Jeruk", 4)
S.enqueue("Tomat", 2)
S.enqueue("Mangga", 0)
S.enqueue("Duku", 5)
S.enqueue("Pepaya", 2)

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no5.py
>>> S.dequeue()
Mangga
>>> S.dequeue()
Tomat
>>> S.dequeue()
Duku
>>> S.dequeue()
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    S.dequeue()
  File "C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 8/no5.py", line 19, in dequeue
    assert not self.isEmpty(), "Antrian sedang kosong"
AssertionError: Antrian sedang kosong
>>>

```

## **MODUL 9**

- 1) Tentukan jumlah level maximum dan minimum dari ukuran pohon biner berikut.

† n = 10

- jumlah level minimum =  $\text{INT}[\log_2 10] + 1 = 4$
- jumlah level maximum = (level 0 sampai level 9) = 10

† n = 35

- jumlah level minimum =  $\text{INT}[\log_2 35] + 1 = 6$
- jumlah level maximum = (level 0 sampai level 34) = 35

† n = 76

- jumlah level minimum =  $\text{INT}[\log_2 76] + 1 = 7$
- jumlah level maximum = (level 0 sampai level 75) = 76

† n = 345

- jumlah level minimum =  $\text{INT}[\log_2 345] + 1 = 9$
- jumlah level maximum = (level 0 sampai level 344) = 345

- 2) Ada berapa kemungkinan gambar yang dapat dibentuk dari pohon biner berukuran 5.

$$\begin{aligned}C_n &= (2n)! / (n+1)! * n! \\&= (2*5)! / (5+1)! + 5! \\&= 10! / 6! * 5! \\&= 3628800 / 86400 \\&= 42 \text{ kemungkinan}\end{aligned}$$

- 3) Tentukan jumlah simpul maksimum suatu pohon biner dengan jumlah level h a. h = 3

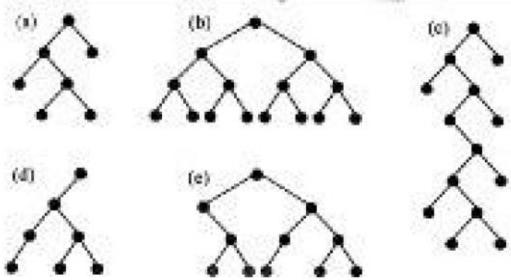
$$\begin{aligned}\text{Jumlah max simpul} &= \text{level 0} + \text{level 1} + \text{level 2} = 2^0 + 2^1 \\&+ 2^2 = 7 \quad \text{b. } h = 4\end{aligned}$$

$$\begin{aligned}\text{Jumlah max simpul} &= \text{level 0} + \text{level 1} + \text{level 2} + \text{level 3} = 2^0 + 2^1 + 2^2 \\&+ 2^3 = 15 \quad \text{c. } h = 5\end{aligned}$$

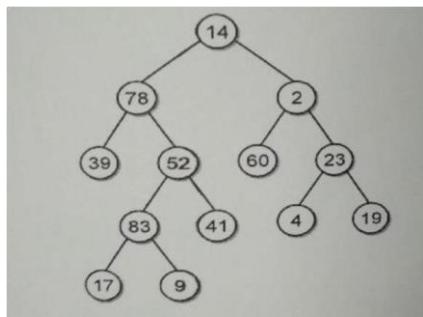
$$\begin{aligned}\text{Jumlah max simpul} &= \text{level 0} + \text{level 1} + \text{level 2} + \text{level 3} + \text{level 4} = 2^0 + 2^1 + 2^2 + \\&2^3 + 2^4 = 31 \quad \text{d. } h = 6\end{aligned}$$

$$\begin{aligned}\text{Jumlah max simpul} &= \text{level 0} + \text{level 1} + \text{level 2} + \text{level 3} + \text{level 4} + \text{level 5} = 2^0 + \\&2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 63\end{aligned}$$

- 4) Diberikan pohon-pohon biner seperti di bawah



- a. Tunjukkan semua properti struktural yang berlaku pada tiap-tiap pohon di atas : *penuh* , *sempurna*, *komplet*. Ingat bahwa sebuah pohon biner bisa saja bersifat penuh sekaligus sempurna dan sebagainya.
- a = penuh  
 b = sempurna  
 c = komplit dan penuh  
 d = komplit  
 e = komplit
- b. Tentukan ukuran tiap pohon.
- a = 7  
 b = 15  
 c = 14  
 d = 7  
 e = 11
- c. Tentukan ketinggian tiap pohon.
- a = 4  
 b = 4  
 c = 8  
 d = 4  
 e = 4
- d. Tentukan lebar tiap pohon.
- a = 2  
 b = 8  
 c = 2  
 d = 3  
 e = 5
- 5) Perhatikan pohon biner berikut.



a. Tunjukan urutan pengunjungan simpul untuk :

- Preorder traversal = 14-78-39-52-83-17-9-41-2-60-23-4-19
- Inorder traversal = 39-78-17-83-9-52-41-14-60-2-4-23-19
- Postorder traversal = 39-17-9-83-41-52-78-60-4-19-23-2-14

b. Simpul mana saja yang merupakan simpul daun ?

- 39, 17, 9, 41, 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

- $83 = 14 - 78 - 52 - 83$
- $39 = 14 - 78 - 39$
- $4 = 14 - 2 - 23 - 4$
- $9 = 14 - 78 - 52 - 83 - 9$

f. Perhatikan simpul 52 Tentukan!

- Keturunannya = 83, 41
- Leluhurnya = 78, 14
- Saudaranya = 39

g. Tentukan kedalaman dari tiap-tiap simpul ini :

- 78 = level 1
- 41 = level 2
- 60 = level 2
- 19 = level 3

6) Buatlah fungsi ukuranPohon(akar) yang akan mendapatkan ukuran sebuah pohon biner!

```

no 6.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 6.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

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

    def ukuranPohon(akar, count=0):
        if akar is None:
            return count

        return ukuranPohon(akar.kiri, ukuranPohon(akar.kanan, count+1))

a = SimpulPohonBiner('Ambarawa')
b = SimpulPohonBiner('Bantul')
c = SimpulPohonBiner('Cimahi')
d = SimpulPohonBiner('Denpasar')
e = SimpulPohonBiner('Enrekang')
f = SimpulPohonBiner('Flores')
g = SimpulPohonBiner('Garut')
h = SimpulPohonBiner('Halmahera Timur')
i = SimpulPohonBiner('Indramayu')
j = SimpulPohonBiner('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

print('Ukuran dari Binary Tree adalah',ukuranPohon(a))

```

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

```

no 7.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 7.py (3.8.5)
File Edit Format Run Options Window Help
#DoniWahyuSaputro
#L200200169

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

    def tinggiPohon(akar, count=0):
        if akar is None:
            return 0
        else:
            return max(tinggiPohon(akar.kiri), tinggiPohon(akar.kanan))+1

a = SimpulPohonBiner('Ambarawa')
b = SimpulPohonBiner('Bantul')
c = SimpulPohonBiner('Cimahi')
d = SimpulPohonBiner('Denpasar')
e = SimpulPohonBiner('Enrekang')
f = SimpulPohonBiner('Flores')
g = SimpulPohonBiner('Garut')
h = SimpulPohonBiner('Halmahera Timur')
i = SimpulPohonBiner('Indramayu')
j = SimpulPohonBiner('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

print ('Tinggi maksimal dari Binary Tree adalah', tinggiPohon(a))

```

8) Buatlah sebuah fungsi yang mencetak data tiap simpul sekaligus level di mana simpul itu berada. Silakan memilih akan memakai preorder traversal, inorder traversal, atau postorder traversal.

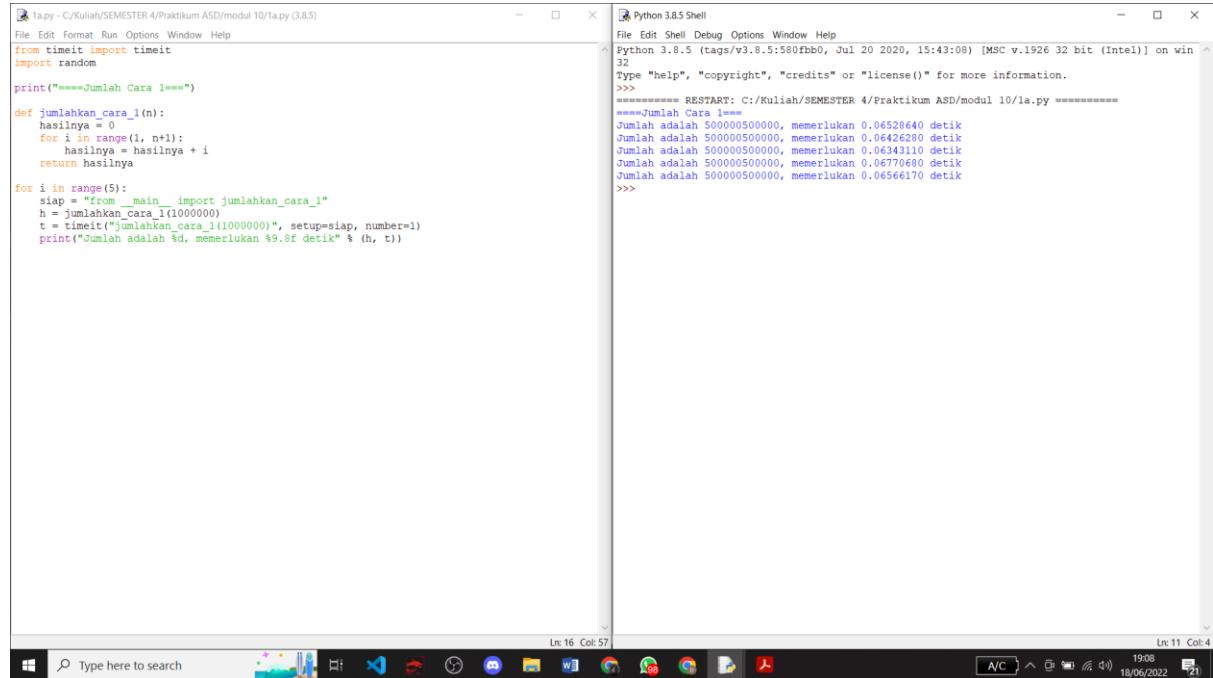
<pre> no 8.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 8.py (3.8.5) File Edit Format Run Options Window Help #DoniWahyuSaputro #L200200169  class simpulbiner(object):     def __init__(self, data):         self.data=data         self.kiri=None         self.kanan=None      def __str__(self):         return str(self.data)  A=simpulbiner('Ambarawa') B=simpulbiner('Bantul') C=simpulbiner('Cimahi') D=simpulbiner('Denpasar') E=simpulbiner('Enrekang') H=simpulbiner('Halmahera Timur')  A.kiri=B; A.kanan=C B.kiri=D; B.kanan=E D.kiri=H;  datalist=[A.data, B.data, C.data, D.data, E.data, H.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 postorder(sub): </pre>	<pre> Python 3.8.5 Shell File Edit Shell Debug Options Window Help Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel] on win32 Type "help", "copyright", "credits" or "license()" for more information. &gt;&gt;&gt; ===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 9/no 8.py ===== Ambarawa , Level 0 Bantul , Level 1 Cimahi , Level 1 Denpasar , Level 2 Enrekang , Level 2 Halmahera Timur , Level 3 =====preorder===== Ambarawa Bantul Denpasar Halmahera Timur Enrekang Cimahi ===== =====inorder===== Halmahera Timur Denpasar Bantul Enrekang Cimahi Ambarawa Cimahi ===== =====postorder===== Halmahera Timur Denpasar Enrekang Bantul Cimahi Ambarawa &gt;&gt;&gt;   </pre>
---	---

## MODUL 10

### Latihan

1.

a



```
1a.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 10/1a.py (3.8.5)
File Edit Format Run Options Window Help
from timeit import timeit
import random

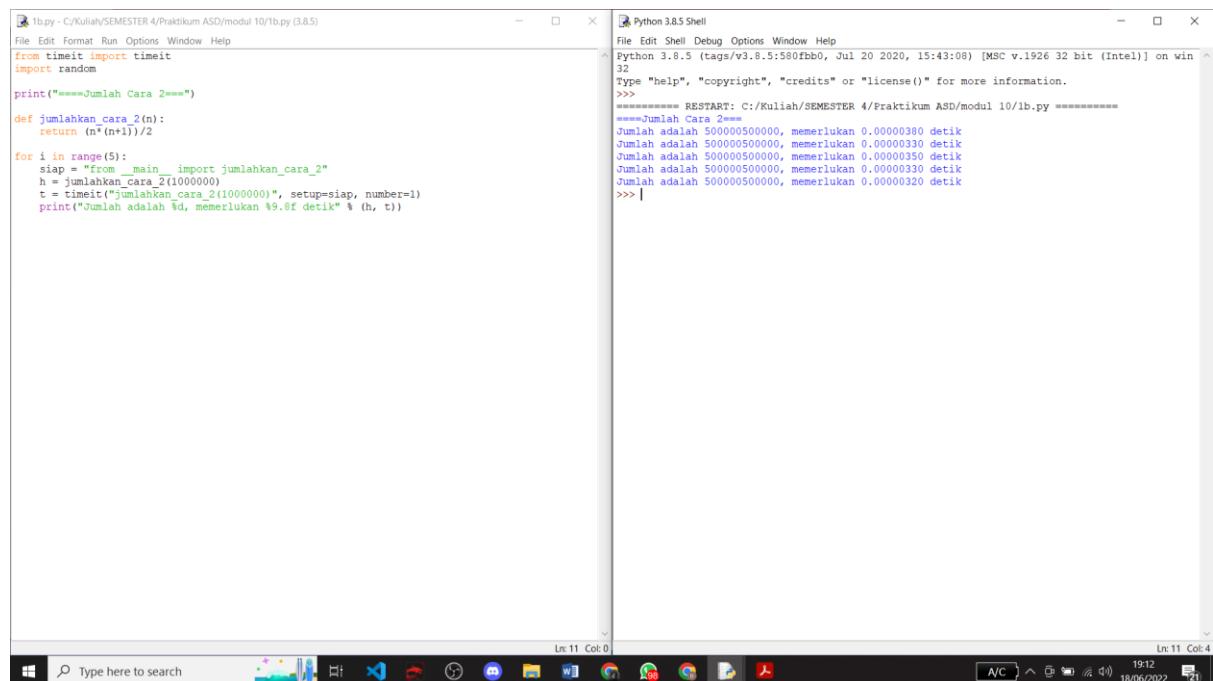
print("====Jumlah Cara 1====")

def jumlahkan_cara_1(n):
    hasilnya = 0
    for i in range(1, n+1):
        hasilnya = hasilnya + i
    return hasilnya

for i in range(5):
    siap = "from __main__ import jumlahkan_cara_1"
    h = jumlahkan_cara_1(1000000)
    t = timeit("jumlahkan_cara_1(1000000)", setup=siap, number=1)
    print("Jumlah adalah %d, memerlukan %9.8f detik" % (h, t))

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 10/1a.py =====
===== Jumlah Cara 1 ====
Jumlah adalah 500000500000, memerlukan 0.06528640 detik
Jumlah adalah 500000500000, memerlukan 0.06426280 detik
Jumlah adalah 500000500000, memerlukan 0.06343110 detik
Jumlah adalah 500000500000, memerlukan 0.06770680 detik
Jumlah adalah 500000500000, memerlukan 0.06566170 detik
>>>
```

b



```
1b.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 10/1b.py (3.8.5)
File Edit Format Run Options Window Help
from timeit import timeit
import random

print("====Jumlah Cara 2====")

def jumlahkan_cara_2(n):
    return (n*(n+1))/2

for i in range(5):
    siap = "from __main__ import jumlahkan_cara_2"
    h = jumlahkan_cara_2(1000000)
    t = timeit("jumlahkan_cara_2(1000000)", setup=siap, number=1)
    print("Jumlah adalah %d, memerlukan %9.8f detik" % (h, t))

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 10/1b.py =====
===== Jumlah Cara 2 ====
Jumlah adalah 500000500000, memerlukan 0.00000380 detik
Jumlah adalah 500000500000, memerlukan 0.00000330 detik
Jumlah adalah 500000500000, memerlukan 0.00000350 detik
Jumlah adalah 500000500000, memerlukan 0.00000330 detik
Jumlah adalah 500000500000, memerlukan 0.00000320 detik
>>> |
```

C

```
1c.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 10/1c.py (3.8.5)
File Edit Format Run Options Window Help
from timeit import timeit
import random

print("====Insertion Sort====")

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

for i in range(5):
    siap = "from __main__ import insertionSort,L"
    L = list(range(3000))
    t = timeit("insertionSort(L)", setup=siap, number=1)
    print("Jumlah adalah %d bilangan, memerlukan %.7f detik" % (len(L), t))

Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 10/1c.py =====
====Insertion Sort====
Jumlah adalah 3000 bilangan, memerlukan 0.0021099 detik
Jumlah adalah 3000 bilangan, memerlukan 0.0021175 detik
Jumlah adalah 3000 bilangan, memerlukan 0.0013292 detik
Jumlah adalah 3000 bilangan, memerlukan 0.0013292 detik
Jumlah adalah 3000 bilangan, memerlukan 0.0013169 detik
>>>
```

2.

```
2.py - C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 10/2.py (3.8.5)
File Edit Shell Debug Options Window Help
from timeit import timeit
import random

print("==Sorted avg case==")

for i in range(5):
    g = list(range(3000))
    random.shuffle(g)
    t = timeit("sorted(g)", setup="from __main__ import g,number=1")
    print("Mengurutkan %d bilangan, memerlukan %.7f detik" % (len(g), t))

print("==Sorted best case==")

for i in range(5):
    g = list(range(3000))
    t = timeit("sorted(g)", setup="from __main__ import g,number=1")
    print("Mengurutkan %d bilangan, memerlukan %.7f detik" % (len(g), t))

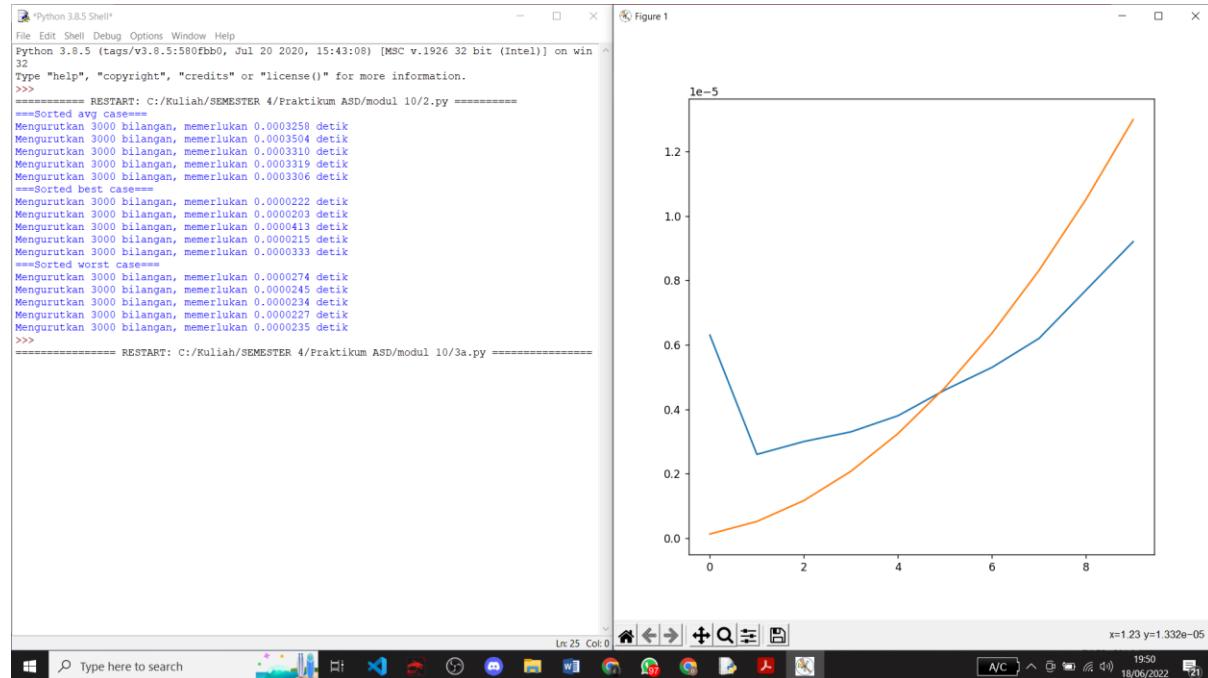
print("==Sorted worst case==")

for i in range(5):
    g = list(range(3000))
    g = g[::-1]
    t = timeit("sorted(g)", setup="from __main__ import g,number=1")
    print("Mengurutkan %d bilangan, memerlukan %.7f detik" % (len(g), t))

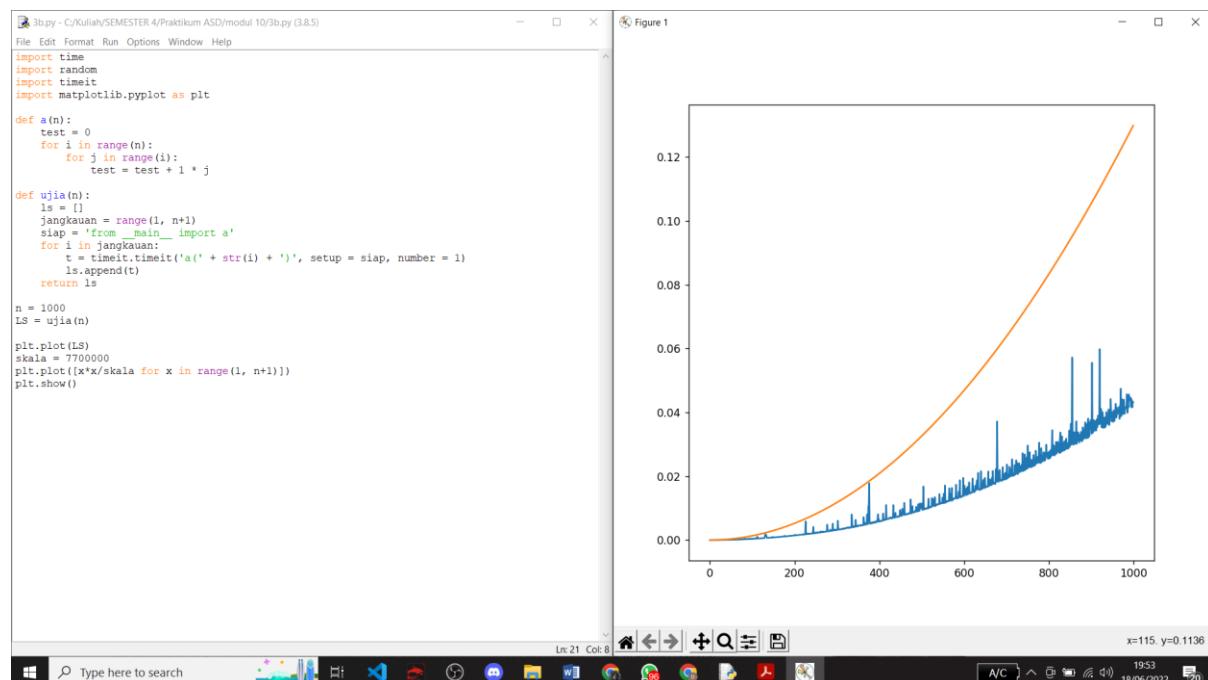
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Kuliah/SEMESTER 4/Praktikum ASD/modul 10/2.py =====
==Sorted avg case==
Mengurutkan 3000 bilangan, memerlukan 0.0003258 detik
Mengurutkan 3000 bilangan, memerlukan 0.0003504 detik
Mengurutkan 3000 bilangan, memerlukan 0.0003310 detik
Mengurutkan 3000 bilangan, memerlukan 0.0003319 detik
Mengurutkan 3000 bilangan, memerlukan 0.0003306 detik
==Sorted best case==
Mengurutkan 3000 bilangan, memerlukan 0.0000223 detik
Mengurutkan 3000 bilangan, memerlukan 0.0000233 detik
Mengurutkan 3000 bilangan, memerlukan 0.0000413 detik
Mengurutkan 3000 bilangan, memerlukan 0.0000215 detik
Mengurutkan 3000 bilangan, memerlukan 0.0000333 detik
==Sorted worst case==
Mengurutkan 3000 bilangan, memerlukan 0.0000274 detik
Mengurutkan 3000 bilangan, memerlukan 0.0000245 detik
Mengurutkan 3000 bilangan, memerlukan 0.000023 detik
Mengurutkan 3000 bilangan, memerlukan 0.0000227 detik
Mengurutkan 3000 bilangan, memerlukan 0.0000235 detik
>>>
```

3.

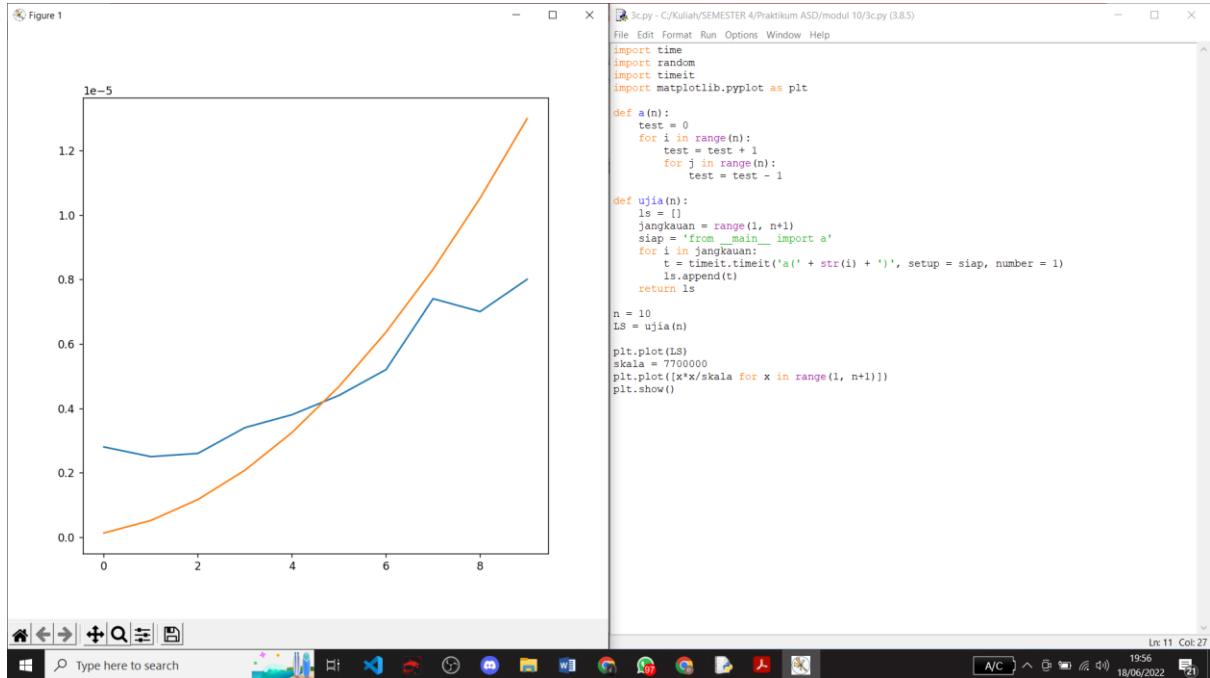
a



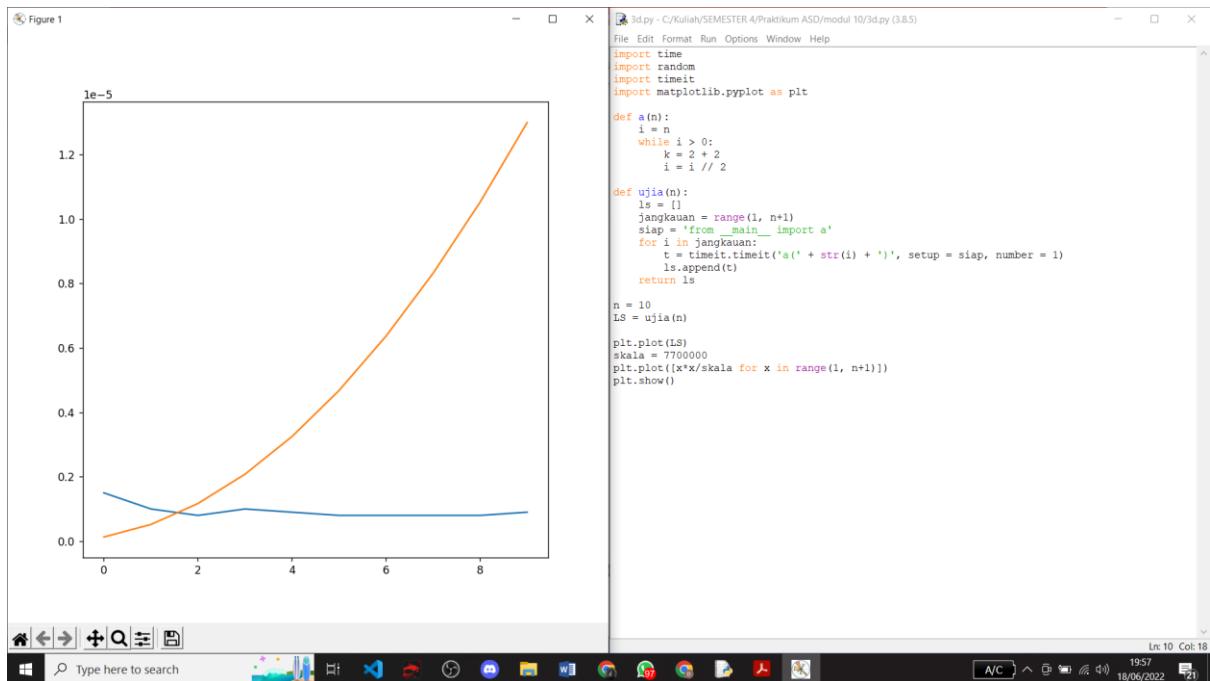
b



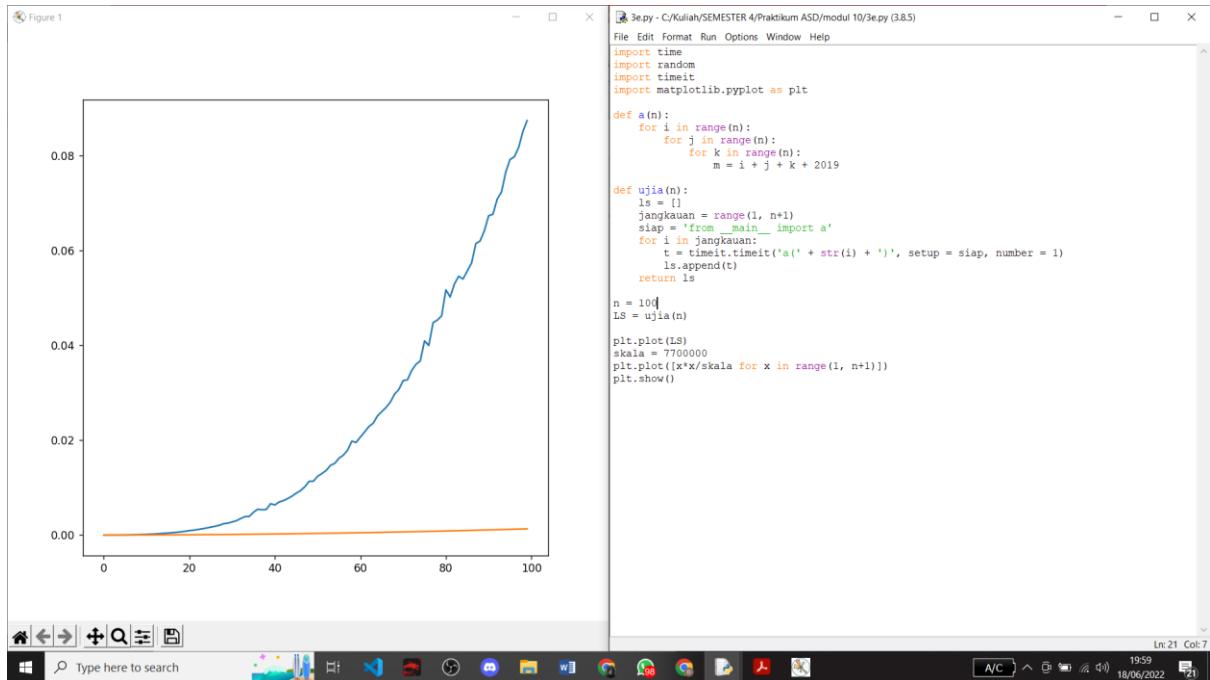
C



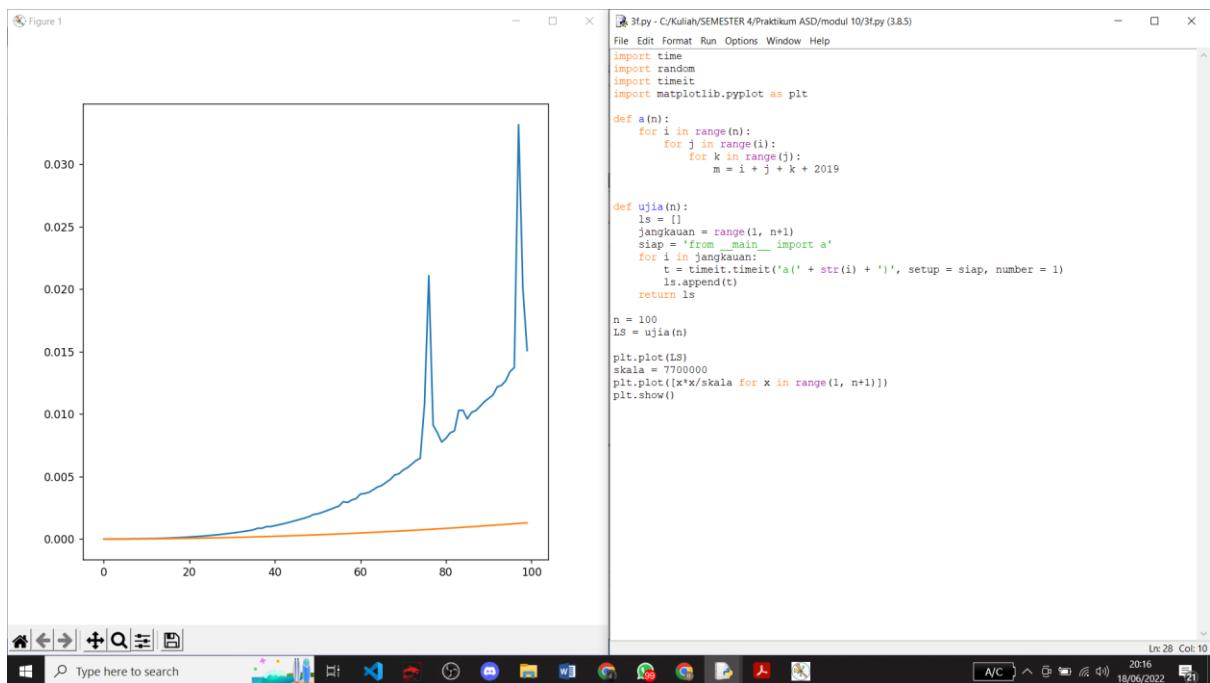
d



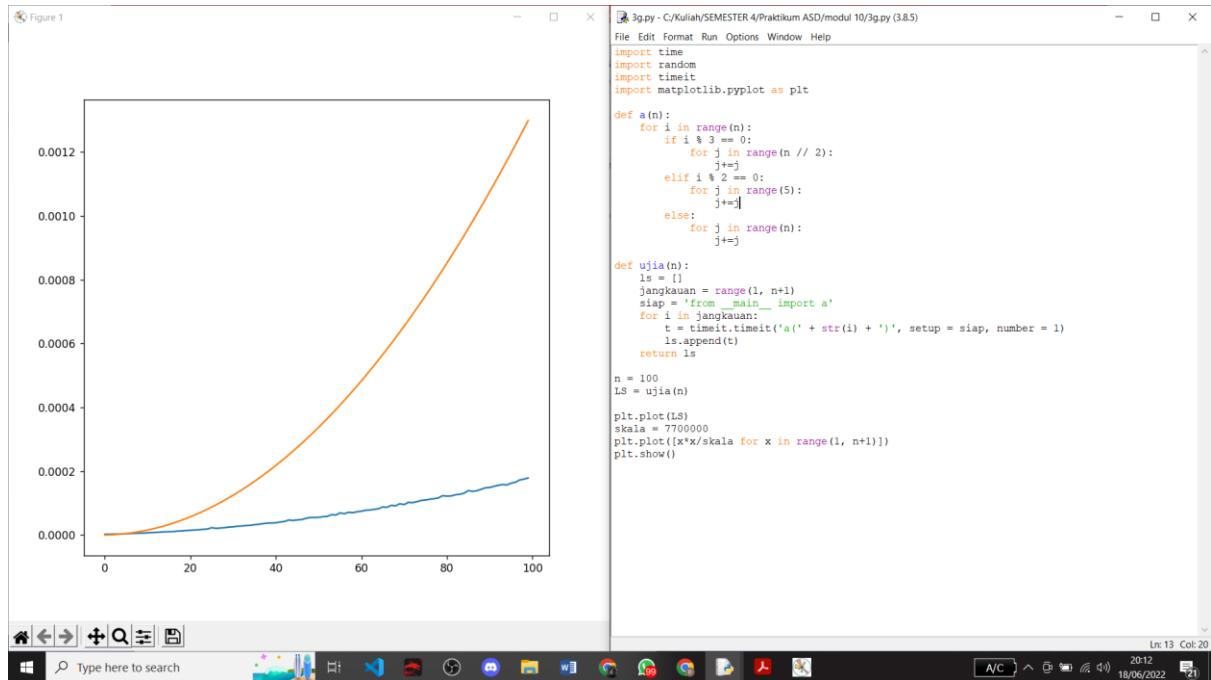
e



f



g



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

5. Tentukan  $O(\cdot)$  dari fungsi-fungsi berikut, yang mewakili banyaknya langkah yang diperlukan 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. (Literatur Review) carilah di internet, kompleksitas metode-metode pada object list di Python.  
Hint

- Google python list methods complexity. Lihat juga bagian 'Images'-nya
- Kunjungi <https://wiki.python.org/moin/TimeComplexity>

wiki.python.org/moin/TimeComplexity

list

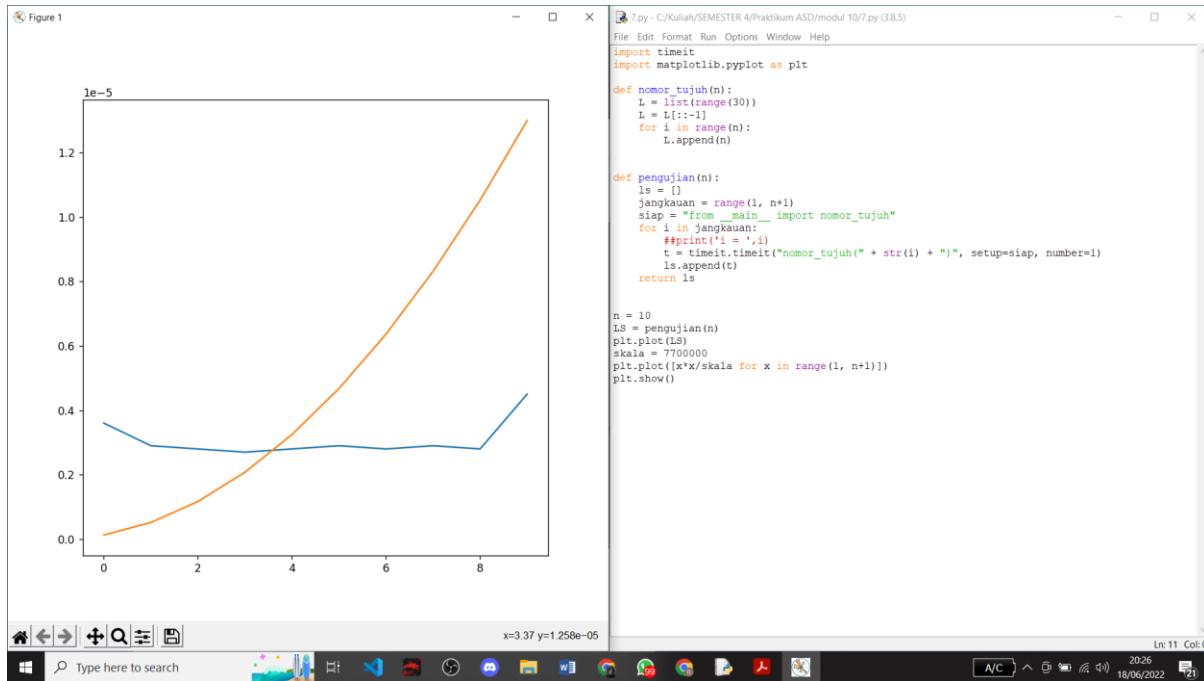
The Average Case assumes parameters generated uniformly at random.

Internally, a list is represented as an array; the largest costs come from growing beyond the current allocation size (because everything must move), or from inserting or deleting somewhere near the beginning (because everything after that must move). If you need to add/remove at both ends, consider using a collections.deque instead.

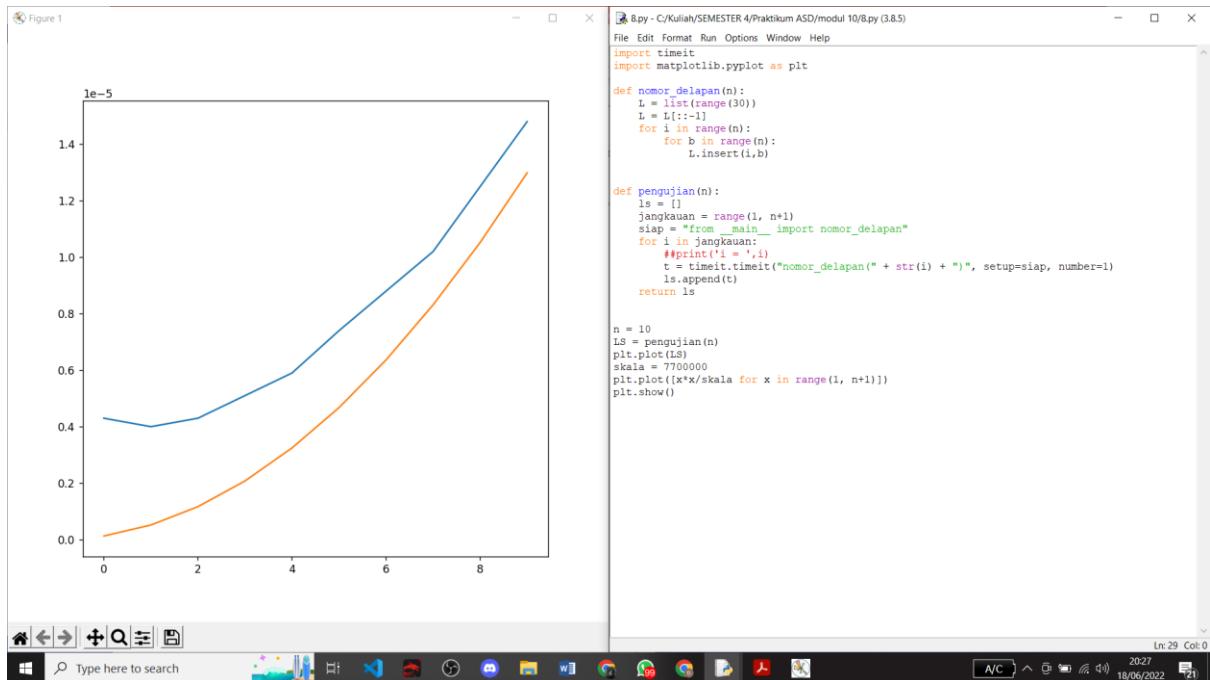
Operation	Average Case	Amortized Worst Case
Copy	$O(n)$	$O(n)$
Append[1]	$O(1)$	$O(1)$
Pop last	$O(1)$	$O(1)$
Pop intermediate[2]	$O(n)$	$O(n)$
Insert	$O(n)$	$O(n)$
Get Item	$O(1)$	$O(1)$
Set Item	$O(1)$	$O(1)$
Iteration	$O(n)$	$O(n)$
Get Slice	$O(k)$	$O(k)$
Del Slice	$O(n)$	$O(n)$
Set Slice	$O(k+n)$	$O(k+n)$
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	$O(1)$	$O(1)$

collections.deque

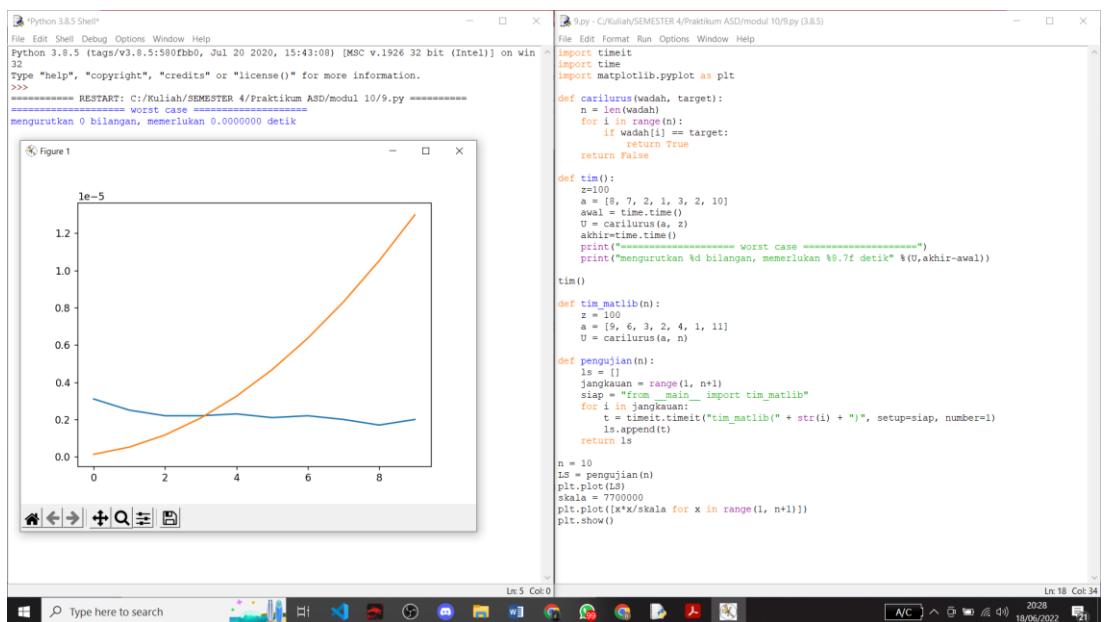
7.



8.



## 9.



## 10.

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.

- Big O dilambangkan dengan notasi  $O(\dots)$  merupakan keadaan terburuk (worst case). Kinerja sebuah algoritma biasanya diukur menggunakan patokan keadaan Big-O ini. Merupakan notasi asymptotic untuk batas fungsi dari atas dan bawah dengan Berperilaku mirip dengan  $\leq$  operator untuk tingkat pertumbuhan.
- Big Theta dilambangkan dengan notasi  $\Theta(\dots)$  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  $\Omega(\dots)$  merupakan notasi asymptotic untuk batas bawah dengan keadaan rata-rata(average case) yang berperilaku mirip dengan  $\geq$ operator untuk tingkat pertumbuhan.

12.

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 untuk melakukan satu urutan operasi pada struktur data terhadap keseluruhan operasi yang dilakukan

# QUIZ

## Source Code No 1,2 dan output 1 2 3

```
quiz.py - C:\Kuliah\SEMESTER 4\Praktikum ASD\quiz.py (3.8.5)
File Edit Format Run Options Window Help
email = "donistudent.ums.ac.id"
check(email)

email = "doni@ums.ac.id"
check(email)

email = "hhsdhfh.com"
check(email)

##No2
print("")
print("No2")
def check(email):
    if re.match("[a-zA-Z]+@[a-zA-Z]+\.\w{2,3}", email):
        print("True")
    else:
        print("False")

if __name__ == '__main__':
    email = "donistudent.ums.ac.id"
    check(email)

    email = "doni@ums.ac.id"
    check(email)

    email = "hhsdhfh@ums.com"
    check(email)

##No3
print("")
print("No3")
from binarytree import Node

root = Node(1)
root.left = Node(2)
root.right = Node(4)
root.right.left = Node(3)
root.right.right = Node(5)
root.right.right.right = Node(8)
root.right.right.right.left = Node(6)
root.right.right.right.right = Node(10)
print(root)

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
>>> ===== RESTART: C:\Kuliah\SEMESTER 4\Praktikum ASD\quiz.py =====
No1
True
True
False

No2
True
False
False

No3
1
 / \
2   4
  / \ / \
3   5   8
     / \ / \
6   10 8 10
>>> ===== RESTART: C:\Kuliah\SEMESTER 4\Praktikum ASD\quiz.py =====
No1
True
True
False

No2
True
False
False

No3
1
 / \
2   4
  / \ / \
3   5   8
     / \ / \
6   10 8 10
```

## Source code no 3

```
##No3
print("")
print("No3")
from binarytree import Node

root = Node(1)
root.left = Node(2)
root.right = Node(4)
root.right.left = Node(3)
root.right.right = Node(5)
root.right.right.right = Node(8)
root.right.right.right.left = Node(6)
root.right.right.right.right = Node(10)
print(root)
```

## No4

### preorderTrav:

cara mengecek preordeer yaitu dari Tengah,Kiri,Kanan

1 dicek tengah hasilnya true maka 1 ditulis, setelah sudah dicek tengah sekarang dicek kiri ada 2,2dicek tengah hasilnya true maka 2 ditulis, karena 1 sudah dicek ke kiri sekarang ke kanan

ada 4,4 dicek tengah hasilnya true maka 4 ditulis,lalu 4 dicek ke kiri ada 3,3 dicek tengah hasilnya true maka 3 ditulis,4 karena tengah sudah dicek,kiri sudah dicek sekarang kanan ada 5,5dicek tengah hasilnya true maka 5 ditulis setelah 3,sekarang 5 dicek setelah tengah,dicek kiri false,dicek kanan ada 8,8 dicek tengah hasilnya true maka 8 ditulis,8 sudah dicek tengah sekarang dicek kiri ada 6,6 dicek tengah hasilnya true maka 6 ditulis,lalu kembali ke 8,8 sudah dicek tengah dan kiri,lalu sekarang dicek ke kana nada 10,10 dicek tengah hasilnya true maka 10 ditulis setelah 6.

Jadi urutannya 1,2,4,3,5,8,6,10

#### **inorderTrav:**

cara mengecek Kiri,Tengah,Kanan

1 dicek ke kiri ada 2,2 dicek ke kiri hasilnya false lalu dicek tengah true maka 2 dimasukan,lalu kembali ke angka 1,setelah dicek kekiri sekarang cek tengah dan hasilnya 1 true maka 1 dimasukkan,lalu ke kana nada 4,4 dicek kiri ada 3,3dicek kiri false lalu dicek tengah hasilnya true maka 3 dimasukan,kembali ke 4 sekarang 4 dicek ke tengah hasilnya true maka 4 ditulis,sekarang ke 5,5 dicek kiri false lalu dicek tengah hasilnya true maka 5 ditulis, sekarang ke 8,8 dicek kiri ada 6,6 dicek kiri false,dicek tengah true maka 6 ditulis, kembali ke 8 lagi,8 dicek tengah true maka 8 ditulis,dan yang terakhir 10,10 dicek kiri false,lalu dicek tengah hasilnya true yang terakhir adalah 10

Jadi urutannya 2,1,3,4,5,6,8,10

#### **postorderTrav:**

cara mengecek Kiri,Kanan,Tengah

1 dicek kiri ada 2,2 dicek kiri false,dicek kanan false,dicek tengah true maka 2 ditulis,kembali ke 1,karena 1 sudah dicek ke kiri sekarang dicek ke kanan ada 4,4 dicek kiri ada 3,3 dicek ke kiri false,dicek kanan false,dicek tengah true maka 3 ditulis,kembali ke 1,1 dicek kiri sudah,dicek kana nada 4,4 dicek kiri sudah dicek kana nada 5,5 dicek kiri false,dicek kanan ada 8,8 dicek kiri ada 6,6 dicek kiri false,kanan false,tengah true maka 6 ditulis.Kembali ke 1,1 dicek kana nada 4,4dicek kana nada 5,5dicek kana nada 8,8dicek kanan ada 10,10 dicek kiri false,kanan false,tengah true maka 10 ditulis setelah 6.Kembali ke 1,1 dicek kanan ada 4,4 dicek kanan ada 5,5 dicek kana nada 8,8 kiri sudah dicek,kanan sudah dicek,sekarang tengah dan hasilnya true maka 8 ditulis setelah 10.Kembali lagi ke 1,1 dicek kanan ada 4,4 dicek kanan ada 5,5 karena kanan sudah dicek sekarang tengah dan hasilnya true, maka 5 ditulis.Kembali ke 1,1 dicek ke kanan ada 4,4 karena sudah dicek kiri dan kanan,sekarang tengah dan hasilnya true maka 4 ditulis.Dan yang terakhir 1,1 kiri sudah dicek,kanan sudah dicek,sekarang tengah dicek dan hasilnya true maka 1 ditulis.

Jadi urutannya 2,3,6,10,8,5,4,1