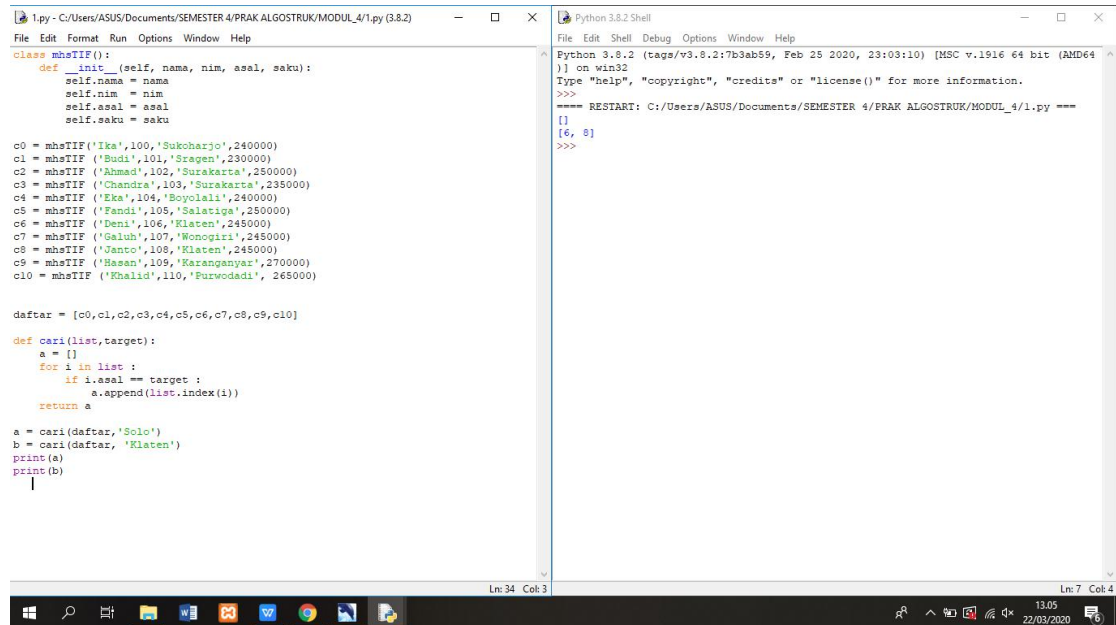


**NAMA : ELSA PUTRI ALIYYA**  
**NIM : L200180108**  
**KELAS : D**

## TUGAS MODUL 4

1.



```
1.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_4/1.py (3.8.2)
Python 3.8.2 Shell

class mhsTIF():
    def __init__(self, nama, nim, asal, saku):
        self.nama = nama
        self.nim = nim
        self.asal = asal
        self.saku = saku

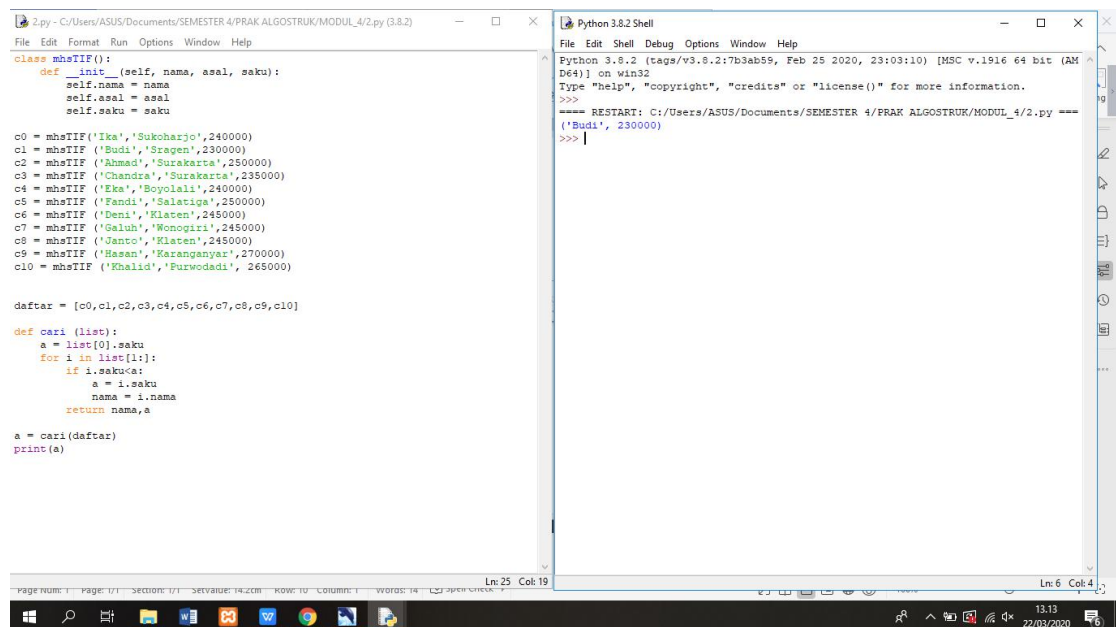
c0 = mhsTIF('Ika',100,'Sukoharjo',240000)
c1 = mhsTIF('Budi',101,'Sragen',230000)
c2 = mhsTIF('Ahmad',102,'Surakarta',250000)
c3 = mhsTIF('Chandra',103,'Surakarta',235000)
c4 = mhsTIF('Eka',104,'Boyolali',240000)
c5 = mhsTIF('Fandi',105,'Salatiga',250000)
c6 = mhsTIF('Deni',106,'Klaten',245000)
c7 = mhsTIF('Galuh',107,'Wonogiri',245000)
c8 = mhsTIF('Janto',108,'Klaten',245000)
c9 = mhsTIF('Hasan',109,'Karanganyar',270000)
c10 = mhsTIF('Khalid',110,'Purwodadi', 265000)

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

def cari(list,target):
    a = []
    for i in list :
        if i.asal == target :
            a.append(list.index(i))
    return a

a = cari(daftar,'Solo')
b = cari(daftar, 'Klaten')
print(a)
print(b)
```

2.



```
2.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_4/2.py (3.8.2)
Python 3.8.2 Shell

class mhsTIF():
    def __init__(self, nama, asal, saku):
        self.nama = nama
        self.asal = asal
        self.saku = saku

c0 = mhsTIF('Ika', 'Sukoharjo', 240000)
c1 = mhsTIF('Budi', 'Sragen', 230000)
c2 = mhsTIF('Ahmad', 'Surakarta', 250000)
c3 = mhsTIF('Chandra', 'Surakarta', 235000)
c4 = mhsTIF('Eka', 'Boyolali', 240000)
c5 = mhsTIF('Fandi', 'Salatiga', 250000)
c6 = mhsTIF('Deni', 'Klaten', 245000)
c7 = mhsTIF('Galuh', 'Wonogiri', 245000)
c8 = mhsTIF('Janto', 'Klaten', 245000)
c9 = mhsTIF('Hasan', 'Karanganyar', 270000)
c10 = mhsTIF('Khalid', 'Purwodadi', 265000)

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

def cari (list):
    a = list[0].saku
    for i in list[1:]:
        if i.saku==a:
            a = i.saku
            nama = i.nama
            return nama,a

a = cari(daftar)
print(a)
```

3.

```

class mhsTIF():
    def __init__(self, nama, asal, saku):
        self.nama = nama
        self.asal = asal
        self.saku = saku

c0 = mhsTIF('Eka', 'Sukoharjo', 240000)
c1 = mhsTIF('Budi', 'Sragen', 230000)
c2 = mhsTIF('Ahmad', 'Surakarta', 250000)
c3 = mhsTIF('Chandra', 'Surakarta', 235000)
c4 = mhsTIF('Eka', 'Boyolali', 240000)
c5 = mhsTIF('Fandi', 'Salatiga', 250000)
c6 = mhsTIF('Deni', 'Klaten', 245000)
c7 = mhsTIF('Galuh', 'Wonogiri', 245000)
c8 = mhsTIF('Janto', 'Klaten', 245000)
c9 = mhsTIF('Hasan', 'Karanganyar', 270000)
c10 = mhsTIF('Khalid', 'Purwodadi', 265000)

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

def cari (list):
    a = [list[0]]
    for i in list[1:]:
        if i.saku < a[0].saku:
            a.append(i.nama)
    return a
a = cari(daftar)
print(a)

```

```

Python 3.8.2 Shell
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_4/3.py ====
[<_main_.mhsTIF object at 0x00000297B97C0D30>, 'Budi', 'Chandra']
>>>

```

4.

```

class mhsTIF():
    def __init__(self, nama, asal, saku):
        self.nama = nama
        self.asal = asal
        self.saku = saku

c0 = mhsTIF('Eka', 'Sukoharjo', 240000)
c1 = mhsTIF('Budi', 'Sragen', 230000)
c2 = mhsTIF('Ahmad', 'Surakarta', 250000)
c3 = mhsTIF('Chandra', 'Surakarta', 235000)
c4 = mhsTIF('Eka', 'Boyolali', 240000)
c5 = mhsTIF('Fandi', 'Salatiga', 250000)
c6 = mhsTIF('Deni', 'Klaten', 245000)
c7 = mhsTIF('Galuh', 'Wonogiri', 245000)
c8 = mhsTIF('Janto', 'Klaten', 245000)
c9 = mhsTIF('Hasan', 'Karanganyar', 270000)
c10 = mhsTIF('Khalid', 'Purwodadi', 265000)

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

def cari (list):
    a = []
    for i in list:
        if i.saku < 250000:
            a.append(i)
    return a
a = cari(daftar)
for i in a:
    print(i.nama)

```

```

Python 3.8.2 Shell
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_4/4.py ====
Eka
Budi
Chandra
Eka
Deni
Galuh
Janto
>>>

```

5.

The screenshot shows a Python IDE with two windows. The left window displays a Python script for a linked list. The right window shows the output of the script.

```

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

    def LinkedList(self, cari):
        curNode = self
        while curNode is not None:
            if curNode.next != None:
                if curNode.data != cari:
                    curNode = curNode.next
                else:
                    print("Nama", cari, "ada dalam linked list")
                    break
            elif curNode.next == None:
                print("Nama", cari, "tidak ada dalam linked list")
                break

a = node('elisa')
menu = a
a.next = node('ayudhia')
a = a.next
a.next = node('berlian')
a = a.next
a.next = node('wulan')

menu.LinkedList('elisa')
menu.LinkedList('hayu')

```

The output window shows the following text:

```

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_4/5.py ====
Name elsa ada dalam linked list
Name hayu tidak ada dalam linked list
>>>

```

6.

The screenshot shows a Python IDE with two windows. The left window displays a Python script for a binary search algorithm. The right window shows the output of the script.

```

def binary(kumpulan, target):
    low = 0
    high = len(kumpulan) - 1

    while low <= high:
        mid = (high + low) // 2
        if kumpulan[mid] == target:
            return "target berada di index " + str(mid)
            break
        elif target < kumpulan[mid]:
            high = mid - 1
        else:
            low = mid + 1

    return False

listnya = [1, 2, 3, 4, 5, 6, 7]
target1 = 2
target2 = 8

print("\nlist", listnya)
print("nilai target =", target1)
print(binary(listnya, target1))

print("\nlist", listnya)
print("nilai target =", target2)
print(binary(listnya, target2))

```

The output window shows the following text:

```

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_4/6.py ====
list [1, 2, 3, 4, 5, 6, 7]
nilai target = 2
target berada di index 1

list [1, 2, 3, 4, 5, 6, 7]
nilai target = 8
False
>>>

```

7.

```

7.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL 4/7.py (3.8.2)
File Edit Format Run Options Window Help

def binary(kumpulan, target):
    low = 0
    high = len(kumpulan) - 1
    listku = []
    while low <= high:
        if kumpulan[low] == target:
            listku.append(low)
            low += 1
        else:
            low += 1
    return listku
x = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 14]
dicari = 6
print('posisi data', 6, 'pada list', x, 'adalah', binary(x, dicari))

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL 4/7.py ====
posisi data 6 pada list [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 14] adalah [3, 4, 5]
>>>

```

8.

```

8.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL 4/8.py (3.8.2)
File Edit Format Run Options Window Help

print("""
Soal :
    Pada permainan tebak angka yang sudah kamu buat di Modul 1
    soal nomor 12, halaman 16) kalau angka yang harus ditebak
    berada diantara 1-100, seharusnya maksimal jumlah tebakan
    adalah 7. Kalau antara 1-1000, maksimal jumlah tebakan
    adalah 10. Mengapa seperti itu? Bagaimana polanya?""")

print("""
Jawab :
    Ada dua kemungkinan pola yang bisa digunakan.
    Misalkan, angka yang akan ditebak adalah 70.
    -POLA PERTAMA-
        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
    -POLA KEDUA-
        menggunakan barisan geometri Sn = 2^n
        barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
        Misal angka yang akan ditebak 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"
""")

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
j on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL 4/8.py ====

Soal :
    Pada permainan tebak angka yang sudah kamu buat di Modul 1
    soal nomor 12, halaman 16) kalau angka yang harus ditebak
    berada diantara 1-100, seharusnya maksimal jumlah tebakan
    adalah 7. Kalau antara 1-1000, maksimal jumlah tebakan
    adalah 10. Mengapa seperti itu? Bagaimana polanya?

Jawab :
    Ada dua kemungkinan pola yang bisa digunakan.
    Misalkan, angka yang akan ditebak adalah 70.
    -POLA PERTAMA-
        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
    -POLA KEDUA-
        menggunakan barisan geometri Sn = 2^n
        barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
        Misal angka yang akan ditebak 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"
>>>

```