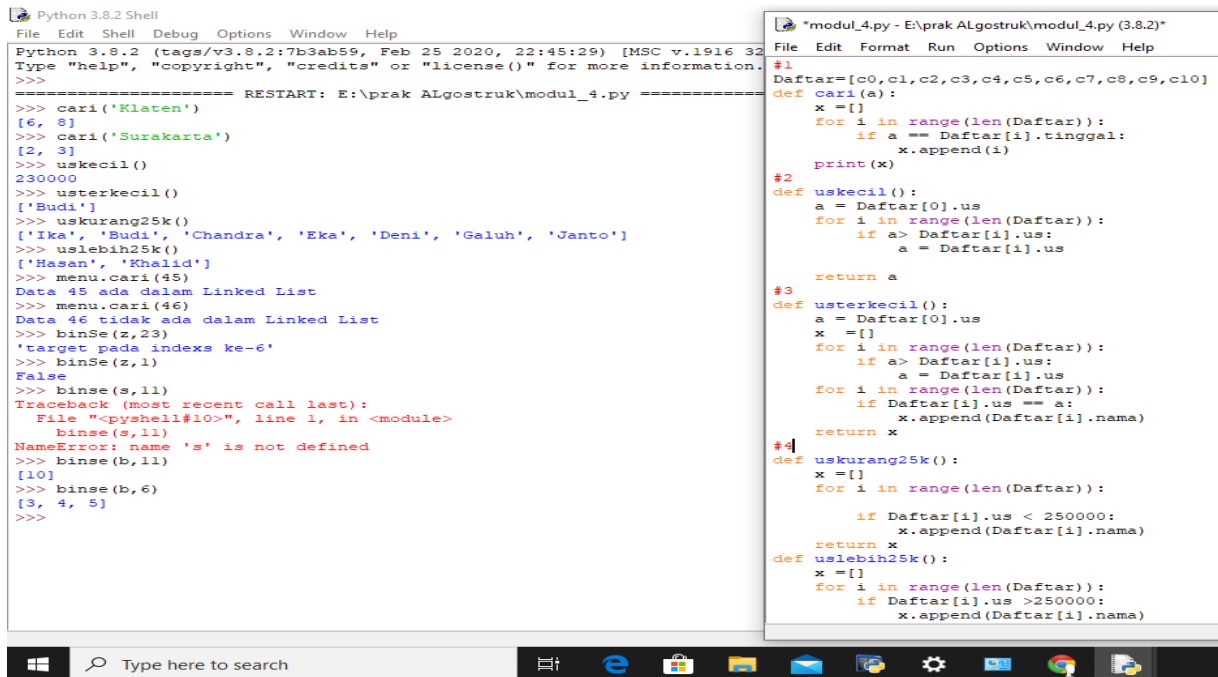


Nama : Nur Fadlilah Azzis

NIM : L200180113

Modul 4

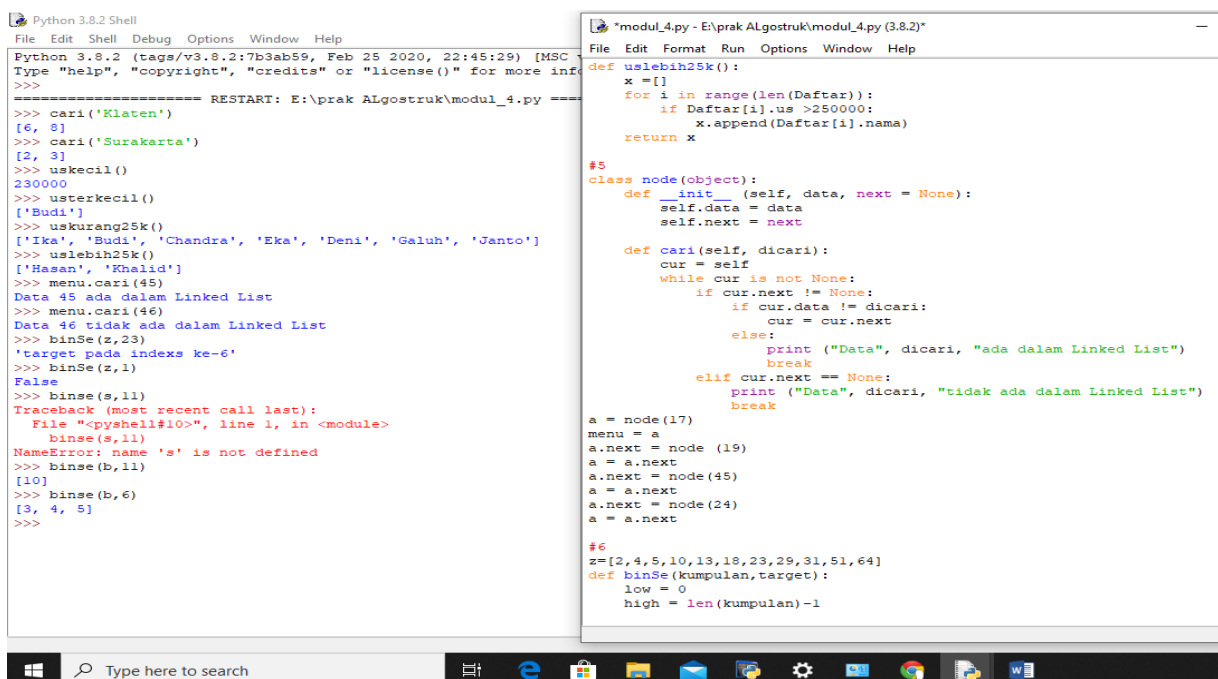
No.1-4



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\prak ALgostruk\modul_4.py =====
>>> cari('Klaten')
[6, 8]
>>> cari('Surakarta')
[2, 3]
>>> uskecil()
230000
>>> usterkecil()
['Budi']
>>> uskurang25k()
['Ika', 'Budi', 'Chandra', 'Eka', 'Deni', 'Galuh', 'Janto']
>>> uslebih25k()
['Hasan', 'Khalid']
>>> menu.cari(45)
Data 45 ada dalam Linked List
>>> menu.cari(46)
Data 46 tidak ada dalam Linked List
>>> binSe(z,23)
'target pada index ke-6'
>>> binSe(z,1)
False
>>> binSe(s,11)
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    binse(s,11)
NameError: name 's' is not defined
>>> binse(b,11)
[10]
>>> binse(b,6)
[3, 4, 5]
>>>

*modul_4.py - E:\prak ALgostruk\modul_4.py (3.8.2)*
File Edit Format Run Options Window Help
#1
Daftar=[c0,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10]
def cari(a):
    x=[]
    for i in range(len(Daftar)):
        if a == Daftar[i].tinggal:
            x.append(i)
    print(x)
#2
def uskecil():
    a = Daftar[0].us
    for i in range(len(Daftar)):
        if a > Daftar[i].us:
            a = Daftar[i].us
    return a
#3
def usterkecil():
    a = Daftar[0].us
    x = []
    for i in range(len(Daftar)):
        if a > Daftar[i].us:
            a = Daftar[i].us
        for i in range(len(Daftar)):
            if Daftar[i].us == a:
                x.append(Daftar[i].nama)
    return x
#4
def uskurang25k():
    x = []
    for i in range(len(Daftar)):
        if Daftar[i].us < 250000:
            x.append(Daftar[i].nama)
    return x
def uslebih25k():
    x = []
    for i in range(len(Daftar)):
        if Daftar[i].us > 250000:
            x.append(Daftar[i].nama)
```

No.5



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\prak ALgostruk\modul_4.py =====
>>> cari('Klaten')
[6, 8]
>>> cari('Surakarta')
[2, 3]
>>> uskecil()
230000
>>> usterkecil()
['Budi']
>>> uskurang25k()
['Ika', 'Budi', 'Chandra', 'Eka', 'Deni', 'Galuh', 'Janto']
>>> uslebih25k()
['Hasan', 'Khalid']
>>> menu.cari(45)
Data 45 ada dalam Linked List
>>> menu.cari(46)
Data 46 tidak ada dalam Linked List
>>> binSe(z,23)
'target pada index ke-6'
>>> binSe(z,1)
False
>>> binse(s,11)
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    binse(s,11)
NameError: name 's' is not defined
>>> binse(b,11)
[10]
>>> binse(b,6)
[3, 4, 5]
>>>

*modul_4.py - E:\prak ALgostruk\modul_4.py (3.8.2)*
File Edit Format Run Options Window Help
def uslebih25k():
    x = []
    for i in range(len(Daftar)):
        if Daftar[i].us > 250000:
            x.append(Daftar[i].nama)
    return x
#5
class node(object):
    def __init__(self, data, next = None):
        self.data = data
        self.next = next
    def cari(self, dicari):
        cur = self
        while cur is not None:
            if cur.next != 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
a = node(17)
menu = a
a.next = node(19)
a = a.next
a.next = node(45)
a = a.next
a.next = node(24)
a = a.next
#6
z=[2,4,5,10,13,18,23,29,31,51,64]
def binSe(kumpulan,target):
    low = 0
    high = len(kumpulan)-1
```

No. 6-7

The image shows two overlapping Python 3.8.2 Shell windows. The left window displays the execution of a script that creates a linked list and attempts a binary search. The right window shows the source code for the script, which includes a linked list structure and a binary search function.

```

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC
Type "help", "copyright", "credits" or "license()" for more info
>>>
===== RESTART: E:\prak ALgostruk\modul_4.py =====
>>> cari('Klaten')
[6, 8]
>>> cari('Surakarta')
[2, 3]
>>> uskecil()
230000
>>> usterkecil()
['Budi']
>>> uskurang25k()
['Ika', 'Budi', 'Chandra', 'Eka', 'Deni', 'Galuh', 'Janto']
>>> uslebin25k()
['Hasan', 'Khalid']
>>> menu.cari(45)
Data 45 ada dalam Linked List
>>> menu.cari(46)
Data 46 tidak ada dalam Linked List
>>> binSe(z,23)
'target pada indexs ke-6'
>>> binSe(z,1)
False
>>> binse(s,11)
Traceback (most recent call last):
  File "<pyshe11#10>", line 1, in <module>
    binse(s,11)
NameError: name 's' is not defined
>>> binse(b,11)
[10]
>>> binse(b,6)
[3, 4, 5]
>>>

Python 3.8.2 Shell
File Edit Format Run Options Window Help
===== RESTART: E:\prak ALgostruk\modul_4.py (3.8.2) =====
>>> break
a = node(17)
menu = a
a.next = node (19)
a = a.next
a.next = node(45)
a = a.next
a.next = node(24)
a = a.next

#6
z=[2,4,5,10,13,18,23,29,31,51,64]
def binSe(kumpulan,target):
    low = 0
    high = len(kumpulan)-1

    x=[]
    while low <=high:
        mid =(high + low)//2
        if kumpulan[mid]==target:
            return 'target pada indexs ke-'+str(mid)

        elif target < kumpulan[mid]:
            high = mid -1
        else:
            low = mid+1
    return False

#7
b = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
def binse(kumpulan,target):
    low = 0
    high = len(kumpulan)-1
    a = []
    while low <= high:
        if kumpulan [low] == target:
            a.append(low)
            low += 1
        else:
            low += 1
    return a
  
```

No.8

The image shows a Python 3.8.2 Shell window with a script that explains the complexity of binary search and calculates the number of guesses needed to find a number using binary search and geometric series.

```

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\prak ALgostruk\modul_4.py =====
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 Sn = 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"

#
>>> |
  
```