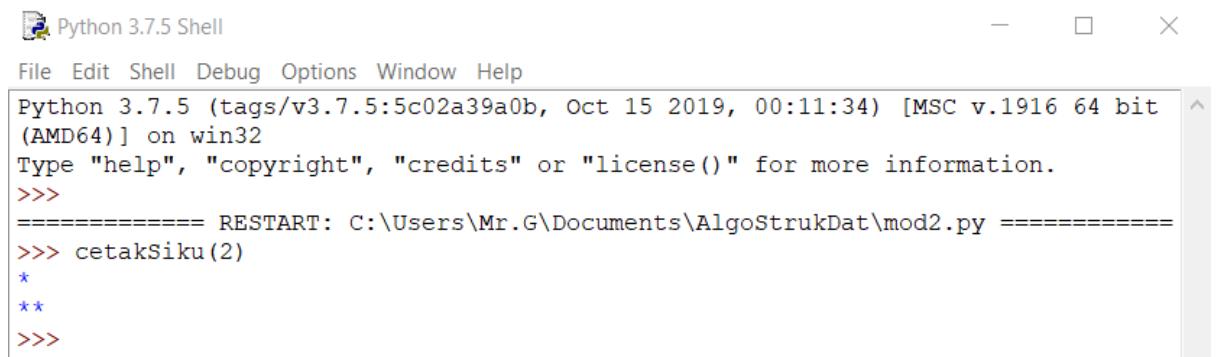


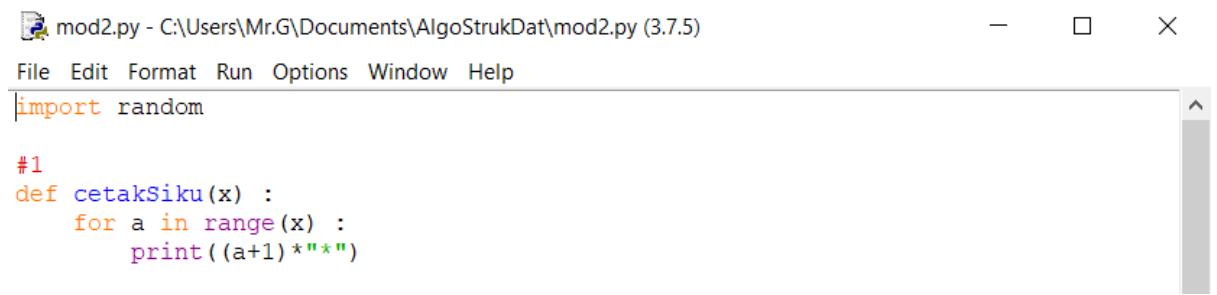
Nama : Guntur Jatmiko  
NIM : L200180039  
Kelas : B

## MODUL 1 ASD

### 1. Output file 1.py



```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
>>> cetakSiku(2)
*
**
>>>
```



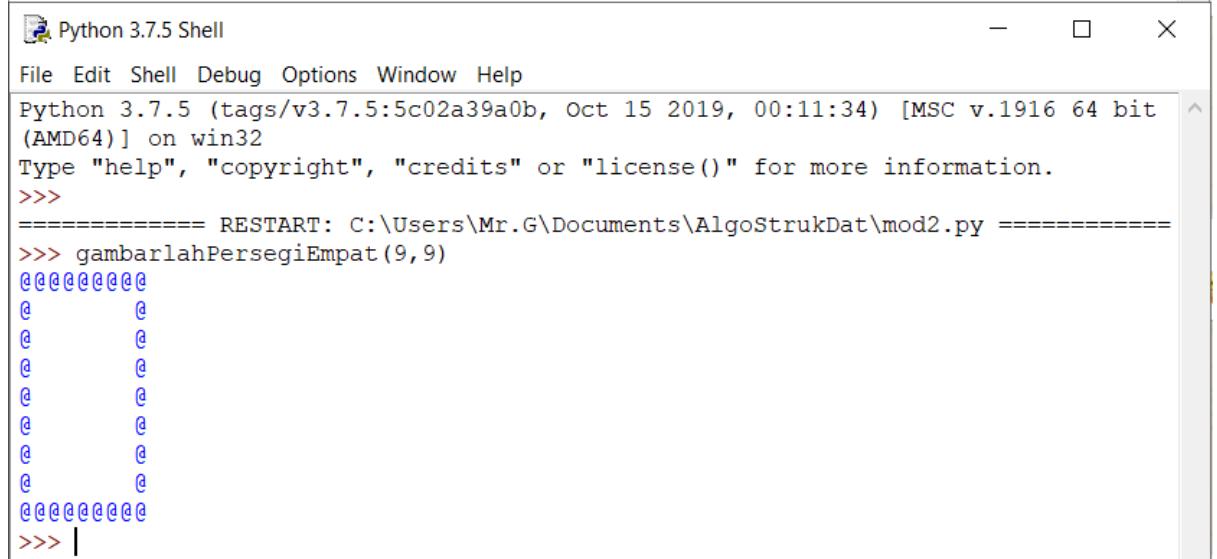
```
mod2.py - C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py (3.7.5)

File Edit Format Run Options Window Help
import random

#1
def cetakSiku(x) :
    for a in range(x) :
        print((a+1)*"**")
```

### 2. Output file 2.py

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



```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
>>> gambarlahPersegiEmpat(9,9)
@@@@@@@ @@
@ @ @ @ @
@ @ @ @ @
@ @ @ @ @
@ @ @ @ @
@ @ @ @ @
@ @ @ @ @
@ @ @ @ @
>>> |
```

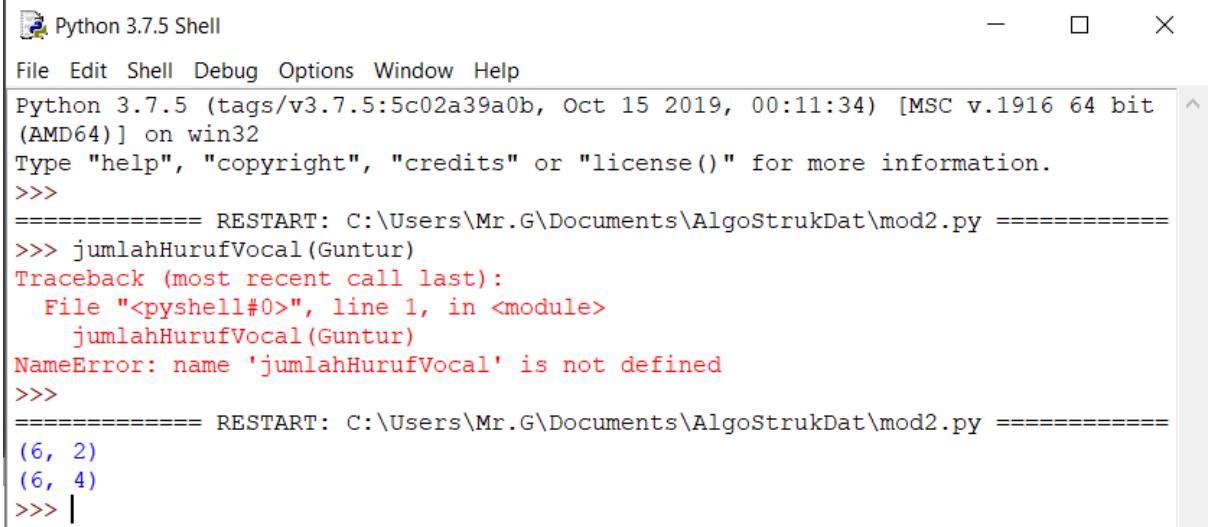
### 3. Output file 3.py

```
#3
def jumlahHurufVokal(ch) :
    b = len(ch)
    a = 0
    for i in ch :
        if (i=='A' or i=='a' or i=='E' or i =='e' or i=='I' or i=='i' or i=='O'
            a += 1
    return b,a

def jumlahHurufKonsonan(ch) :
    b = len(ch)
    a = 0
    for i in ch :
        if (i=='A' or i=='a' or i=='E' or i =='e' or i=='I' or i=='i' or i=='O'
            a += 1
    return b,b-a

v = jumlahHurufVokal("Guntur")
k = jumlahHurufKonsonan("Guntur")

print(v)
print(k)
```

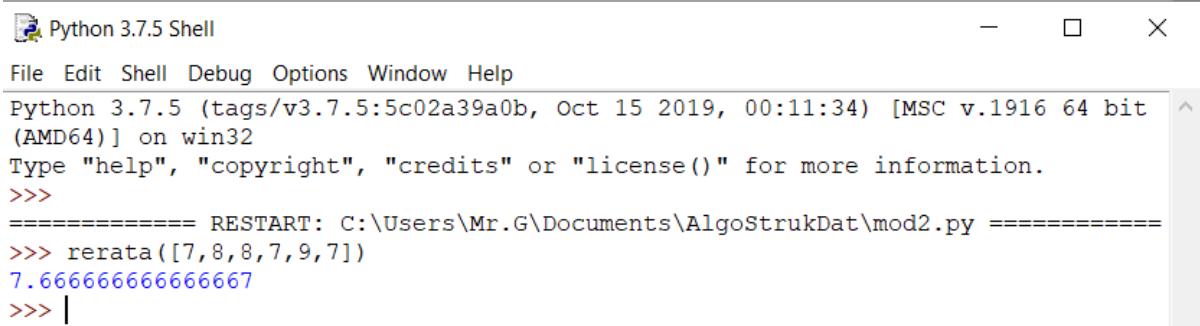


The screenshot shows a Python 3.7.5 shell window. The code in the editor is identical to the one above. When run, it prints the total length of the string "Guntur" (6) and the count of consonants (4). However, when attempting to call the function `jumlahHurufVocal` directly, it results in a NameError because the function is defined in a different scope (the module level). The shell output is as follows:

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
>>> jumlahHurufVocal(Guntur)
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    jumlahHurufVocal(Guntur)
NameError: name 'jumlahHurufVocal' is not defined
>>>
=====
RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
(6, 2)
(6, 4)
>>> |
```

#### 4. Output file 4.py

```
#4
def rerata(x) :
    a = 0
    b = 0
    for i in x :
        a += 1
        b = b + i
        a = float(a)
        b = float(b)
    return(b/a)
```



Python 3.7.5 Shell

File Edit Shell Debug Options Window Help

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32

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

>>>

===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====

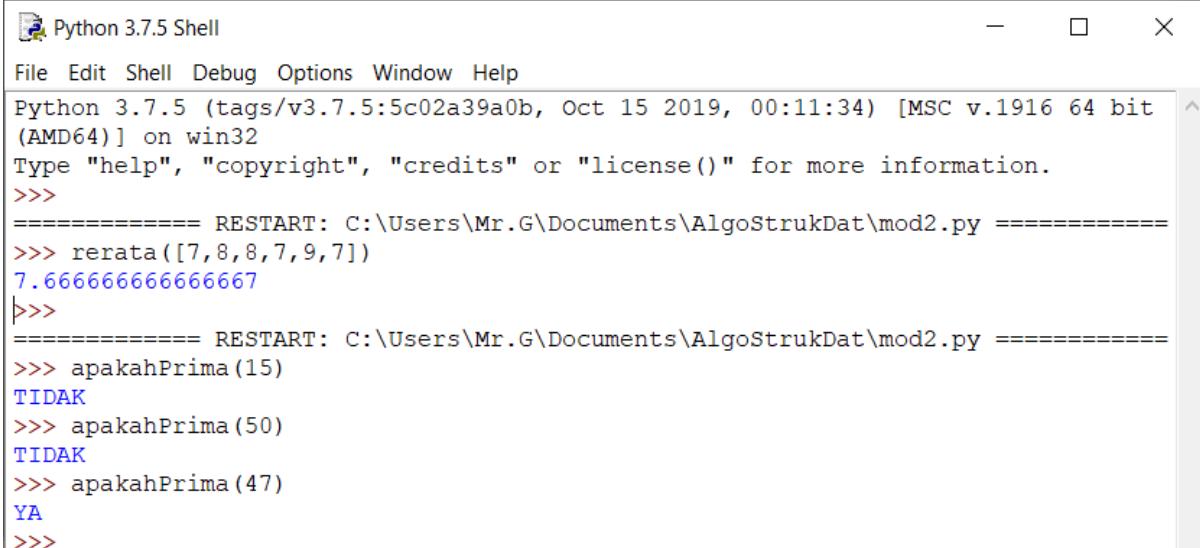
>>> rerata([7,8,8,7,9,7])

7.666666666666667

>>> |

#### 5. Output file 5.py

```
#5
def apakahPrima(a) :
    x = 0
    for i in range(a) :
        if a % (i+1) == 0 :
            x += 1
    if x == 2 :
        print("YA")
    else :
        print("TIDAK")
```



Python 3.7.5 Shell

File Edit Shell Debug Options Window Help

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32

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

>>>

===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====

>>> rerata([7,8,8,7,9,7])

7.666666666666667

>>>

===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====

>>> apakahPrima(15)

TIDAK

>>> apakahPrima(50)

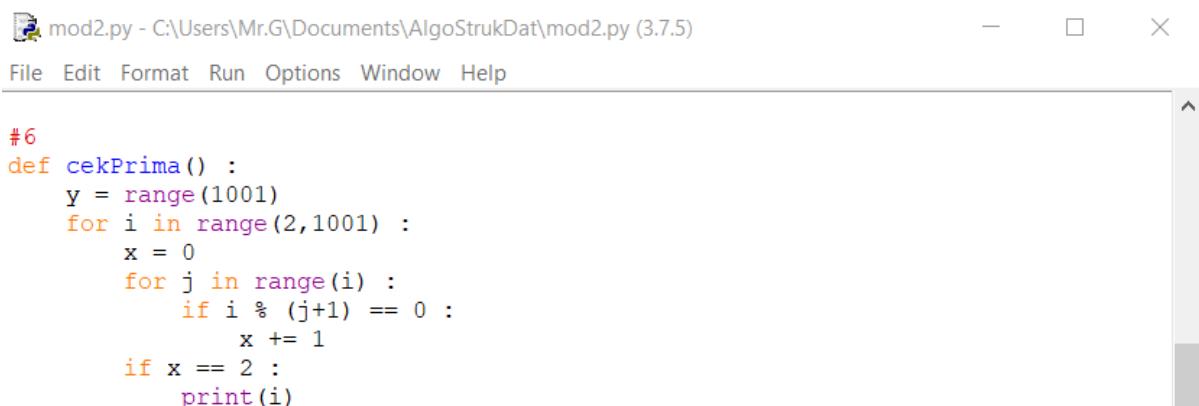
TIDAK

>>> apakahPrima(47)

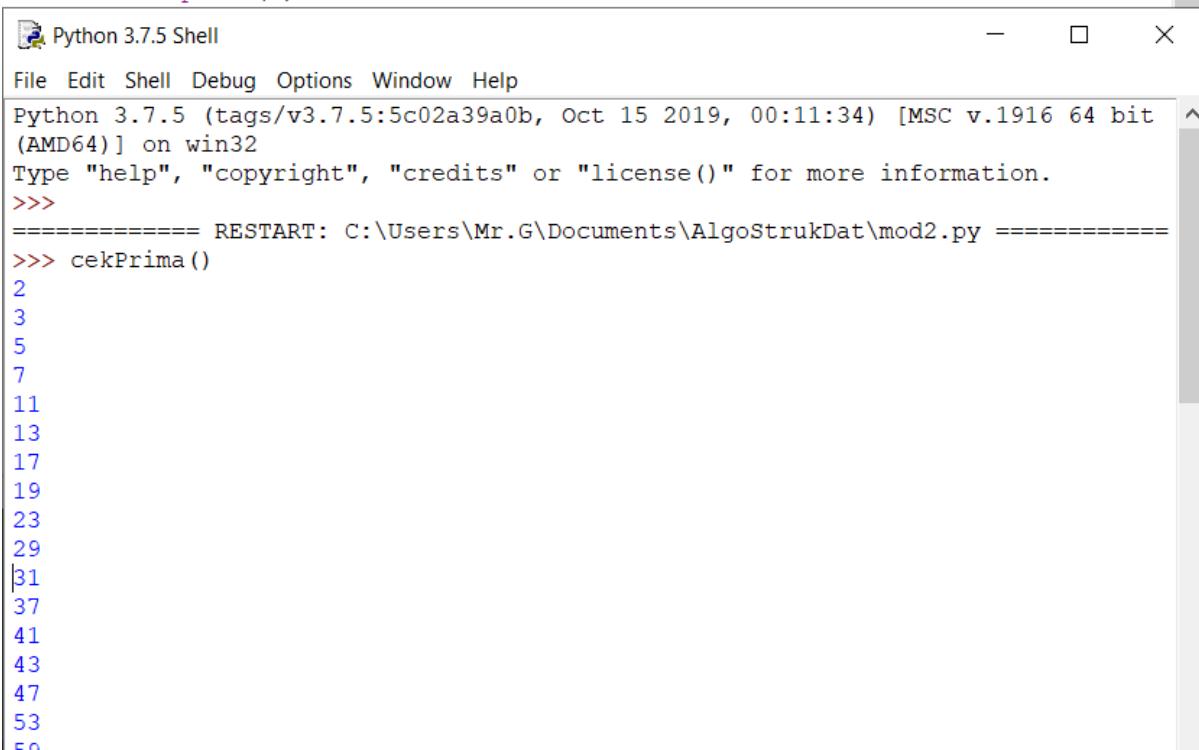
YA

>>>

## 6. Output file 6.py



```
#6
def cekPrima() :
    y = range(1001)
    for i in range(2,1001) :
        x = 0
        for j in range(i) :
            if i % (j+1) == 0 :
                x += 1
        if x == 2 :
            print(i)
```



```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
>>> cekPrima()
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
```

7. Output file 7.py

```
#7
def faktorPrima(x):
    listprima=[]
    prima=2
    while prima<=x:
        if x%prima==0:
            x/=prima
            listprima.append(prima)
        else:
            prima+=1
    return listprima
```

Python 3.7.5 Shell

File Edit Shell Debug Options Window Help

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32

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

>>>

===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====

>>> faktorPrima(143)

[11, 13]

8. Output file 8.py

```
#8
def apakahTerkandung(a,b) :
    if a in b :
        return True
    else :
        return False
```

Python 3.7.5 Shell

File Edit Shell Debug Options Window Help

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32

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

>>>

===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====

>>> apakahTerkandung("db","awddasadadawaws")

False

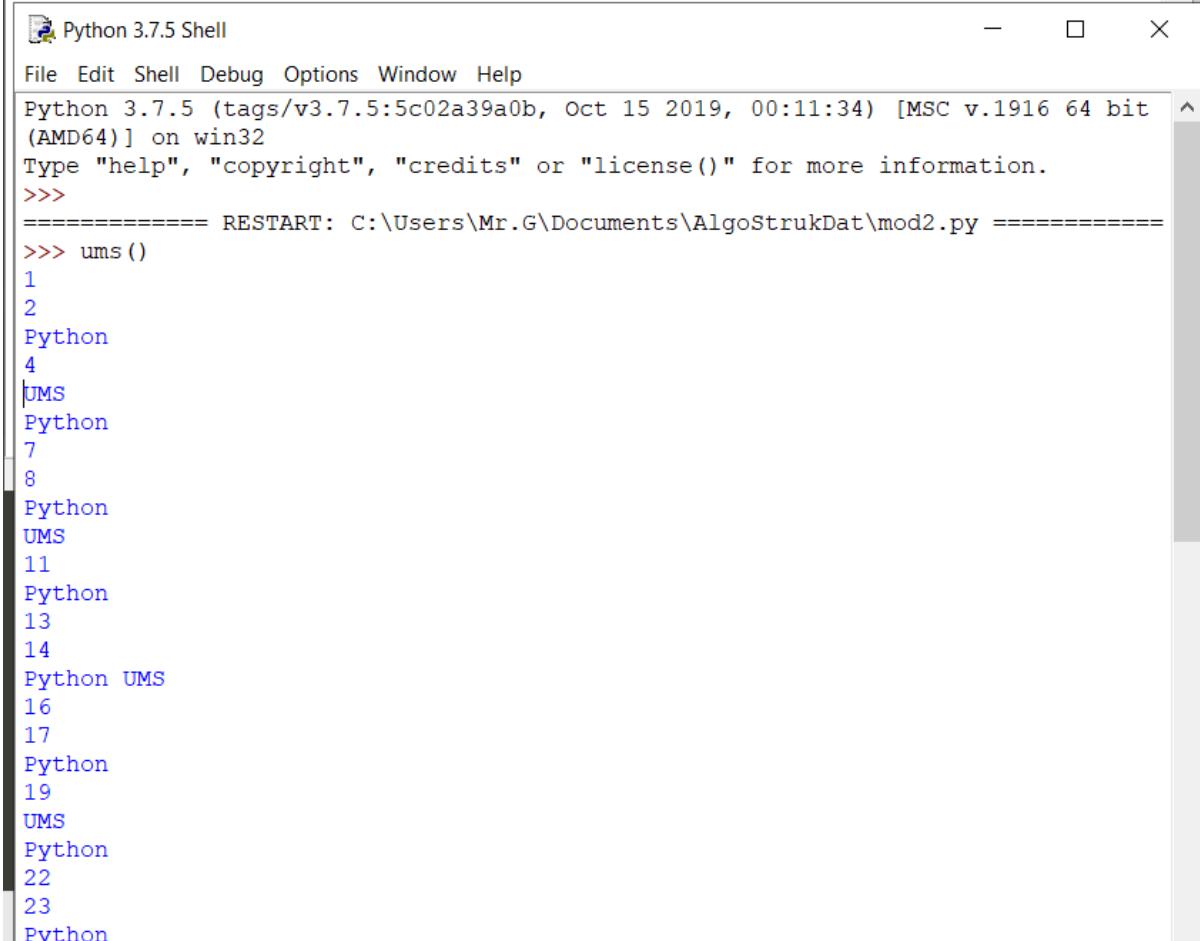
>>> apakahTerkandung("awkarin","awokawokawokoawkarin")

True

>>> |

## 9. Output file 9.py

```
#9
def ums() :
    for i in range(101) :
        if (i+1) % 15 == 0 :
            print("Python UMS")
        elif (i+1) % 3 == 0 :
            print("Python")
        elif (i+1) % 5 == 0 :
            print("UMS")
        else :
            print(i+1)
```



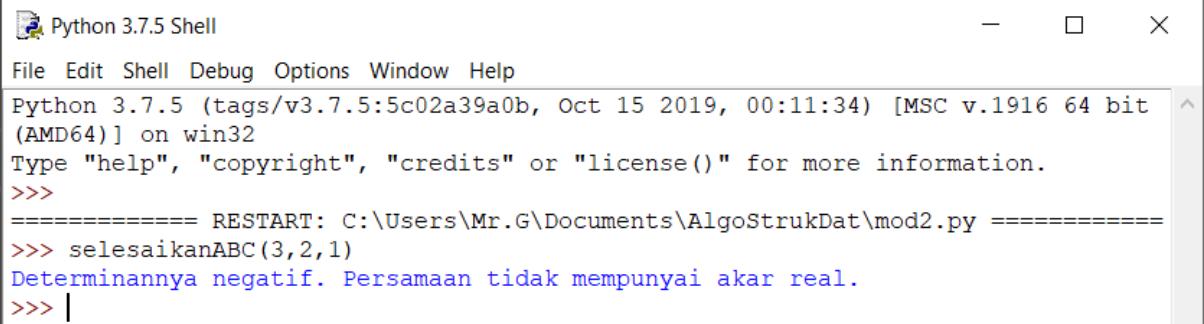
The screenshot shows a Python 3.7.5 Shell window. The title bar reads "Python 3.7.5 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell displays the following output:

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
>>> ums()
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
```

#### 10. Output file 10.py

```
#10
def selesaikanABC(a,b,c) :
    res = 0
    res = (b**2) - (4*a*c)

    if res == 0 :
        print("Determinannya nol. Persamaan mempunyai satu akar kembar.")
    elif res > 0 :
        print("Determinannya positif. Persamaan mempunyai akar real dan berlaina")
    elif res < 0 :
        print("Determinannya negatif. Persamaan tidak mempunyai akar real.")
```



Python 3.7.5 Shell

File Edit Shell Debug Options Window Help

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32

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

>>>

===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====

>>> selesaikanABC(3,2,1)

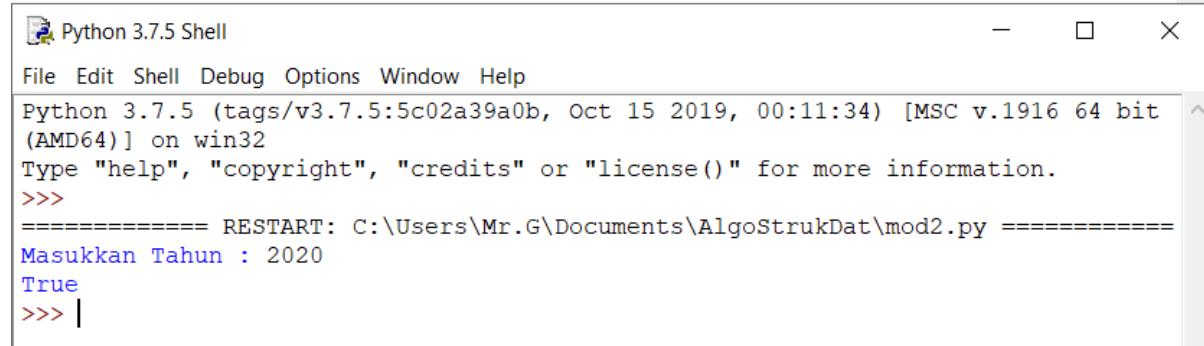
Determinannya negatif. Persamaan tidak mempunyai akar real.

>>> |

#### 11. Output file 11.py

```
#11
def apakahKabisat() :
    thn = int(input("Masukkan Tahun : "))
    if thn % 4 == 0 :
        if thn % 100 == 0 :
            if thn % 400 == 0 :
                print(True)
            else :
                print(False)
        else :
            print(True)
    else :
        print(False)
```

```
apakahKabisat()
```



Python 3.7.5 Shell

File Edit Shell Debug Options Window Help

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32

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

>>>

===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====

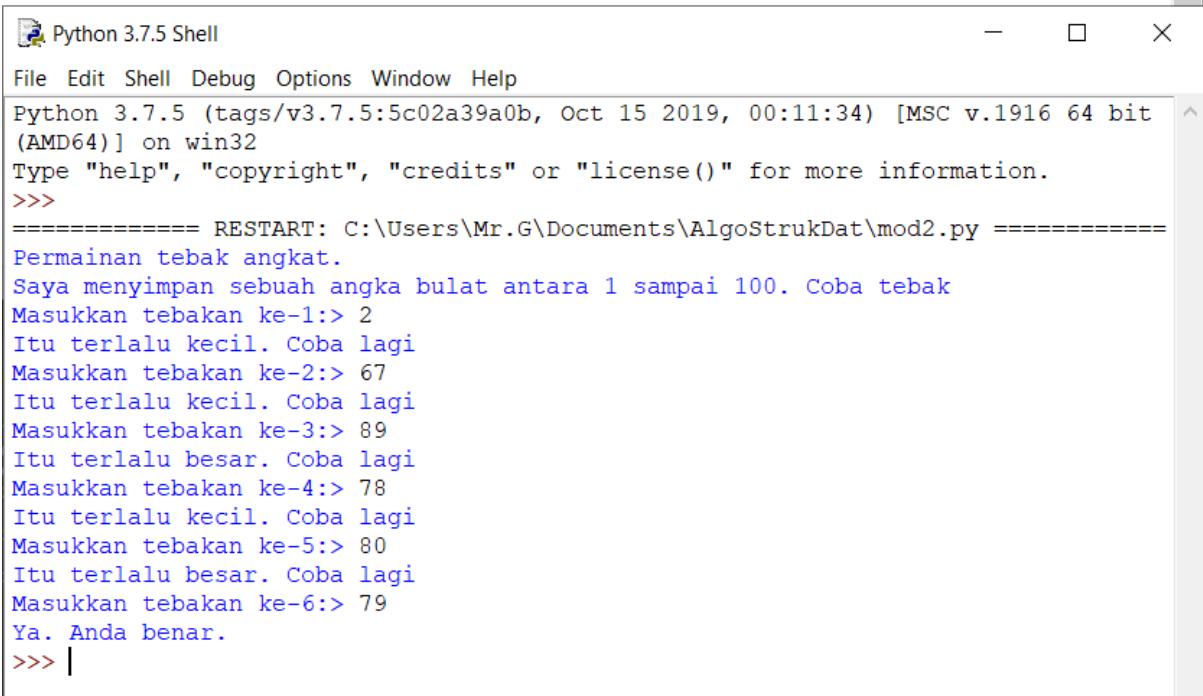
Masukkan Tahun : 2020

True

>>> |

## 12. Output file 12.py

```
#12
def tebak() :
    a = random.randrange(1,101)
    b = -1
    n = 0
    print("Permainan tebak angkat.")
    print("Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba tebak")
    while a != b :
        n = n + 1
        b= int(input("Masukkan tebakan ke-"+str(n)+":> "))
        if b < a :
            print("Itu terlalu kecil. Coba lagi")
        elif b > a :
            print("Itu terlalu besar. Coba lagi")
        else :
            print("Ya. Anda benar.")
            break
tebak()
```

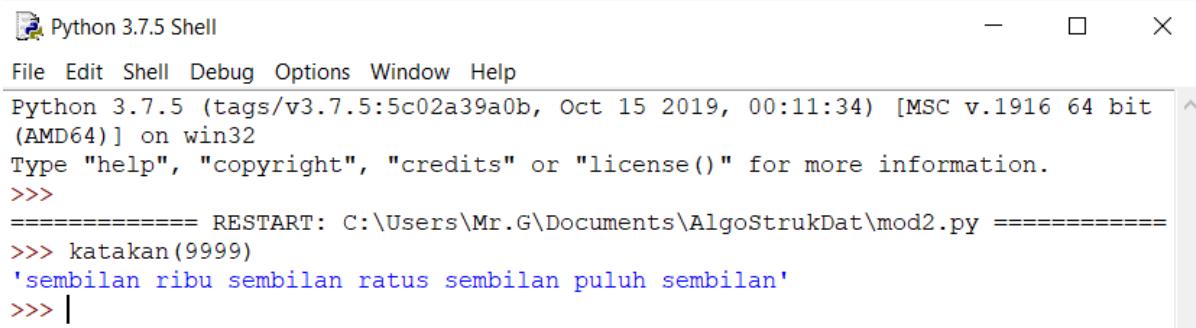


The screenshot shows the Python 3.7.5 Shell window. The title bar says "Python 3.7.5 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell area displays the following text:

```
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
Permainan tebak angkat.
Saya menyimpan sebuah angka bulat antara 1 sampai 100. Coba tebak
Masukkan tebakan ke-1:> 2
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-2:> 67
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-3:> 89
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-4:> 78
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-5:> 80
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-6:> 79
Ya. Anda benar.
>>> |
```

### 13. Output file 13.py

```
#13
def katakan(x):
    satuan = [' ', 'satu', 'dua', 'tiga', 'empat', 'lima', 'enam', 'tujuh', 'del
    hasil = ""
    if x <= 0:
        hasil += 'Bilangan Haruslah Positif\nndan Bilangan Asli'
    elif x < 12 :
        hasil += satuan[x]
    elif x < 20 :
        hasil += katakan(x-10) + " belas "
    elif x < 100:
        hasil += katakan(int(x/10)) + " puluh " + katakan(x%10)
    elif x < 200 :
        hasil += "seratus " + katakan(x-100)
    elif x < 1000 :
        hasil += katakan(int(x/100)) + " ratus " + katakan(x%100)
    elif x < 2000 :
        hasil += "seribu " + katakan(x-1000)
    elif x < 1000000 :
        hasil += katakan(int(x/1000)) + " ribu " + katakan(x%1000)
    elif x < 1000000000 :
        hasil += katakan(int(x/1000000)) + " juta " + katakan(x%1000000)
    elif x >= 1000000000 :
        hasil += katakan(int(x/1000000000)) + " milyar " + katakan(x%1000000000)
    return hasil
```

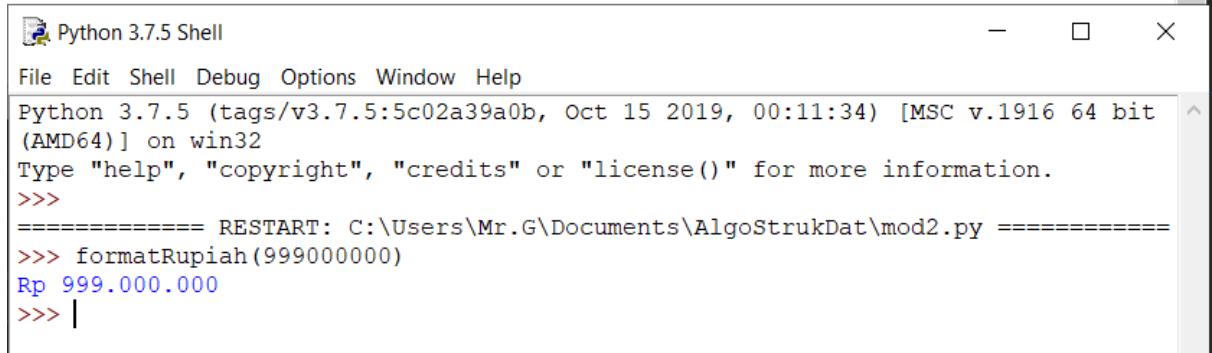


The screenshot shows a Python 3.7.5 Shell window. The title bar reads "Python 3.7.5 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the Python interpreter's prompt and the output of the code execution.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
>>> katakan(9999)
'sembilan ribu sembilan ratus sembilan puluh sembilan'
>>> |
```

#### 14. Output file 14.py

```
#14
def formatRupiah(a) :
    a = list(str(a))
    b = len(a)
    if b % 3 == 0 :
        b = int(b/3) - 1
    else :
        b = int(b/3)
    n = 0
    for i in range(b) :
        x = -3*(i+1)
        a.insert(int(x)+n, ".")
        n = n - 1
    a = "".join(a)
    print("Rp "+a)
```



The image shows a screenshot of the Python 3.7.5 Shell window. The title bar reads "Python 3.7.5 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the Python interpreter's prompt (>>>). It shows the version information: "Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32". Below this, it says "Type "help", "copyright", "credits" or "license()" for more information." A command is entered: ">>> formatRupiah(999000000)". The output is "Rp 999.000.000", followed by a new line indicator "|".

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\mod2.py =====
>>> formatRupiah(999000000)
Rp 999.000.000
>>> |
```

Nama : Guntur Jatmiko  
NIM : L200180039  
Kelas : B

## MODUL 2

1.

The screenshot shows a Windows desktop with two windows open. On the left is a code editor window titled 'no1.py - C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no1.py (3.7.5)'. It contains Python code for a class 'Pesan' with methods 'apakahTerkandung', 'hitungKonsonan', 'hitungVokal', and 'makan'. On the right is a Python 3.7.5 Shell window titled 'Python 3.7.5 Shell'. It shows the execution of the code, creating an instance 'p9' of the 'Pesan' class with the string 'indonesia adalah negri yang indah', and then checking if 'riya' is in the string, which returns False. It also shows the calculation of consonants ('Surakarta') and vowels ('Surakarta') in the string.

```
class Pesan(object):
    #a
    def __init__(self, kata):
        self.kata = kata

    def apakahTerkandung(self, yo):
        if yo in self.kata:
            return True
        else:
            return False

    #b
    def hitungKonsonan(self):
        vokal = 'AIUEOaiueo'
        v = 0
        for x in self.kata:
            if x in vokal:
                v+=1
        kon = len(self.kata) - v
        return kon

    #c
    def hitungVokal(self):
        vokal = 'AIUEOaiueo'
        v = 0
        for x in self.kata:
            if x in vokal:
                v+=1
        return v

    def makan(self, s):
        print('Saya baru saja makan', s)
        self.keadaan = 'kenyang'
        olahraga(self, k):
            print('Saya baru saja latihan', k)
            self.keadaan = 'lapar'
            mengalikanDenganDua(self, n):
                return n*2

class Mahasiswa(Manusia):
    """Class 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.kota = kota
        self.us = us

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

    def ambilNama(self):
        return self.nama

    def ambilNim(self):
        return self.NIM

    def ambilUangSaku(self):
        return self.us

    def tambahUangSaku(self, tambahUang):
        self.us = self.us + tambahUang

    def perbaruiKotaTinggal(self, kotabaru):
        self.kota = kotabaru

    def makan(self, s):
        """Metode ini menutupi metode 'makan'-nya class Manusia.
        Mahasiswa kalau sambil belajar ."""
        print("Saya baru saja makan", s, "sambil belajar")
        self.keadaan = 'kenyang'

m1 = Mahasiswa('Jamil', 234, 'Surakarta', 250000)
m2 = Mahasiswa('Andi', 365, 'Magelang', 275000)
m3 = Mahasiswa('Sri', 676, 'Yogyakarta', 240000)
```

```
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no1.py ====
>>> p9 = Pesan('indonesia adalah negri yang indah')
>>> p9.apakahTerkandung('riya')
False
>>> p9.apakahTerkandung('one')
True
>>> p10 = Pesan("Surakarta")
>>> p10.hitungKonsonan()
5
>>> p10.hitungVokal()
4
>>> |
```

2.

The screenshot shows a Windows desktop with two windows open. On the left is a code editor window titled 'no2.py - C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no2.py (3.7.5)'. It contains Python code for a class 'Mahasiswa' that inherits from 'Manusia'. It includes methods for eating, running, getting name, ID, and balance, and a constructor for initializing these variables. It also includes a method for changing city. On the right is a Python 3.7.5 Shell window titled 'Python 3.7.5 Shell'. It shows the creation of three instances of 'Mahasiswa' (m1, m2, m3) and their properties being modified.

```
class Pesan(object):
    #a
    def __init__(self, kata):
        self.kata = kata

    def apakahTerkandung(self, yo):
        if yo in self.kata:
            return True
        else:
            return False

    #b
    def hitungKonsonan(self):
        vokal = 'AIUEOaiueo'
        v = 0
        for x in self.kata:
            if x in vokal:
                v+=1
        kon = len(self.kata) - v
        return kon

    #c
    def hitungVokal(self):
        vokal = 'AIUEOaiueo'
        v = 0
        for x in self.kata:
            if x in vokal:
                v+=1
        return v

    def makan(self, s):
        print('Saya baru saja makan', s)
        self.keadaan = 'kenyang'
        olahraga(self, k):
            print('Saya baru saja latihan', k)
            self.keadaan = 'lapar'
            mengalikanDenganDua(self, n):
                return n*2

class Mahasiswa(Manusia):
    """Class 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.kota = kota
        self.us = us

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

    def ambilNama(self):
        return self.nama

    def ambilNim(self):
        return self.NIM

    def ambilUangSaku(self):
        return self.us

    def tambahUangSaku(self, tambahUang):
        self.us = self.us + tambahUang

    def perbaruiKotaTinggal(self, kotabaru):
        self.kota = kotabaru

    def makan(self, s):
        """Metode ini menutupi metode 'makan'-nya class Manusia.
        Mahasiswa kalau sambil belajar ."""
        print("Saya baru saja makan", s, "sambil belajar")
        self.keadaan = 'kenyang'

m1 = Mahasiswa('Jamil', 234, 'Surakarta', 250000)
m2 = Mahasiswa('Andi', 365, 'Magelang', 275000)
m3 = Mahasiswa('Sri', 676, 'Yogyakarta', 240000)
```

```
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no2.py ====
>>> m1.ambilKotaTinggal()
'Surakarta'
>>> m1.perbaruiKotaTinggal('Aceh')
>>> m1.ambilKotaTinggal()
'Aceh'
>>> m2.ambilUangSaku()
275000
>>> m2.tambahUangSaku(50000)
>>> m2.ambilUangSaku()
325000
>>> |
```

### 3.

The screenshot shows a Windows desktop with two windows open. On the left is a code editor window titled 'no3.py - C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no3.py (3.7.5)'. It contains Python code defining two classes: 'Manusia' and 'Mahasiswa'. The 'Manusia' class has methods for eating, sleeping, and calculating double. The 'Mahasiswa' class inherits from 'Manusia' and adds methods for taking notes and adding money. On the right is a Python 3.7.5 Shell window titled 'Python 3.7.5 Shell'. It shows a user interacting with the 'Mahasiswa' class, inputting student information and asking for a note. The shell also displays the current Python version and some help text.

```
no3.py - C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no3.py (3.7.5)
File Edit Format Run Options Window Help
class Manusia(object):
    """Class 'Manusia' dengan inisiasi 'nama'"""
    keadaan = 'lapar'

    def __init__(self, nama):
        self.nama = nama
    def ucapanSalam(self):
        print('Salam, namaku', self.nama)
    def makan(self, s):
        print('saya baru saja makan', s)
        self.keadaan = 'kenyang'
    def olahraga(self, k):
        print('saya baru saja latihan', k)
    def mengalikanDenganDua(self, n):
        return n*2

class Mahasiswa(Manusia):
    """Class 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.kota = kota
        self.us = us

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

    def ambilNama(self):
        return self.nama
    def ambilNim(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.us
    def tambahUangSaku(self, tambahUang):
        self.us = self.us + tambahUang
    def ambilKotaTinggal(self):
        return self.kota
    def perbaruiKotaTinggal(self, kotaBaru):
        self.kota = kotaBaru

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no3.py ====
Masukkan nama: Guntur J
Masukkan nim: L200180039
Masukkan kota tinggal: Sragen
Masukkan uang saku: 300000
Guntur J, NIM L200180039. Tinggal di Sragen. Uang saku Rp. 300000 tiap bulannya.
>>>
```

### 4.

The screenshot shows a Windows desktop with two windows open. On the left is a code editor window titled 'no4.py - C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no4.py (3.7.5)'. It contains Python code defining a 'Mahasiswa' class that inherits from 'Manusia'. The 'Mahasiswa' class adds methods for eating, taking notes, and adding money. It also includes a list of subjects. On the right is a Python 3.7.5 Shell window titled 'Python 3.7.5 Shell'. It shows a user creating three student objects and querying their lists of subjects.

```
no4.py - C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no4.py (3.7.5)
File Edit Format Run Options Window Help
self.keadaan = 'lapar'
def mengalikanDenganDua(self, n):
    return n*2

class Mahasiswa(Manusia):
    """Class yang dibangun dari class Manusia"""
    def __init__(self, nama, NIM, kota, us, lk = []):
        """Metode inisiasi ini menutupi metode inisiasi di class Manusia"""
        self.nama = nama
        self.NIM = NIM
        self.kota = kota
        self.us = us
        self.listkuliah = lk

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

    def ambilNama(self):
        return self.nama
    def ambilNim(self):
        return self.NIM
    def ambilUangSaku(self):
        return self.us
    def tambahUangSaku(self, tambahUang):
        self.us = self.us + tambahUang
    def ambilKotaTinggal(self):
        return self.kota
    def perbaruiKotaTinggal(self, kotaBaru):
        self.kota = kotaBaru
    def makan(self, s):
        """Metode ini menutupi metode 'makan'-nya class Manusia.
        Mahasiswa kali ini sambil belajar."""
        print('saya baru saja makan', s, "sambil belajar")
        self.keadaan = 'kenyang'
    def ambilKuliah(self, ambil):
        self.listkuliah.append(ambil)

m1 = Mahasiswa('Roni', 121, 'Surakarta', 300000)
m2 = Mahasiswa('Budi', 134, 'Karanganyar', 450000)
m3 = Mahasiswa('Paijo', 132, 'Sragen', 350000)
```

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no4.py ====
>>> m2.listkuliah
[]
>>> m2.ambilKuliah('ALGOSTRUK')
>>> m2.listkuliah
['ALGOSTRUK']
>>> m2.ambilKuliah('ALGOPRO')
>>> m2.listkuliah
['ALGOSTRUK', 'ALGOPRO']
>>> |
```

## 5.

The screenshot shows a Windows desktop with two windows open. On the left is a code editor window titled 'no5.py - C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no5.py (3.7.5)'. It contains Python code for a 'Manusia' class with methods like \_\_init\_\_, \_\_str\_\_, ambilNama, tambahUang, ambilUang, makan, and tambahUng. It also includes code to create three instances (m1, m2, m3) and print their details. On the right is a Python 3.7.5 Shell window titled 'Python 3.7.5 Shell'. It shows the execution of the script, including the creation of a list 'm2.listkuliah' containing 'ALGOSTRUK', 'ALGOPRO', and 'ALGOFPRO', and the removal of 'ALGOPRO' from the list.

```
def __init__(self, nama, NIM, kota, us, lk = []):
    """Metode inisiasi ini menutupi metode inisiasi di class Manusia"""
    self.nama = nama
    self.nim = NIM
    self.kota = kota
    self.us = us
    self.listkuliah = lk

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

def ambilNama(self):
    return self.nama

def ambilNim(self):
    return self.nim

def ambilKota(self):
    return self.kota

def tambahUangSaku(self, tambahUang):
    self.us = self.us + tambahUang

def ambilKotaTinggal(self):
    return self.kota

def perbaruiStatistikTinggal(self, kotabaru):
    self.kota = kotabaru

def makan(self, s):
    """Metode ini menutupi metode 'makan'-nya class Manusia.
    Mahasiswa kalau sambil belajar ,"""
    print("saya baru saja makan", s, "sambil belajar")
    self.keadaan = 'kenyang'

def ambilKuliah(self, ambil):
    self.listkuliah.append(ambil)

def hapusListKuliah(self, hapus):
    for x in self.listkuliah:
        if hapus in self.listkuliah:
            self.listkuliah.remove(hapus)
        else:
            print("Maaf mata kuliah tidak ada dalam list mata kuliah yang diambil")

m1 = Mahasiswa('Roni', 121, 'Surakarta', 300000)
m2 = Mahasiswa('Budi', 134, 'Karanganyar', 450000)
m3 = Mahasiswa('Paij', 132, 'Slragen', 350000)
```

```
[ALGOSTRUK]
>>> m2.listkuliah('ALGOSTRUK')
Traceback (most recent call last):
  File <ipython-input-1>, line 1, in <module>
    m2.listkuliah('ALGOSTRUK')
TypeError: 'list' object is not callable
>>> m2.ambilKuliah('ALGOSTRUK')
>>> m2.listkuliah
['ALGOSTRUK']
>>> m2.ambilKuliah('ALGOPRO')
>>> m2.listkuliah
['ALGOPRO']
>>> m2.hapusListKuliah('ALGOPRO')
>>> m2.listkuliah
['ALGOSTRUK']
>>>
```

## 6.

The screenshot shows a Windows desktop with two windows open. On the left is a code editor window titled 'no6.py - C:\Users\Mr.G\Documents\AlgoStrukDat\Modul-2-L200170102\no6.py (3.7.5)'. It contains Python code for a 'Manusia' class with methods like \_\_init\_\_, \_\_str\_\_, ucapanSalam, olahraga, mengalikanDenganDua, and SiswaMA. It also includes code to create instances (g, q) and print their details. On the right is a Python 3.7.5 Shell window titled 'Python 3.7.5 Shell'. It shows the execution of the script, including the creation of instances 'g' and 'q', and printing their details.

```
from datetime import date
class Manusia(object):
    """Class 'Manusia' dengan inisiasi 'nama'"""
    keadaan = 'lapar'

    def __init__(self, nama):
        self.nama = nama
    def ucapanSalam(self):
        print('Halo', self.nama)
    def __str__(self):
        print('saya baru saja makan', self.keadaan)
    def keadaan = 'kenyang'
    def olahraga(self, k):
        print('saya baru saja latihan', k)
    def keadaan = 'lapar'
    def mengalikanDenganDua(self, n):
        return n*2

class SiswaMA(Manusia):
    def __init__(self, nama, NIS, umur, us):
        """Metode inisiasi ini menutupi metode inisiasi di class Manusia"""
        self.nama = nama
        self.nis = NIS
        self.umur = umur
        self.us = us

    def __str__(self):
        s = self.nama +', '+str(self.nis) \
            +'. Berumur '+str(self.umur) \
            +'!. Uang saku Rp. '+str(self.us) \
            +' tiap harinya.'
        return s

    def tahunLahir(self):
        thnskr = date.today().year
        tl = thnskr - self.umur
        return tl
```

```
>>> g = SiswaMA('GunturJ', '39', '17', 9999999999)
>>> print(g)
GunturJ, NIS 39. Berumur 17. Uang saku Rp. 9999999999 tiap harinya.
>>> |
```

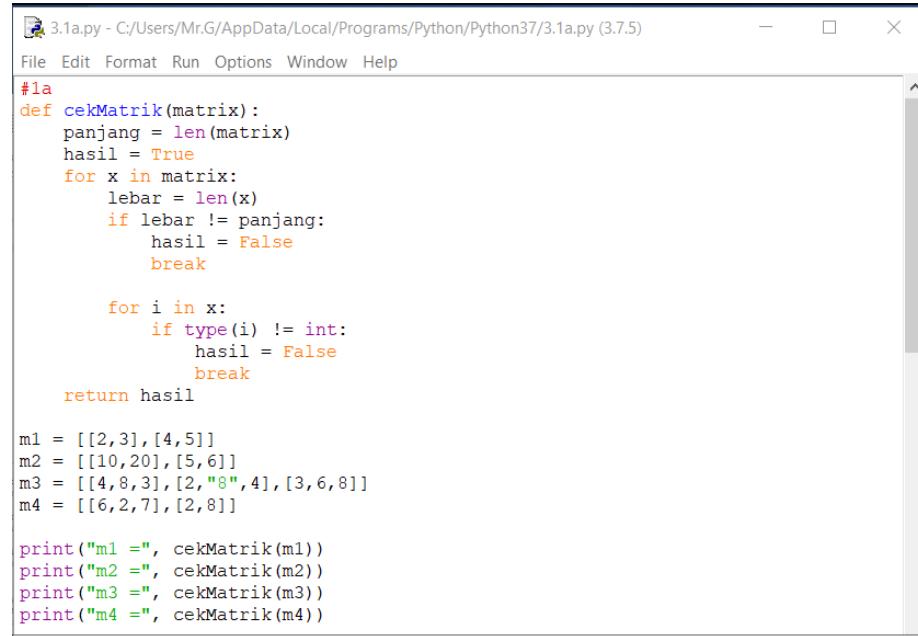
## 7.

Metode/state itu berasal dari class Mahasiswa

Nama : Guntur Jatmiko  
NIM : L200180039  
Kelas : B

## TUGAS

### Latihan

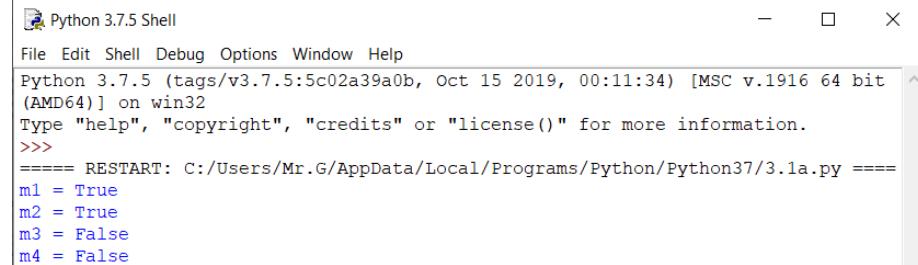


```
#1a
def cekMatrik(matrix):
    panjang = len(matrix)
    hasil = True
    for x in matrix:
        lebar = len(x)
        if lebar != panjang:
            hasil = False
            break

        for i in x:
            if type(i) != int:
                hasil = False
                break
    return hasil

m1 = [[2,3],[4,5]]
m2 = [[10,20],[5,6]]
m3 = [[4,8,3],[2,"8",4],[3,6,8]]
m4 = [[6,2,7],[2,8]]

print("m1 =", cekMatrik(m1))
print("m2 =", cekMatrik(m2))
print("m3 =", cekMatrik(m3))
print("m4 =", cekMatrik(m4))
```



```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/3.1a.py =====
m1 = True
m2 = True
m3 = False
m4 = False
```

## TUGAS

# 1.

The image shows three windows from a Python development environment:

- 3.1a.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/3.1a.py (3.7.5)**: This window contains the source code for a script named 3.1a.py. It defines a function `cekMatrik` that checks if a given matrix is square and all elements are integers. It also prints the results for four matrices: m1, m2, m3, and m4.
- Python 3.7.5 Shell**: This window shows the output of running the script. It prints the value of `m1` as `True`, `m2` as `True`, `m3` as `False`, and `m4` as `False`. This indicates that `m1` and `m2` are square matrices with integer elements, while `m3` and `m4` are not.
- 3.1a.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/3.1a.py (3.7.5)**: This window shows the same code as the first one, but it includes additional code at the bottom for a function `Ukuran` which returns the size of a matrix as a string.

```
#1a
def cekMatrik(matrix):
    panjang = len(matrix)
    hasil = True
    for x in matrix:
        lebar = len(x)
        if lebar != panjang:
            hasil = False
            break

    for i in x:
        if type(i) != int:
            hasil = False
            break
    return hasil

m1 = [[2,3],[4,5]]
m2 = [[10,20],[5,6]]
m3 = [[4,8,3],[2,"8",4],[3,6,8]]
m4 = [[6,2,7],[2,8]]

print("m1 =", cekMatrik(m1))
print("m2 =", cekMatrik(m2))
print("m3 =", cekMatrik(m3))
print("m4 =", cekMatrik(m4))

#1b
def Ukuran(matrix):
    return ("Ukuran matrix = "+str(len(matrix))+" x "+str(len(matrix[0])))

m1 = [[2,3],[4,5]]
m2 = [[10,20],[5,6]]

print(Ukuran(m1))
print(Ukuran(m2))
```

```

#1c
a = [[1,2],[3,4]]
b = [[7,2],[1,4]]
c = [[1,"a","b"],[3,4,"c"]]
d = [[2,1],[3,4],[6,5]]
e = [[3,2,1],[5,4,3]]
f = [[1,2,3],[4,5,6],[1,5,6]]

def jumlah(n,m):
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    xy = [[0 for j in range(x)] for i in range(y)]

    z = 0
    if(len(n)==len(m)) :
        for i in range(len(n)):
            if(len(n[i]) == len(m[i])):
                z+=1
    if(z==len(n) and z==len(m)) :
        print("Ukuran sama")
        for i in range(len(n)):
            for j in range(len(n[i])):
                xy[i][j] = n[i][j] + m[i][j]
        print(xy)
    else:
        print("Ukuran beda")

jumlah(a,b)
jumlah(a,d)

```

Ukuran sama  
[[8, 4], [4, 8]]  
Ukuran beda

3.1a.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/3.1a.py (3.7.5)

File Edit Format Run Options Window Help

```

#1d
def kali(n,m):
    aa = 0
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    v,w = 0,0
    for i in range(len(m)):
        v+=1
        w = len(m[i])

    if(y==v) :
        print("Dapat Dikalikan")
        vwxy = [[0 for j in range(w)] for i in range(x)]
        for i in range(len(n)):
            for j in range(len(m[0])):
                for k in range(len(m)):
                    #print(n[i][k], m[k][j])
                    vwxy[i][j] += n[i][k] * m[k][j]
        print(vwxy)

    else:
        print("Tidak memenuhi syarat")

zz = [[1,2,3],[1,2,3]]
zx = [[1],[2],[3]]
kali(zz,zx)
kali(a,b)
kali(a,e)
kali(a,zx)

```

Dapat Dikalikan  
[[14], [14]]  
Dapat Dikalikan  
[[9, 10], [25, 22]]  
Dapat Dikalikan  
[[13, 10, 7], [29, 22, 15]]  
Tidak memenuhi syarat

## 2.

```
#2A
def buatNol(m, n):
    """Menggunakan dua input"""
    matrix = [[0 for x in range(m)] for i in range(n)]
    print(matrix)

def buatNol2(m):
    """Menggunakan satu input"""
    n = m
    matrix = [[0 for x in range(m)] for i in range(n)]
    print(matrix)

#2B
def buatIdentitas(m):
    n = m
    matrix = [[1 if j == i else 0 for j in range(m)] for i in range(n)]
    print(matrix)

#2
buatNol(3,3)
buatNol2(3)
buatIdentitas(4)
```

The screenshot shows the Python 3.7.5 Shell window. The title bar says "Python 3.7.5 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main area displays the Python interpreter's prompt (>>>) and the output of the executed code. The output shows three matrices:

```
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
```

## 3.

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def pushAw(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
    def pushAk(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)
    return self.head
    def insert(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
    return self.head
    def deleteNode(self, position):
        if self.head == None:
            return
        temp = self.head
        if position == 0:
            self.head = temp.next
            temp = None
        else:
            prev = None
            current = self.head
            current_pos = 0
            while(current_pos < position) and current.next:
                prev = current
                current = current.next
                current_pos +=1
            if current == None:
                return
            prev.next = current.next
            current = None
```

```

    if position == 0:
        self.head = temp.next
        temp = None
        return
    for i in range(position - 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 search(self, x):
    current = self.head
    while current != None:
        if current.data == x:
            return "True"
        current = current.next
    return "False"
def display(self):
    current = self.head
    while current is not None:
        print(current.data, end = ' ')
        current = current.next

llist = LinkedList()
llist.pushAw(21)
llist.pushAw(22)
llist.pushAw(12)
llist.pushAw(14)
llist.pushAw(2)
llist.pushAw(19)
llist.pushAk(9)
llist.deleteNode(0)
llist.insert(1,6)
print(llist.search(21))
print(llist.search(29))
llist.display()

```

```

True
False
2 14 12 22 21 1 9
>>>

```

#### 4.

```

class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
class DoublyLinkedList:
    def __init__(self):
        self.head = None
    def awal(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
        return
    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
llist = DoublyLinkedList()
llist.awal(7)
llist.awal(1)
llist.akhir(6)
llist.akhir(4)
llist.printList(llist.head)

```

Menambah pada Awal 7  
Menambah pada Awal 1  
Menambah pada Akhir 6  
Menambah pada Akhir 4

Dari Depan :

1  
7  
6  
4

Dari Belakang :

4  
6  
7  
1

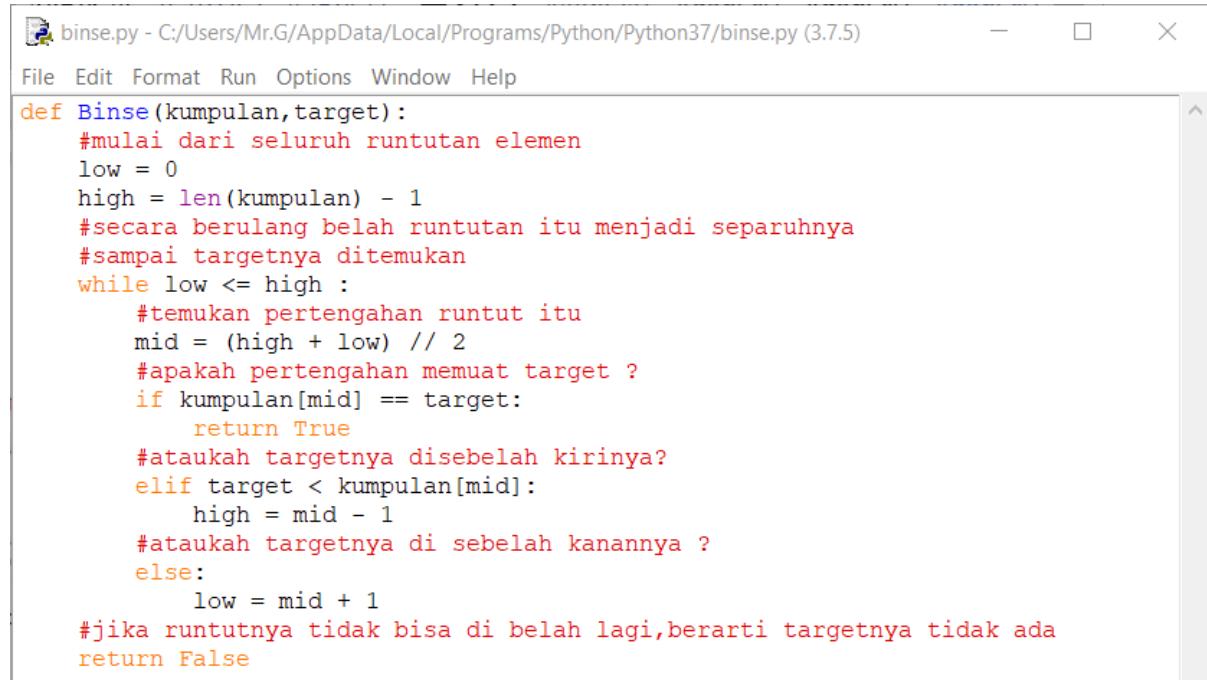
>>> |

Nama : Guntur Jatmiko  
NIM : L200180039  
Kelas : B

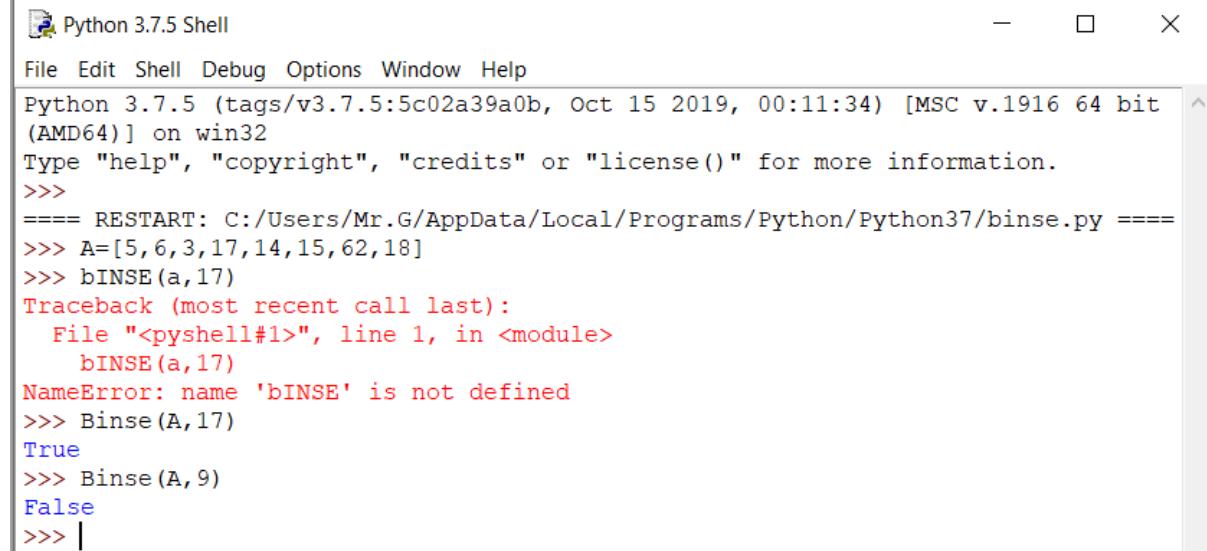
## Modul 4

### Latihan

#### Binse

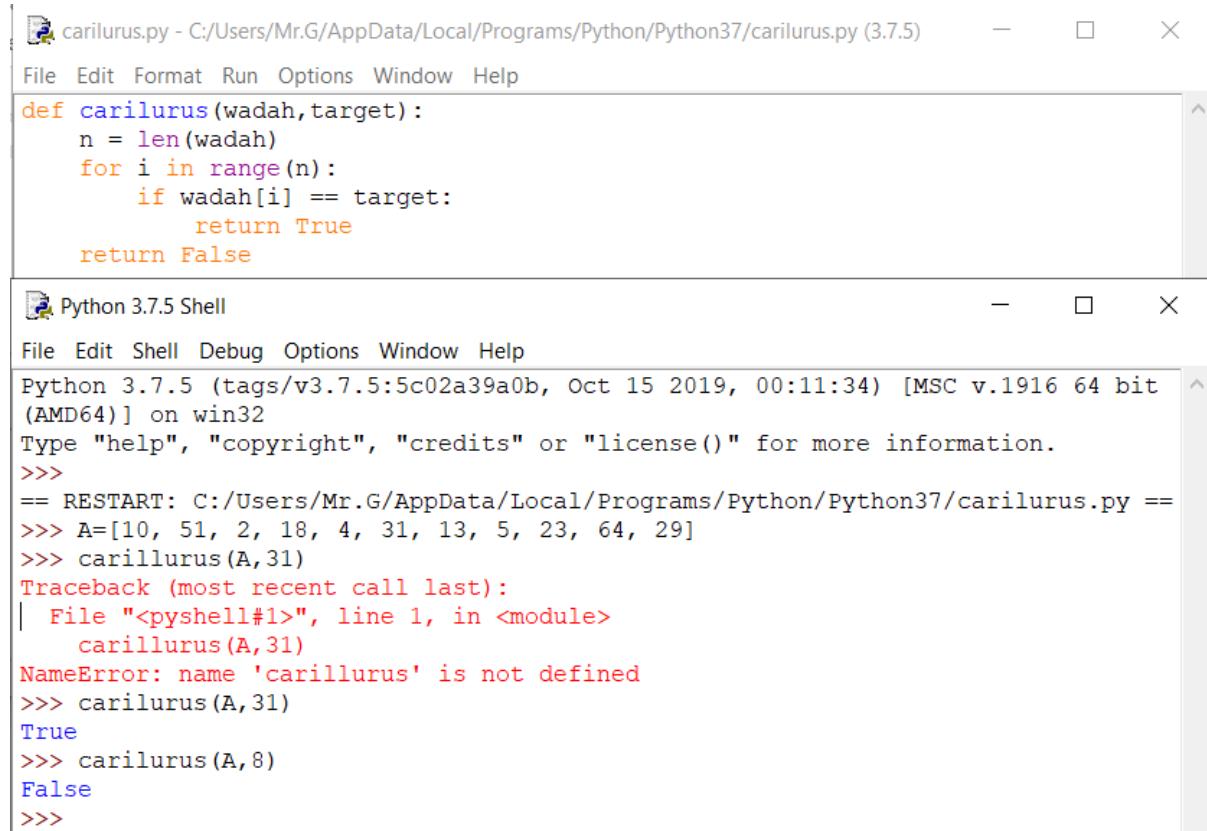


```
binse.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/binse.py (3.7.5)
File Edit Format Run Options Window Help
def Binse(kumpulan,target):
    #mulai dari seluruh runtutan elemen
    low = 0
    high = len(kumpulan) - 1
    #secara berulang belah runtutan itu menjadi separuhnya
    #sampai targetnya ditemukan
    while low <= high :
        #temukan pertengahan runtut itu
        mid = (high + low) // 2
        #apakah pertengahan memuat target ?
        if kumpulan[mid] == target:
            return True
        #ataukah targetnya disebelah kirinya?
        elif target < kumpulan[mid]:
            high = mid - 1
        #ataukah targetnya di sebelah kanannya ?
        else:
            low = mid + 1
    #jika runtutnya tidak bisa di belah lagi,berarti targetnya tidak ada
    return False
```



```
Python 3.7.5 Shell
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/binse.py =====
>>> A=[5,6,3,17,14,15,62,18]
>>> bINSE(a,17)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    bINSE(a,17)
NameError: name 'bINSE' is not defined
>>> Binse(A,17)
True
>>> Binse(A, 9)
False
>>> |
```

## Carilurus



The image shows a Windows desktop environment with two windows open. The top window is titled 'carilurus.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/carilurus.py (3.7.5)' and contains Python code. The bottom window is titled 'Python 3.7.5 Shell' and contains a command-line session.

```
def carilurus(wadah,target):
    n = len(wadah)
    for i in range(n):
        if wadah[i] == target:
            return True
    return False
```

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/carilurus.py ==
>>> A=[10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
>>> carillurus(A,31)
Traceback (most recent call last):
| File "<pyshell#1>", line 1, in <module>
|     carillurus(A,31)
NameError: name 'carillurus' is not defined
>>> carilurus(A,31)
True
>>> carilurus(A,8)
False
>>>
```

## Mhstarget



A screenshot of a Windows-style code editor window titled "mhstarget.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/mhstarget.py (3.7.5)". The window shows Python code with syntax highlighting. The code defines two classes: "Manusia" and "Mahasiswa". The "Manusia" class has methods for eating, exercising, and calculating squares. The "Mahasiswa" class inherits from "Manusia" and adds attributes for NIM and address, and overrides the "makan" method to include studying. At the bottom, two instances of "Mahasiswa" are created.

```
class Manusia(object):
    """kelas 'manusia' dengan inisiasi 'nama' """
    keadaan = 'lapar'
    def __init__(self,nama):
        self.nama = nama
    def ucapkansalam(self):
        print ("salam, namaku",self.nama)
    def makan(self,s):
        print("saya baru saja makan",s)
        self.keadaan = "kenyang"
    def olahraga (self,k):
        print("saya baru saja latihan",k)
        self.keadaan = 'lapar'
    def mengalikandengandua(self,n):
        return n*2

##p1 = Manusia("Fatimah")
##p1.ucapkansalam()

class Mahasiswa(Manusia):
    """class mahasiswa yang dibangun dari kelas manusia"""
    def __init__(self,nama,NIM,kota,us):
        """metode inisiasi ini menutupi metode inisiasi di kelas manusia"""
        self.nama = nama
        self.NIM = NIM
        self.kotatinggal =kota
        self.uangsaku = us
    def __str__(self):
        s = self.nama +",NIM"+ str(self.NIM) \
            +",tinggaldi" + self.kotatinggal \
            +", uangsaku 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):
        """metode ini menutupi metode 'makan' nya classs manusia.
        mahasiswa kalau makan sambil belajar."""
        print("saya baru saja makan", s,"sambil belajar")
        self.keadaan = "kenyang"

m1 = Mahasiswa("Jamil",234,"surakarta",250000)
m2 = Mahasiswa("andi",365,"magelang",375000)
```

Ln: 74 Col: 0

mhstarget.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/mhstarget.py (3.7.5) — □ ×

File Edit Format Run Options Window Help

```
m2 = Mahasiswa("andi",365,"magelang",375000)
m3 = Mahasiswa ("Sri",676,"yogyakarta",240000)

class MhsTIF(Mahasiswa):
    """class MhsTIF yang dibangun dari class mahasiswa"""
    def katanypy(self):
        print("python is cool")

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

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

target = "Klaten"
for i in Daftar:
    if i.kotatinggal == target:
        print(i.nama + " tinggal di "+target)
```

Python 3.7.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/mhstarget.py ==
Deni tinggal di Klaten
Janto tinggal di Klaten
>>> |
```

Ln: 7 Col: 4

## Terkecil

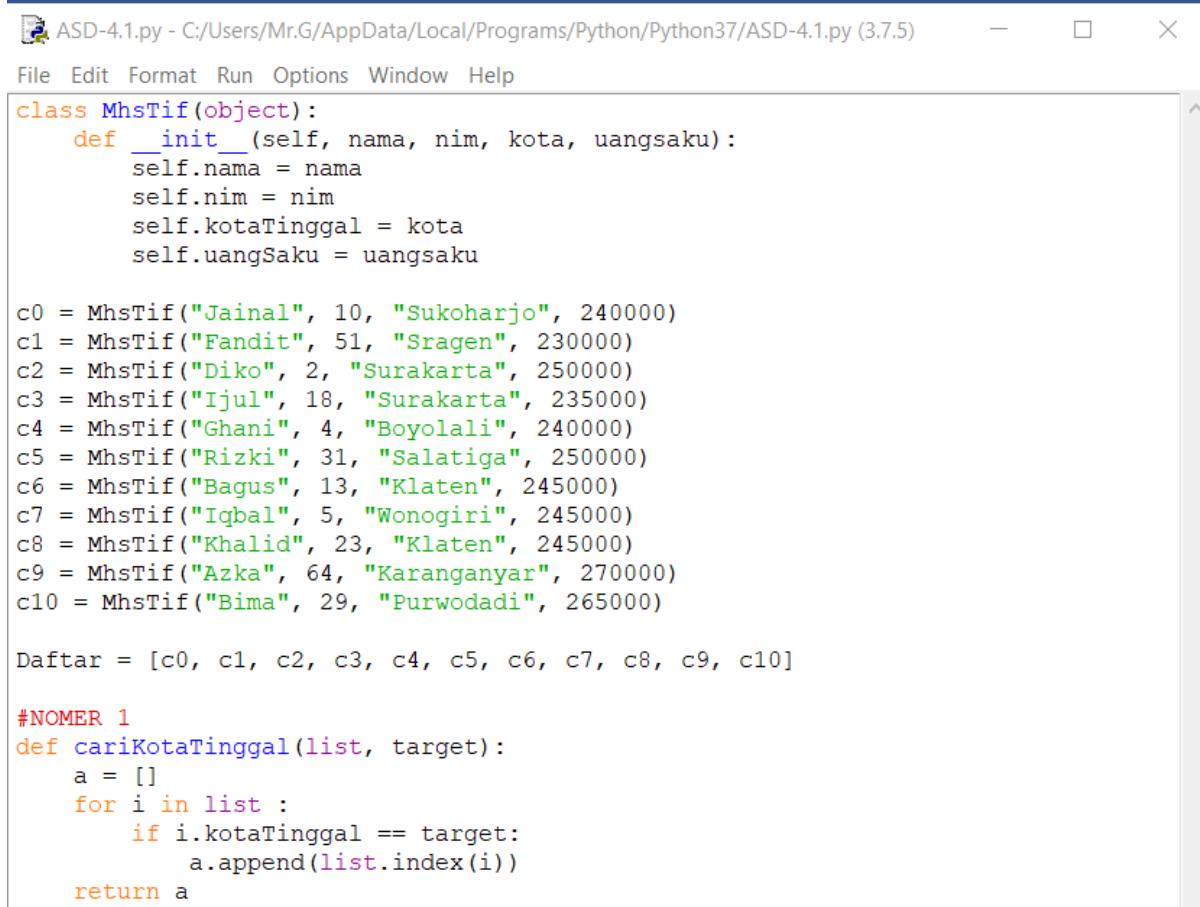
The screenshot shows two windows from a Python development environment. The top window is titled 'terkecil' and contains the Python code for finding the smallest element in a list. The bottom window is titled 'Python 3.7.5 Shell' and shows the execution of this code.

```
terkecil.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/terkecil.py (3.7.5)
File Edit Format Run Options Window Help
def cariTerkecil(kumpulan):
    n=len(kumpulan)
    terkecil=kumpulan[0]
    for i in range(1,n):
        if kumpulan[i] < terkecil:
            terkecil=kumpulan[i]
    return terkecil

Python 3.7.5 Shell
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/terkecil.py ==
>>> a=[9,5,8,3,6,3,6,1]
>>> cariTerkecil(a)
1
>>> |
```

TUGAS

1.



```
ASD-4.1.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.1.py (3.7.5) - X

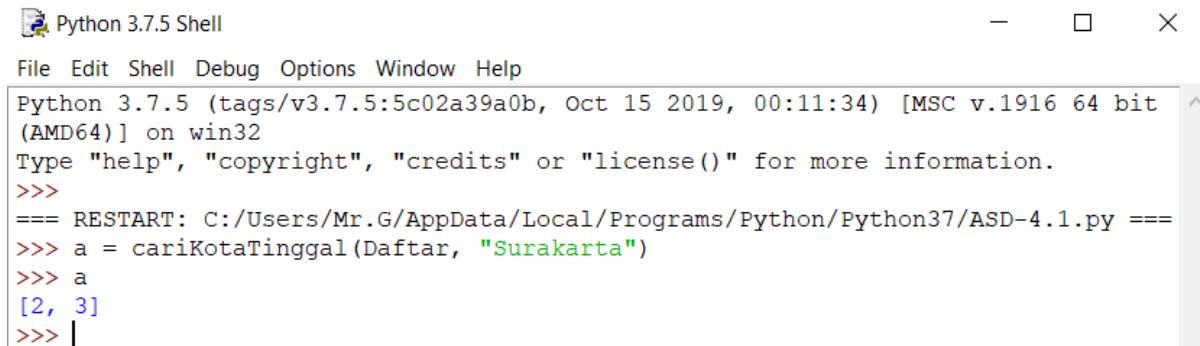
File Edit Format Run Options Window Help

class MhsTif(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

c0 = MhsTif("Jainal", 10, "Sukoharjo", 240000)
c1 = MhsTif("Fandit", 51, "Sragen", 230000)
c2 = MhsTif("Diko", 2, "Surakarta", 250000)
c3 = MhsTif("Ijul", 18, "Surakarta", 235000)
c4 = MhsTif("Ghani", 4, "Boyolali", 240000)
c5 = MhsTif("Rizki", 31, "Salatiga", 250000)
c6 = MhsTif("Bagus", 13, "Klaten", 245000)
c7 = MhsTif("Iqbal", 5, "Wonogiri", 245000)
c8 = MhsTif("Khalid", 23, "Klaten", 245000)
c9 = MhsTif("Azka", 64, "Karanganyar", 270000)
c10 = MhsTif("Bima", 29, "Purwodadi", 265000)

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

#NOMER 1
def cariKotaTinggal(list, target):
    a = []
    for i in list :
        if i.kotaTinggal == target:
            a.append(list.index(i))
    return a
```



```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.1.py ==
>>> a = cariKotaTinggal(Daftar, "Surakarta")
>>> a
[2, 3]
>>> |
```

2.

ASD-4.2.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.2.py (3.7.5)

File Edit Format Run Options Window Help

```
class MhsTif(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

c0 = MhsTif("Jainal", 10, "Sukoharjo", 240000)
c1 = MhsTif("Fandit", 51, "Sragen", 230000)
c2 = MhsTif("Diko", 2, "Surakarta", 250000)
c3 = MhsTif("Ijul", 18, "Surakarta", 235000)
c4 = MhsTif("Ghani", 4, "Boyolali", 240000)
c5 = MhsTif("Rizki", 31, "Salatiga", 250000)
c6 = MhsTif("Bagus", 13, "Klaten", 245000)
c7 = MhsTif("Iqbal", 5, "Wonogiri", 245000)
c8 = MhsTif("Khalid", 23, "Klaten", 245000)
c9 = MhsTif("Azka", 64, "Karanganyar", 270000)
c10 = MhsTif("Bima", 29, "Purwodadi", 265000)

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

#NOMER 2
def cariUangSakuTerkecil(list):
    temp = list[0].uangSaku
    for i in list[1:]:
        if i.uangSaku < temp:
            temp = i.uangSaku
    return temp
```

Python 3.7.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916
(AMD64) ] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.2
>>> a = cariUangSakuTerkecil(Daftar)
>>> a
230000
>>> |
```

3.

```
ASD-4.3.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.3.py (3.7.5) — File Edit Format Run Options Window Help

class MhsTif(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

c0 = MhsTif("Jainal", 10, "Sukoharjo", 240000)
c1 = MhsTif("Fandit", 51, "Sragen", 230000)
c2 = MhsTif("Diko", 2, "Surakarta", 250000)
c3 = MhsTif("Ijul", 18, "Surakarta", 235000)
c4 = MhsTif("Ghani", 4, "Boyolali", 240000)
c5 = MhsTif("Rizki", 31, "Salatiga", 250000)
c6 = MhsTif("Bagus", 13, "Klaten", 245000)
c7 = MhsTif("Iqbal", 5, "Wonogiri", 245000)
c8 = MhsTif("Khalid", 23, "Klaten", 245000)
c9 = MhsTif("Azka", 64, "Karanganyar", 270000)
c10 = MhsTif("Bima", 29, "Purwodadi", 265000)

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

#NOMER 3
def uangSakuterkecil():
    a = Daftar[0].uangSaku
    x = []
    for i in range(len(Daftar)):
        if a > Daftar[i].uangSaku:
            a = Daftar[i].uangSaku
    for i in range(len(Daftar)):
        if Daftar[i].uangSaku == a:
            x.append(Daftar[i].nama)
    return x
```

```
Python 3.7.5 Shell — File Edit Shell Debug Options Window Help

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.3.py ====
>>> uangSakuterkecil()
['Fandit']
>>> |
```

4.

```
ASD-4.4.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.4.py (3.7.5)

File Edit Format Run Options Window Help

class MhsTif(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

c0 = MhsTif("Jainal", 10, "Sukoharjo", 240000)
c1 = MhsTif("Fandit", 51, "Sragen", 230000)
c2 = MhsTif("Diko", 2, "Surakarta", 250000)
c3 = MhsTif("Ijul", 18, "Surakarta", 235000)
c4 = MhsTif("Ghani", 4, "Boyolali", 240000)
c5 = MhsTif("Rizki", 31, "Salatiga", 250000)
c6 = MhsTif("Bagus", 13, "Klaten", 245000)
c7 = MhsTif("Iqbal", 5, "Wonogiri", 245000)
c8 = MhsTif("Khalid", 23, "Klaten", 245000)
c9 = MhsTif("Azka", 64, "Karanganyar", 270000)
c10 = MhsTif("Bima", 29, "Purwodadi", 265000)

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

#NOMER 4
def uangSakukurang25k():
    x = []
    for i in range(len(Daftar)):

        if Daftar[i].uangSaku < 250000:
            x.append(Daftar[i].nama)
    return x
```

```
Python 3.7.5 Shell

File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.4.py ====
>>> uangSakukurang25k()
['Jainal', 'Fandit', 'Ijul', 'Ghani', 'Bagus', 'Iqbal', 'Khalid']
>>> |
```

## 5.

The screenshot shows two side-by-side Python shells. The left shell is a standard Python 3.7.5 Shell, and the right shell is titled 'ASD-4.5.py'. Both shells are running the same code, which defines a class 'MhsTif' and uses it to create a linked list of 10 student objects. The objects are stored in a list 'Daftar' and have attributes: nama, nim, kota, and uangSaku. The right shell also contains a 'cari' function that searches for a student by name and prints whether they are found in the list or not. The output shows the creation of 10 students and a search for 'Data 45' which is not found.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==> RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.5.py ===
>>> a = node(17)
>>> draf = a
>>> a.next = node(19)
>>> a = a.next
>>> a.next = node(11)
>>> a = a.next
>>> a.next = node(45)
>>> a = a.next
>>> draf.cari(45)
Data 45 tidak ada dalam Linked List
>>> draf.cari(25)
Data 25 tidak ada dalam Linked List
>>> |
```

```
ASD-4.5.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.5.py (3.7.5)
File Edit Format Run Options Window Help
class MhsTif(object):
    def __init__(self, nama, nim, kota, uangSaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangSaku

c0 = MhsTif("Jainal", 10, "Sukoharjo", 240000)
c1 = MhsTif("Pandit", 51, "Sragen", 230000)
c2 = MhsTif("Diko", 2, "Surakarta", 250000)
c3 = MhsTif("Ijul", 18, "Surakarta", 235000)
c4 = MhsTif("Ghani", 4, "Boyolali", 240000)
c5 = MhsTif("Rizki", 31, "Salatiga", 250000)
c6 = MhsTif("Bagus", 13, "Klaten", 245000)
c7 = MhsTif("Iqbal", 5, "Wonogiri", 245000)
c8 = MhsTif("Khalid", 23, "Klaten", 245000)
c9 = MhsTif("Azka", 64, "Karanganyar", 270000)
c10 = MhsTif("Bima", 29, "Purwodadi", 265000)

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]
#NOMER 5
class node(object):
    def __init__(self, data, next = None):
        self.data = data
        self.next = next

    def cari(self, dicari):
        cur = self
        while cur != None:
            if cur.data == dicari:
                cur = cur.next
            else:
                print ("Data", dicari, "ada dalam Linked List")
                break
            elif cur.next == None:
                print ("Data", dicari, "tidak ada dalam Linked List")
                break
        return
```

## 6.

The screenshot shows two side-by-side Python shells. The left shell is a standard Python 3.7.5 Shell, and the right shell is titled 'ASD-4.6.py'. Both shells are running the same code, which defines a class 'MhsTif' and uses it to create a linked list of 10 student objects. The objects are stored in a list 'Daftar' and have attributes: nama, nim, kota, and uangSaku. The right shell also contains a 'binSe' function that performs a binary search on the list based on the 'nim' attribute. The output shows the creation of 10 students and a search for 'kumpulan' which is not defined, resulting in a NameError.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==> RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.6.py ===
>>>
==> RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.6.py ===
>>> kumpulan = [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>> (binSe(kumpulan, 5))
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    (binSe(kumpulan, 5))
NameError: name 'kumpulan' is not defined
>>> (binSe(kumpulan, 5))
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    (binSe(kumpulan, 5))
NameError: name 'binSe' is not defined
>>> (binSe(kumpulan, 5))
>>> |
```

```
ASD-4.6.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.6.py (3.7.5)
File Edit Format Run Options Window Help
class MhsTif(object):
    def __init__(self, nama, nim, kota, uangSaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangSaku

c0 = MhsTif("Jainal", 10, "Sukoharjo", 240000)
c1 = MhsTif("Pandit", 51, "Sragen", 230000)
c2 = MhsTif("Diko", 2, "Surakarta", 250000)
c3 = MhsTif("Ijul", 18, "Surakarta", 235000)
c4 = MhsTif("Ghani", 4, "Boyolali", 240000)
c5 = MhsTif("Rizki", 31, "Salatiga", 250000)
c6 = MhsTif("Bagus", 13, "Klaten", 245000)
c7 = MhsTif("Iqbal", 5, "Wonogiri", 245000)
c8 = MhsTif("Khalid", 23, "Klaten", 245000)
c9 = MhsTif("Azka", 64, "Karanganyar", 270000)
c10 = MhsTif("Bima", 29, "Purwodadi", 265000)

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]
#NOMER 6
def binSe(kumpulan, target):
    low = 0
    high = len(kumpulan)-1
    while low <= high:
        mid = (high+low)//2
        if kumpulan[mid] == target:
            return mid
        elif target < kumpulan[mid]:
            high = mid-1
        else:
            low = mid+1
    return False
```

## 7.

The image shows two windows side-by-side. On the left is the Python 3.7.5 Shell window, showing a command-line session. On the right is a code editor window titled 'ASD-4.7.py'.

**Python 3.7.5 Shell:**

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>>> RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.7.py ===
>>> kumpulan = [2, 4, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
>>> (binSeMass(kumpulan, 6))
[3, 4, 5]
>>> |
```

**ASD-4.7.py:**

```
File Edit Format Run Options Window Help
class MhsTif(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku
c0 = MhsTif("Jainal", 10, "Sukoharjo", 240000)
c1 = MhsTif("Fandit", 51, "Sraged", 230000)
c2 = MhsTif("Diko", 2, "Surakarta", 250000)
c3 = MhsTif("Ijui", 18, "Surakarta", 235000)
c4 = MhsTif("Ghani", 4, "Boyalali", 240000)
c5 = MhsTif("Rizki", 31, "Salatiga", 250000)
c6 = MhsTif("Bagus", 13, "Klaten", 245000)
c7 = MhsTif("Iqbal", 5, "Wonogiri", 245000)
c8 = MhsTif("Khalid", 23, "Klaten", 245000)
c9 = MhsTif("Azka", 64, "Karanganyar", 270000)
c10 = MhsTif("Bima", 29, "Purwodadi", 265000)
Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]
#NOMER 7
def binSeMass(kumpulan, target):
    temp = []
    low = 0
    high = len(kumpulan)-1
    while low <= high :
        mid = (high+low)//2
        if kumpulan[mid] == target:
            midKiri = mid-1
            while kumpulan[midKiri] == target:
                temp.append(midKiri)
                midKiri = midKiri-1
            temp.append(mid)
            midKanan = mid+1
            while kumpulan[midKanan] == target:
                temp.append(midKanan)
                midKanan = midKanan+1
        return temp
    elif target < kumpulan[mid]:
        high = mid-1
    else:
        low = mid+1
```

8.

```
Python 3.7.5 Shell
File Edit Shell Debug Options Window Help
[AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/ASD-4.8.py ====
Nomer 8
Ada dua pola
Pertama menggunakan konsep Big-O. Dimana yang dipakai
adalah rumus  $O(\log n)$  dengan rincian  $1 = 1$ ,  $2 = 2$ ,  $4 = 3$ ,  $10 = 4$ ,  $100 = 7$ ,  $1000 = 10$ .
Di mana log berasal dari pangkat log berbasis 2. Dengan begitu dapat mengetahui
jumlah
maksimal tebakan.
Untuk pola sendiri:
    apabila ingin menebak angka 70

    a = nilai tebakan pertama // 2
    tebakan selanjutnya = nilai tebakan "lebih dari" + a
    *jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
    tetap nilai lebih dari sebelumnya*
    a = a // 2
Simulasi
    tebakan ke 1: 50 (mengambil nilai tengah) jawaban= "lebih dari itu"
    tebakan ke 2: 75 (dari 50 + 25) jawaban = "kurang dari itu"
    tebakan ke 3: 62 (dari 50 + 12) jawaban = "lebih dari itu"
    tebakan ke 4: 68 (dari 62 + 6) jawaban = "lebih dari itu"
    tebakan ke 5: 71 (dari 68 + 3) jawaban = "kurang dari itu"
    tebakan ke 6: 69 (dari 68 + 1) jawaban = "lebih dari itu"
    tebakan ke 7: antara 71 dan 69 hanya ada 1 angka = 70!!!

Kedua menggunakan barisan geometri  $S_n = 2^n$ 
    barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
    Misal angka yang akan diebak adalah 68
    Tebakan ke-1 : 64 dijawab lebih dari itu
    Tebakan ke-2 : 96(dari 64 + 32) dijawab "Kurang dari itu"
    Tebakan ke-3 : 80(dari 64 + 16) dijawab "Kurang dari itu"
    Tebakan ke-4 : 72(dari 64 + 8) dijawab "Kurang dari itu"
    Tebakan ke-5 : 68(dari 64 + 4) dijawab "Lebih dari itu"
    Tebakan ke-6 : 70(dari 68 + 2) dijawab "TEPAT"

>>> | Ln: 38 Col: 4
```

Nama : Guntur Jatmiko

NIM : L200180039

Kelas : B

MODUL 5

The image shows two windows side-by-side. The left window is a code editor titled "modul5.py - C:\Users\Mr.G\Documents\AlgoStrukDat\REKAP\MODUL - 5\modul5.py (3.7.5)". It contains Python code for a class MhsTIF and various variable assignments. The right window is a "Python 3.7.5 Shell" titled "Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32". It shows the execution of the code, including imports, variable definitions, and a loop that prints student names, IDs, and ages.

```

#NO 1
class MhsTIF(object):
    def __init__(self,nama,nim,tinggal,us):
        self.nama = nama
        self.nim = nim
        self.tinggal = tinggal
        self.us = us

c0 = MhsTIF('Guntur', "L200180039", 'Sragen', 150000)
c1 = MhsTIF('Kelvin', "L200180102", 'Boylali', 146000)
c2 = MhsTIF('Anton', "L200180103", 'Makasar', 12500)
c3 = MhsTIF('Lovi', "L200180110", 'Solo', 340000)
c4 = MhsTIF('Dava', "L200180056", 'Kudus', 450000)
c5 = MhsTIF('Bol', "L200180222", 'Karanganyar', 460000)
c6 = MhsTIF('Aan', "L200180059", 'Solo', 770000)
c7 = MhsTIF('Isyana', "L200180239", 'kudus', 530000)
c8 = MhsTIF('Dodi', "L200180213", 'Banten', 235000)
c9 = MhsTIF('Amal', "L200180040", 'Sragen', 450000)

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

def swap(a,b,c):
    tmp=a[b]
    a[b]=a[c]
    a[c]=tmp

def ceknim(Daftar):
    for i in Daftar:
        print(i.nama,i.nim,i.tinggal)

def urutnim(a):
    n = len(a)
    for x in range(n-1):
        for y in range(n-x-1):
            if a[y].nim > a[y+1].nim:
                swap(a,y,y+1)

#NO 2
a = [1, 8, 22, 54, 89, 111, 102, 60]
b = [10, 2, 6, 12, 9, 23, 14, 33]

def urut(a,b):
    c = a + b
    for i in range(1,len(c)):
        nilai = c[i]

```

```

A[p] = A[q]
A[q] = tmp

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

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

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

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

aw = detak();bubbleSort(u_bub);ak=detak();print("Bubble      : %g detik"%(ak-aw));
aw = detak();selectionSort(u_sel);ak=detak();print("Selection   : %g detik"%(ak-aw))
aw = detak();insertionSort(u_ins);ak=detak();print("Insertion   : %g detik"%(ak-aw))

```

Ln: 1 Col: 0

```

#NO 2
a = [1, 8, 22, 54, 89, 111, 102, 60]
b = [10, 2, 6, 12, 9, 23, 14, 33]

def urut(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
    print(c)

#NO 3
def swap(A,p,q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp

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

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):

```

Ln: 1 Col: 0

Nama : Guntur Jatmiko

NIM : L200180039

Kelas : B

MODUL 6

File Edit Format Run Options Window Help

```
from mahasiswa import *
c0 = mahasiswa('Guntur', 50, 'Sragen', 300000)
c1 = mahasiswa('Kelvin', 23, 'Boylali', 200000)
c2 = mahasiswa('Anton', 13, 'Makasar', 700000)
c3 = mahasiswa('Lovi', 79, 'Patti', 600000)
c4 = mahasiswa('Dava', 48, 'Solo', 500000)
c5 = mahasiswa('Boi', 27, 'Demak', 400000)
c6 = mahasiswa('Isyana', 88, 'Jepara', 100000)
c7 = mahasiswa('Aan', 72, 'Soolo', 450000)
c8 = mahasiswa('Dodi', 92, 'Jember', 800000)
c9 = mahasiswa('Amal', 69, 'Grogol', 850000)

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

def cek(Daftar):
    for i in Daftar:
        print(i.nama,i.nim,i.tinggal)
#nomer 1
#mergesort
def mergesort(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].nim < separuhkanan[j].nim :
                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
    #quicksort
#def quicksort(A) :
```

---

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

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

    def appendSorted(self, data):
        if self.head == None:
            self.head = Node(data)
        else:
            current = self.head
            while current.next != None and current.next.data < data:
                current = current.next
            new_node = Node(data)
            new_node.next = current.next
            current.next = new_node

    def mergeSorted(self, list1, list2):
        if list1 is None:
            return list2
        if list2 is None:
            return list1

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

        return temp

    def printList(self):
        current = self.head
        while current != None:
            print(current.data)
            current = current.next

list1 = Linked()
list1.appendSorted(12)
list1.appendSorted(34)
list1.appendSorted(22)
list1.appendSorted(25)
list1.appendSorted(14)

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

list2 = Linked()
list2.appendSorted(14)
list2.appendSorted(21)
list2.appendSorted(11)

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

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

print("Mergesort Linked list :"),
list3.printList()
```

```
        print(cur.data)
        cur = cur.next
    def appendList(self, data):
        node = Node(data)
        if self.head == None:
            self.head = node
        else:
            curr = self.head
            while curr.next != None:
                curr = curr.next
            curr.next = node

    def appendSorted(self, data):
        node = Node(data)
        curr = self.head
        prev = None

        while curr is not None and curr.data < data:
            prev = curr
            curr = curr.next

        if prev == None:
            self.head = node
        else:
            prev.next = node

        node.next = curr

    def printList(self):
        curr = self.head
        while curr != None:
            print ("%d"%curr.data),
            curr = curr.next

    def mergeSorted(self, list1, list2):
        if list1 is None:
            return list2
        if list2 is None:
            return list1

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

```

a = L[low]
b = L[mid]
c = L[high - 1]
if a <= b <= c:
    return b, mid
if c <= b <= a:
    return b, mid
if a <= c <= b:
    return c, high - 1
if b <= c <= a:
    return c, high - 1
return a, low

daftar = [23,1,3,56,44,33,75,86,34,21,34,11,24,35]
from time import time as detak
from random import shuffle as kocok
import time

k = [[i] for i in range(1, 6001)]
kocok(k)
u_mer = k[:]
u_mer5 = k[:]
u_qui = k[:]
u_qui6 = k[:]

aw=detak();mergesort(u_mer);ak=detak();print("mergesort           : %g detik" %(ak-aw));
aw=detak();mergesort_5(u_mer5);ak=detak();print("mergesort terbaru : %g detik" %(ak-aw));
aw=detak();quicksort(u_qui);ak=detak();print("quicksort          : %g detik" %(ak-aw));
aw=detak();quicksort_6(u_qui6);ak=detak();print("quicksort terbaru : %g detik" %(ak-aw));

#8
class Node():
    def __init__(self,data,next= None,prev = None):
        self.data = data
        self.next = next
        self.prev = prev

class Linked():
    def __init__(self,head = None):
        self.head = head

    def cetak(self):
        cur = self.head
        while cur != None:
            print(cur.data)
            cur = cur.next
        else:
            print("Data tidak ditemukan")

```

```

        l += 1
        a += 1
#proses penggabungan
    if f <= mid:
        tmp[a:] = A[f:mid+1]
    if l <= akhir:
        tmp[a:] = A[l:akhir+1]
#memindah isi tmp ke A
a = 0
while awal <= akhir:
    A[awal] = tmp[a]
    awal += 1
    a += 1

def mergesort_5(A):
    mergesort2_5(A, 0, len(A)-1)

#quick sort terbaru
def quicksort_6(L, ascending = True):
    quicksorthelp(L, 0, len(L), ascending)

def quicksorthelp(L, low, high, ascending = True):
    result = 0
    if low < high:
        pivot_location, result = Partition(L, low, high, ascending)
        result += quicksorthelp(L, low, pivot_location, ascending)
        result += quicksorthelp(L, pivot_location + 1, high, ascending)
    return result

def Partition(L, low, high, ascending = True):
    result = 0
    pivot, pidx = median_of_three(L, low, high)
    L[low], L[pidx] = L[pidx], L[low]
    i = low + 1
    for j in range(low + 1, high, 1):
        result += 1
        if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot):
            L[i], L[j] = L[j], L[i]
            i += 1
    L[low], L[i - 1] = L[i - 1], L[low]
    return i - 1, result

def median_of_three(L, low, high):
    mid = (low + high - 1) // 2
    a = L[low]
    b = L[mid]
    c = L[high - 1]

```

```

penandakanan = akhir
selesai = False

while not selesai:
    while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
        penandakiri +=1
    while A[penandakanan] >= nilaipivot and penandakanan >= penandakiri :
        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)

#merge sort terbaru
def mergesort2_5(A, awal, akhir):
    mid = (awal+akhir)//2
    if awal < akhir:
        mergesort2_5(A, awal, mid)
        mergesort2_5(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

```

```

    if a <= b <= c:
        return b, mid
    if c <= b <= a:
        return b, mid
    if a <= c <= b:
        return c, high - 1
    if b <= c <= a:
        return c, high - 1
    return a, low

print("sebelum","\n",daftar)
quickSort(daftar)
print("sesudah","\n",daftar)

#nomor 7
def mergesort(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+=1
            else:
                A[k] = separuhkanan[j]
                j+=1
            k+=1
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i+=1
            k+=1
        while j< len(separuhkanan):
            A[k] = separuhkanan[j]
            j+=1
            k+=1
    alist = [23,1,3,56,44,33,75,86,34,21,34,11,24,35]

def partisi(A,awal,akhir):
    nilaipivot = A[awal]
    penandakiri = awal + 1
    penandakanan = akhir
    selesai = False

```

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

print("sebelum","\n",daftar)
mergeSort(daftar)
print("sesudah","\n",daftar)

#nomor 6
daftar = [54,26,93,17,77,31,44,55,20]
def quickSort(L, ascending = True):
    quicksorthelp(L, 0, len(L), ascending)

def quicksorthelp(L, low, high, ascending = True):
    result = 0
    if low < high:
        pivot_location, result = Partition(L, low, high, ascending)
        result += quicksorthelp(L, low, pivot_location, ascending)
        result += quicksorthelp(L, pivot_location + 1, high, ascending)
    return result

def Partition(L, low, high, ascending = True):
    result = 0
    pivot, pidx = median_of_three(L, low, high)
    L[low], L[pidx] = L[pidx], L[low]
    i = low + 1
    for j in range(low + 1, high, 1):
        result += 1
        if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot):
            L[i], L[j] = L[j], L[i]
            i += 1
    L[low], L[i - 1] = L[i - 1], L[low]
    return i - 1, result

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

---

```

u_bub = k[:]
u_sel = k[:]
u_ins = k[:]
u_mrg = k[:]
u_qck = k[:]

aw=detak();bubbleSort(u_bub);ak=detak();print("bubble: %g detik" %(ak-aw));
aw=detak();selectionSort(u_sel);ak=detak();print("selection: %g detik" %(ak-aw));
aw=detak();insertionSort(u_ins);ak=detak();print("insertion: %g detik" %(ak-aw));
aw=detak();mergeSort(u_mrg);ak=detak();print("merge: %g detik" %(ak-aw));
aw=detak();quickSort(u_qck);ak=detak();print("quick: %g detik" %(ak-aw));

#nomor 5
daftar = [23,44,12,45,78,45,34,97,56,43,34,22,67,88,77]
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
    #proses penggabungan
    if f <= mid:
        tmp[a:] = A[f:mid+1]
    if l <= akhir:
        tmp[a:] = A[l:akhir+1]
#memindah isi tmp ke A
    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

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

print("sebelum","\n",daftar)
mergeSort(daftar)
print("sesudah","\n",daftar)

```

```
penandakanan = akhir

selesai = False
while not selesai:

    while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
        penandakiri = penandakiri + 1

    while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
        penandakanan = penandakanan - 1

    if penandakanan < penandakiri:
        selesai = True
    else:
        temp = A[penandakiri]
        A[penandakiri] = A[penandakanan]
        A[penandakanan] = temp

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

return penandakanan

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

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

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

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

k = [[i] for i in range(1, 6001)]
kocok(k)
u_bub = k[:]
u_sel = k[:]
..
```

```

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

        mergeSort(separuhkiri)
        mergeSort(separuhkanan)

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

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

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

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

    penandakiri = awal + 1
    penandakanan = akhir

    selesai = False
    while not selesai:

        while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
            penandakiri = penandakiri + 1

        while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
            penandakanan = penandakanan - 1

        if penandakanan < penandakiri:
            selesai = True

```

```
#nomer 3
from time import time as detak
from random import shuffle as kocok
import time

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

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

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

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

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

def mergeSort(A):
    #print("Membelah      ",A)
    if len(A) < 1:
```

```

        k = k+1
    #quicksort
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].nim
    penandakiri = awal + 1
    penandakanan = akhir
    selesai = False

    while not selesai:
        while penandakiri <= penandakanan and A[penandakiri].nim <= nilaipivot:
            penandakiri +=1
        while A[penandakanan].nim >= nilaipivot and penandakanan >= penandakiri :
            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

cek(Daftar)
print("====")
print("mergesortnya")
print("====")
mergesort(Daftar)
cek(Daftar)
print("====")
print("quicksortnya")
quicksort(Daftar)
print("====")
cek(Daftar)

#nomer 3
from time import time

```

Nama : Guntur Jatmiko

NIM : L200180039

Kelas : B

## LAPRAK ASD MODUL 7

### NO 1.

The screenshot shows two side-by-side Python 3.7.5 shells. The left shell contains a script named 'asd71.py' that reads the file 'indonesia.txt' and prints its contents. The right shell shows the output of running this script, displaying the text from the file.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/asd71.py =====
['menjadi', 'menjalin', 'membawa', 'memonopoli', 'menyatakan', 'mendapatkan', 'Merakau', 'Melanesia', 'mendiami', 'mencapai', 'membentuk', 'memiliki', 'memiliki', 'mendukung', 'merupakan', 'menjadi']
>>>
```

```
asd71.py - C:\Users\Mr.G\AppData\Local\Programs\Python\Python37\asd71.py (3.7.5)
File Edit Shell Debug Options Window Help
import re
#No 1
f = open('indonesia.txt', 'r', encoding='latin1')
x = []
teks = f.read()
f.close()
p = r'\s([Mm]e\w+)'
cocok = re.findall(p, teks)
x.append(cocok)
print(x)
```

### NO 2.

The screenshot shows two side-by-side Python 3.7.5 shells. The left shell contains a script named 'asd72.py' that reads 'indonesia.txt' and prints specific words. The right shell shows the output of running this script, displaying the words 'dilebut', 'dilintasi', 'dipakai', 'dipiilih', 'dipengaruhi', and 'diikuti'.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/asd72.py =====
['dilebut', 'dilintasi', 'dipakai', 'dipiilih', 'dipengaruhi', 'diikuti', 'dinya takan']
>>>
```

```
asd72.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/asd72.py (3.7.5)
File Edit Format Run Options Window Help
import re
#No 2
f = open('indonesia.txt', 'r', encoding='latin1')
z = []
teks = f.read()
f.close()
p = r'\s(di\w+)' 
cocok = re.findall(p, teks)
z.append(cocok)
print(z)
```

### NO 3.

The screenshot shows two side-by-side Python 3.7.5 shells. The left shell contains a script named 'asd73.py' that reads 'indonesia.txt' and prints specific words. The right shell shows the output of running this script, displaying the words 'di Asia', 'di antara', 'di dunia', 'di Pulau', 'di Pulau', 'di India', 'di Palembang', 'di bawah', 'di akhir', 'di ujung', 'di tanah', 'di mana', and 'di dunia'.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/asd73.py =====
['di Asia', 'di antara', 'di dunia', 'di dunia', 'di Pulau', 'di Pulau', 'di India', 'di Palembang', 'di bawah', 'di akhir', 'di ujung', 'di tanah', 'di mana', 'di dunia']]
>>>
```

```
asd73.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/asd73.py (3.7.5)
File Edit Format Run Options Window Help
import re
#No 3
f = open('indonesia.txt', 'r', encoding='latin1')
y = []
teks = f.read()
f.close()
p = r'\s(di\w+)' 
cocok = re.findall(p, teks)
y.append(cocok)
print(y)
```

### NO 4.

The screenshot shows two side-by-side Python 3.7.5 shells. The left shell contains a script named 'asd74.py' that reads 'K6I.html' and calculates averages for various countries. The right shell shows the output of running this script, displaying the average values for each country.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/asd74.py =====
>>> print(cocok)
[('Denmark', 9.57), ('Sweden', 9.79), ('Finland', 9.66), ('Netherlands', 9.48), ('Norway', 9.06), ('Canada', 9.43), ('Switzerland', 9.89), ('Kingdom', 9.18), ('States', 9.45), ('Australia', 8.72), ('Ireland', 9.04), ('Austria', 8.9), ('Iceland', 7.98), ('Germany', 9.0), ('Zealand', 8.65), ('Belgium', 8.96), ('Taiwan', 9.24), ('Luxembourg', 8.91), ('Japan', 9.15), ('France', 8.61), ('Estonia', 7.49), ('Slovenia', 8.31), ('Spain', 8.14), ('Singapore', 9.56), ('Israel', 9.34), ('China', 8.64), ('Italy', 8.04), ('Hungary', 8.14), ('Republic', 7.6), ('Lithuania', 6.59), ('Korea', 8.47), ('Latvia', 6.4), ('Cyprus', 7.65), ('Portugal', 7.43), ('Greece', 7.63), ('Poland', 6.92), ('Slovakia', 6.86), ('Barbados', 7.51), ('Croatia', 7.54), ('Chile', 6.81), ('Bulgaria', 6.43), ('Emirates', 6.74), ('Romania', 5.66), ('Uruguay', 5.26), ('Qatar', 5.77), ('Dominica', 3.76), ('Rica', 6.24), ('Malaysia', 6.83), ('Federation', 6.89), ('Bahrain', 4.21), ('Kuwait', 5.05), ('Ukraine', 5.77), ('Argentina', 6.85), ('Tobago', 6.02), ('Brazil', 6.07), ('Turkey', 5.67), ('Africa', 6.92), ('Jordan', 5.66), ('Armenia', 6.17), ('Mexico', 5.82), ('Thailand', 5.98), ('Oman', 4.95), ('Macedonia', 4.76), ('Mauritius', 3.7), ('Arabia', 4.04), ('Jamaica', 5.36), ('Moldova', 4.39), ('Kazakhstan', 3.77), ('Belarus', 5.54), ('Lebanon', 4.69), ('Tunisia', 4.58), ('Panama', 5.45), ('Georgia', 5.39), ('Peru', 3.88), ('Mongolia', 2.06), ('Colombia', 4.26), ('China', 5.12), ('Guyana', 4.47), ('Philippines', 3.63), ('Venezuela', 5.73), ('Namibia', 3.3), ('Lanka', 4.44), ('Albania', 3.1), ('Egypt', 4.55), ('Botswana', 4.34), ('Republic', 2.91), ('Salvador', 3.19), ('Azerbaijan', 3.05), ('Kyrgyzstan', 2.7), ('Paraguay', 3.47), ('Ecuador', 3.55), ('Morocco', 3.67), ('Bolivia', 3.05), ('Iran', 3.02), ('Uzbekistan', 3.51), ('Algeria', 3.49), ('Verde', 2.25), ('Indonesia', 3.32), ('Honduras', 3.3), ('India', 3.97), ('Guatemala', 2.47), ('Vietnam', 2.83), ('Swaziland', 4.55), ('Republic', 3.44), ('Nicaragua', 1.99), ('Kenya', 3.87), ('Tajikistan', 2.33), ('Senegal', 2.77), ('Zimbabwe', 4.09), ('Ghana', 2.08), ('Uganda', 2.72), ('Madagascar', 2.54), ('Mauritania', 1.75), ('Tanzania', 2.39), ('Pakistan', 2.75), ('Lesotho', 2.7), ('Benin', 2.33), ('Nigeria', 2.72), ('Yemen', 1.68), ('Malta', 1.69), ('Mozambique', 1.86), ('Angola', 2.44), ('Cameroon', 2.49), ('Faso', 2.15), ('Nepal', 2.04), ('Malawi', 2.11), ('Laos', 1.43), ('Bangladesh', 1.71), ('Myanmar', 1.17), ('Rwanda', 1.47), ('Ethiopia', 1.57), ('Djibouti', 1.29), ('Eritrea', 1.56), ('Leone', 1.7), ('Herzegovina', 3.29), ('Montenegro', 4.85), ('Haiti', 1.15), ('Ivoire', 2.52), ('Sudan', 1.97), ('Zambia', 2.37)]
```

```
asd74.py - C:/Users/Mr.G/AppData/Local/Programs/Python/Python37/asd74.py (3.7.5)
File Edit Format Run Options Window Help
import re
#No 4
f = open('K6I.html', 'r', encoding='latin1')
teks = f.read()
f.close()
i = r'\s<td>[\d.\w/]</td>' 
p = r'(\w+)</td></td>' + i + i + i + r'\s<td>([\d.\w/]*)</td>' 
cocok = re.findall(p, teks)
cocok1 = [(i[0], float(i[1])) for i in cocok]
```

Nama : Guntur Jatmiko  
NIM : L200180039  
Kelas : B

## MODUL 8

### 1.

The screenshot shows a Python development environment with two windows. On the left is a code editor window titled '1.py - C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\1.py (3.7.5)'. It contains Python code for a stack implementation and a hex conversion function. On the right is a Python Shell window titled 'Python 3.7.5 Shell' showing the execution of the code. The shell output shows the stack operations and the resulting hex conversion for various numbers.

```
class stack():
    def __init__(self):
        self.list = []
    def empty(self):
        return len(self.list) == 0
    def __len__(self):
        return len(self.list)
    def push(self, data):
        self.list.append(data)
    def pop(self):
        assert not self.empty(), 'No file'
        return self.list.pop()

def cetak_hexa(angka):
    x = stack()

    if angka == 0:
        x.push(0)
    while angka != 0:
        sisa = angka % 16
        angka = angka // 16
        x.push(sisa)
    hexa = [0,1,2,3,4,5,6,7,8,9,'A','B','C','D','E','F']
    hasil = ''
    for i in range(len(x)):
        hasil += str(hexa[x.pop()])
    return hasil

print(cetak_hexa(12))
print(cetak_hexa(31))
print(cetak_hexa(229))
print(cetak_hexa(225))
print(cetak_hexa(31519))
```

```
>>> Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\1.py =====
C
1F
E5
E1
7B1F
>>> |
```

### 2.

The screenshot shows a Python development environment with two windows. On the left is a code editor window titled '2.py - C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\2.py (3.7.5)'. It contains Python code for a Stack class and a script that uses it to calculate remainders of division by 3 for numbers from 0 to 15. On the right is a Python Shell window titled 'Python 3.7.5 Shell' showing the execution of the code. The shell output shows the results of the calculations.

```
class Stack(object):
    """docstring for Stack"""
    def __init__(self):
        #membuat stack kosong
        self.item=[]
    def isEmpty(self):
        #mengembalikan nilai boolean yg menunjukkan apakah stack ini kosong atau tidak
        return len(self)==0
    def __len__(self):
        #mengembalikan banyaknya item di stack
        return len(self.item)
    def peek(self):
        #mengembalikan nilai posisi atas tanpa menghapus dan mengembalikannya
        assert not self.isEmpty(), "stack is Empty"
        return self.item[-1]
    def pop(self):
        #mengembalikan nilai posisi atas lalu menghapus, stack kosong
        assert not self.isEmpty(), "Stack is Empty"
        return self.item.pop()
    def push(self, data):
        #memindahkan item ke stack, menambahkan item ke puncak stack
        self.item.append(data)

nilai=Stack() #menyimpan nilai stack di variable nilai
for i in range(16): #perulangan dalam range 0-15
    if i % 3 == 0: #jika nilai dibagi 3 memiliki sisa hasil bagi 0
        nilai.push(i) #menambahkan nilai tersebut pada variable nilai

print(nilai.item) #menampilkan hasil akhir pada variable nilai

#nilai 0 : sisa hasil bagi sama dengan 0 (memenuhi)
#menambahkan nilai 0 pada variable nilai
#    >>>[0]
#nilai 1 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
#nilai 2 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
#nilai 3 : sisa hasil bagi sama dengan 0 (memenuhi)
#    menambahkan nilai 3 pada variable nilai
#    >>>[0,3]
#nilai 4 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
#nilai 5 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
#nilai 6 : sisa hasil bagi sama dengan 0 (memenuhi)
#    menambahkan nilai 6 pada variable nilai
#    >>>[0,3,6]
#nilai 7 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
#nilai 8 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
#nilai 9 : sisa hasil bagi sama dengan 0 (memenuhi)
#    menambahkan nilai 9 pada variable nilai
#    >>>[0,3,6,9]
#nilai 10 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
#nilai 11 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
#nilai 12 : sisa hasil bagi sama dengan 0 (memenuhi)
#    menambahkan nilai 12 pada variable nilai
#    >>>[0,3,6,9,12]
#nilai 13 : sisa hasil bagi tidak sama dengan 0 (tidak memenuhi)
```

```
>>> Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\2.py =====
[0, 3, 6, 9, 12]
>>> |
```

### 3.

```
3.py - C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\3.py (3.7.5)
File Edit Format Run Options Window Help
class Stack(object):
    """docstring for Stack"""
    def __init__(self): #membuat stack kosong
        self.item=[]

    def isEmpty(self): #mengembalikan nilai boolean yg menunjukkan apakah stack kosong
        return len(self)==0

    def __len__(self): #mengembalikan banyaknya item di stack
        return len(self.item)

    def peek(self): #mengembalikan nilai posisi atas tanpa menghapus
        assert not self.isEmpty(), "Stack is Empty"
        return self.item[-1]

    def pop(self): #mengembalikan nilai posisi atas lalu menghapus
        assert not self.isEmpty(), "Stack is Empty"
        return self.item.pop()

    def push(self, data): #memindahkan item ke stack, menambah item ke puncak
        self.item.append(data)

nilai=Stack() #menyimpan nilai stack di variable nilai
for i in range(16): #perulangan dalam range 0-15
    if i % 3 == 0: #jika nilai dibagi 3 memiliki sisa hasil bagi 0
        nilai.push(i) #menambahkan nilai tersebut pada variable nilai
    elif i % 4 == 0: #jika nilai dibagi 4 memiliki sisa hasil bagi 0
        nilai.pop() #menghapus nilai tersebut pada variable nilai

print(nilai.item) #menampilkan hasil akhir pada variable nilai
```

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> ===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\3.py ======  
[0, 9, 12, 15]  
>>>

### 4.

```
4.py - C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\4.py (3.7.5)
File Edit Format Run Options Window Help
class Queue():
    def __init__(self):
        self.qlist = []
    def is_empty(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.is_empty(), 'Antrian sedang kosong'
        return self.qlist.pop(0)
    def get_front_most(self):
        assert not self.is_empty(), 'Antrian sedang kosong'
        return self.qlist[0]
    def get_rear_most(self):
        assert not self.is_empty(), 'Antrian sedang kosong'
        return self.qlist[-1]

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

print(bb.get_front_most())
print(bb.get_rear_most())
print(bb.qlist)
```

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> ===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\4.py ======  
28  
7  
[28, 19, 45, 13, 7]  
>>>

### 5.

```
5.py - C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\5.py (3.7.5)
File Edit Format Run Options Window Help
class PriorityQueue():
    def __init__(self):
        self.qlist = []
    def __len__(self):
        return len(self.qlist)
    def is_empty(self):
        return len(self) == 0
    def enqueue(self, data, priority):
        entry = PriorityQEntry(data, priority)
        self.qlist.append(entry)
    def dequeue(self):
        A = []
        for i in self.qlist:
            A.append(i)
        s = 0
        for i in range(1, len(self.qlist)):
            if A[i].priority < A[s].priority:
                s = i
        hasil = self.qlist.pop(s)
        return hasil.item

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

bc = PriorityQueue()
bc.enqueue("Rambutan",4)
bc.enqueue("Apel",2)
bc.enqueue("Pepaya",0)
bc.enqueue("Pisang",5)
bc.enqueue("Manggis",2)

print(bc.dequeue())
print(bc.dequeue())
print(bc.dequeue())
print(bc.dequeue())
print(bc.dequeue())
```

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> ===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\MODDUOL8\5.py ======  
Pepaya  
Apel  
Manggis  
Rambutan  
Pisang  
>>> |

Nama : Guntur Jatmiko  
NIM : L200180039  
Kelas : B

## LAPRAK MODUL 9

### Soal – Soal Untuk Mahasiswa

1. Diberikan pohon biner dengan ukuran n, berapakah jumlah level minimum yang bisa dimuatnya? Berapakah jumlah level maksimumnya? Tentukan untuk nilai n berikut.

a. N = 10

- jumlah level minimumnya adalah 3
- Jumlah level maksimumnya adalah 9

b. N = 35

- jumlah level minimumnya adalah 6
- Jumlah level maksimumnya adalah 34

c. N = 76

- jumlah level minimumnya adalah 7
- jumlah level maksimumnya adalah 75

d. N = 345

- jumlah level minimumnya adalah 8
- jumlah level maksimumnya adalah 334

2. Gambarlah semua bentuk pohon biner berukuran 5 yang mungkin. Ada beberapa kemungkinan?

$$C_n = (2n)! / (n+1)! * n!$$

$$= (2*5)! / (5+1)! + 5!$$

$$= 10! / 6! * 5!$$

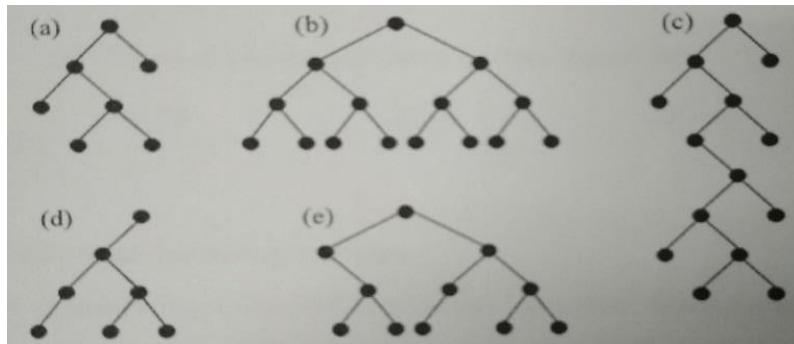
$$= 3628800 / 86400$$

$$= 42 \text{ kemungkinan}$$

3. Berapakah jumlah simpul maksimum suatu pohon biner dengan jumlah level h? tentukan untuk nilai h berikut.

- a.  $H = 3$  jumlah simpul maksimum adalah 7
- b.  $H = 4$  jumlah simpul maksimum adalah 15
- c.  $H = 5$  jumlah simpul maksimum adalah 31
- d.  $H = 6$  jumlah simpul maksimum adalah 63

4. Diberikan pohon-pohon biner seperti di bawah



- a. Tunjukkan semua properti struktural yang berlaku pada tiap-tiap pohon di atas: penuh, sempurna, komplit. 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

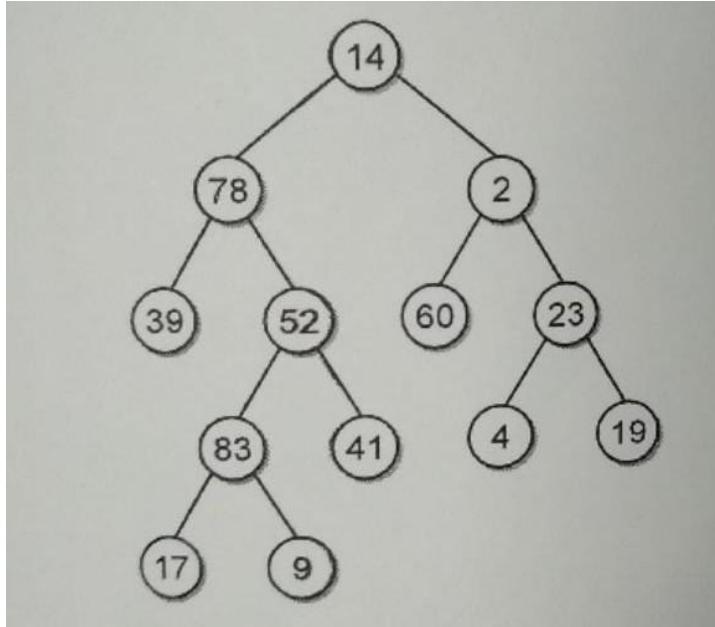
- a = 7
- b = 15
- c = 14
- d = 7
- e = 11

- c. Tentukan lebar tiap pohon

- a = 4
- b = 4
- c = 8

- $d = 4$
- $e = 4$

5. Perhatikan pohon biner berikut



a. Tunjukkan urutan pengunjungan simpul untuk

i. Preorder traversal

14-78-39-52-83-17-9-41-2-60-23-4-19 ii.

Inorder traversal

39-78-17-83-9-52-41-14-60-2-4-23-19

iii. 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  
i. 83

15 – 78 – 52 – 83

- ii. 39

14 - 78 - 39

- iii. 4

14 – 2 – 23 – 4

- iv. 9

14 - 78 - 52 - 83 - 9

- f. Perhatikan simpul 52. tentukan

- i. Keturunannya

83,41,17,9 ii.

Leluhurnya

78,14 iii.

Saudaranya

39,60,23

- g. Tentukan kedalaman dari tiap-tiap simpul ini:

1) 78 = level 1

2) 41 = level 2

3) 60 = level 2

4) 19 = level 3

## Soal-soal pemrograman

6.



```
#nomer 6
class SimpulPohonBiner(object):
    def __init__(self, data):
        self.data = data
        self.kiri = None
        self.kanan = None

A = SimpulPohonBiner("Ambarawa")
B = SimpulPohonBiner("Bantul")
C = SimpulPohonBiner("Cimahi")
D = SimpulPohonBiner("Denpasar")
E = SimpulPohonBiner("Surakarta")
F = SimpulPohonBiner("Klaten")
G = SimpulPohonBiner("Bogor")
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

def ukuranPohon(akar):
    ukuran = 0
    if akar is not None:
        if akar.kiri is None and akar.kanan is None:
            ukuran +=1
        else:
            hasil = ukuranPohon(akar.kiri)
            ukuran += hasil
            hasil = ukuranPohon(akar.kanan)
            ukuran += hasil
    return ukuran
```

7.

```
#nomer 7
def tinggiPohon(akar):
    if akar is None:
        return 0
    else:
        kiri = tinggiPohon(akar.kiri)
        kanan = tinggiPohon(akar.kanan)
        if kiri > kanan :
            return kiri +1
        else:
            return kanan +1
```

8.

```
#nomer 8
def cetakDataDanLevel(akar, level =-1):
    level += 1
    if akar is not None:
        print(akar.data, "Level", level)
        cetakDataDanLevel(akar.kiri,level)
        cetakDataDanLevel(akar.kanan,level)
```

Ln: 1 Col: 0

```
File Edit Shell Debug Options Window Help
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Mr.G\Documents\AlgoStrukDat\6.py =====
>>> cetakDataDanLevel(A)
Ambarawa Level 0
Bantul Level 1
Denpasar Level 2
Surakarta Level 2
Halmahera Timur Level 3
Cimahi Level 1
Klaten Level 2
Bogor Level 2
Indramayu Level 3
>>>
```

Nama : Guntur J  
NIM : L200180039  
Kelas : B

## LAPRAK ASD MODUL 10

The image shows four windows of a Python 3.7.5 IDE. The windows are arranged in a grid:

- Python 3.7.5 Shell (Top Left):** Shows the Python interpreter prompt (">>>> ) and the execution of a script named '1.py'.
- 1.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/1.py (Top Right):** Contains the code for a function 'jumlahkan\_cara\_1(n)' which calculates the sum of integers from 1 to n using a simple loop.
- 2.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/2.py (Middle Left):** Shows the Python interpreter prompt and the execution of a script named '2.py'.
- 2.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/2.py (Middle Right):** Contains the code for a function 'jumlahkan\_cara\_1(n)' and a timing test that measures the execution time for summing 10,000,000 integers.
- 3.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/3.py (Bottom Left):** Shows the Python interpreter prompt and the execution of a script named '3.py'.
- 3.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/3.py (Bottom Right):** Contains the code for an insertion sort algorithm and a timing test that measures the execution time for sorting 3,000 random integers.

**1.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/1.py (Top Right):**

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

**2.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/2.py (Middle Right):**

```
import time

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

for i in range(5):
    awal = time.time()
    h = jumlahkan_cara_1(1000000)
    akhir = time.time()
    print("jumlah adalah %d, memerlukan %.8f detik" % (h, akhir-awal))
```

**3.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/3.py (Bottom Right):**

```
import time
import random

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

for i in range(5):
    L = list(range(3000))
    random.shuffle(L)
    awal = time.time()
    U = insertionSort(L)
    akhir = time.time()
    print("mengurutkan %d bilangan, memerlukan %.7F detik" % (len(L), akhir-awal))
```

**3.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/3.py (Bottom Right):**

```
===== RESTART: C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/3.py ======
mengurutkan 3000 bilangan, memerlukan 0.4555678 detik
mengurutkan 3000 bilangan, memerlukan 0.4279096 detik
mengurutkan 3000 bilangan, memerlukan 0.4335997 detik
mengurutkan 3000 bilangan, memerlukan 0.4288647 detik
mengurutkan 3000 bilangan, memerlukan 0.4324627 detik
>>>
```

The image shows four windows side-by-side, each running Python 3.7.5. The windows are arranged in two pairs, with a vertical gap between them.

**Top Left Window:**

```

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/4.py =====
mengurutkan 3000 bilangan, memerlukan 0.8351205 detik
mengurutkan 3000 bilangan, memerlukan 0.8150674 detik
mengurutkan 3000 bilangan, memerlukan 0.8319249 detik
mengurutkan 3000 bilangan, memerlukan 0.8266730 detik
mengurutkan 3000 bilangan, memerlukan 0.8274131 detik
>>> |

```

**Top Right Window:**

```

4.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/4.py (3.7.5)
File Edit Format Run Options Window Help
import time
import random
def insertionSort(L):
    n = len(L)
    for i in range (1, n):
        nilai = L[i]
        pos = i
        while pos > 0 and nilai < L[pos -1]:
            L[pos] = L[pos - 1]
            pos = pos -1
        L[pos] = nilai

for i in range(5):
    L = list(range(3000))
    L = L[::-1]
    awal = time.time()
    U = insertionSort(L)
    akhir = time.time()
    print("mengurutkan %d bilangan, memerlukan %.7F detik" % (len(L),akhir-awal))

```

**Bottom Left Window:**

```

Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/5.py =====
mengurutkan 3000 bilangan, memerlukan 0.0019925 detik
mengurutkan 3000 bilangan, memerlukan 0.0020292 detik
mengurutkan 3000 bilangan, memerlukan 0.0019941 detik
mengurutkan 3000 bilangan, memerlukan 0.0020278 detik
mengurutkan 3000 bilangan, memerlukan 0.0010438 detik
>>> |

```

**Bottom Right Window:**

```

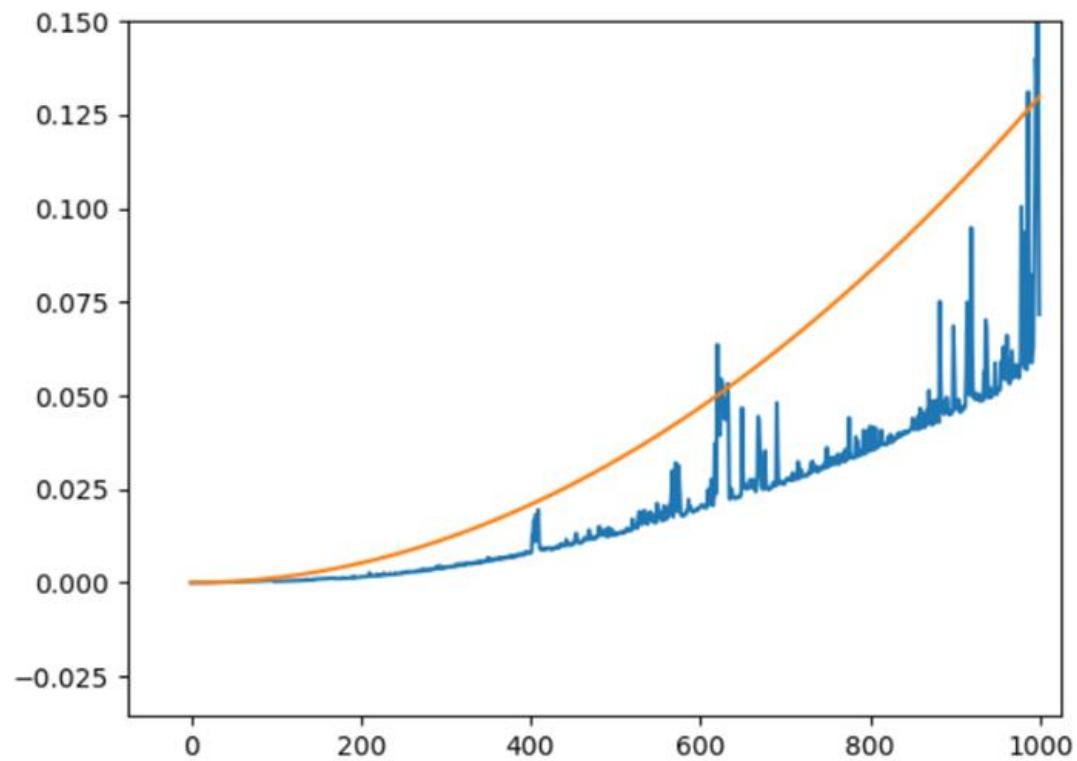
5.py - C:/Users/Mr.G/Documents/AlgoStrukDat/MODULL 10/5.py (3.7.5)
File Edit Format Run Options Window Help
import time
import random
def insertionSort(L):
    n = len(L)
    for i in range (1, n):
        nilai = L[i]
        pos = i
        while pos > 0 and nilai < L[pos -1]:
            L[pos] = L[pos - 1]
            pos = pos -1
        L[pos] = nilai

for i in range(5):
    L = list(range(3000))
    awal = time.time()
    U = insertionSort(L)
    akhir = time.time()
    print("mengurutkan %d bilangan, memerlukan %.7F detik" % (len(L),akhir-awal))

```

Figure 1

- □ ×



pan/zoom

```
import timeit
import matplotlib.pyplot as plt

##Ini fungsi nested loop yang akan diuji:
def kalangBersusuh(n):
    for i in range(n):
        for j in range(n):
            i+j

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

## Pemanggilan pengujian
n = 1000
LS = ujiKalangBersusuh(n) # dari 1 sampai 1000.
## LS adalah list hasil uji kecepatan, dari n sedikit ke banyak

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