

TUGAS MODUL 4

```
MODUL4(TUGAS).py - E:\Praktikum AlgoPro\MODUL4(TUGAS).py (3.8.2)*
Python 3.8.2 Shell

class MhsTIF(object):
    def __init__(self, nama, umur, tinggal, us):
        self.nama = nama
        self.umur = umur
        self.tinggal = tinggal
        self.us = us

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

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

##Nomer1##
def cari(a):
    x=[]
    for i in range(len(Daftar)):
        if a == Daftar[i].tinggal:
            x.append(i)
    print(x)

Nomer2##
def uskecil():
    a = Daftar[0].us
    for i in range(len(Daftar)):
        if a > Daftar[i].us:
            a = Daftar[i].us
    return a

Nomer3##
def usterkecil():
    a = Daftar[0].us
    x=[]
    for i in range(len(Daftar)):
        if a > Daftar[i].us:
            a = Daftar[i].us
            x.append(Daftar[i].nama)
    return x

##Nomer4##
def uskurang25k():
    x=[]
    for i in range(len(Daftar)):
        if Daftar[i].us < 250000:
            x.append(Daftar[i].nama)
    return x

##Nomer5##
def uslebih25k():
    x=[]
    for i in range(len(Daftar)):
        if Daftar[i].us > 250000:
            x.append(Daftar[i].nama)
    return x

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.data != dicari:
                cur = cur.next
            else:
                return cur.data
        return None
```

```
Python 3.8.2 Shell

>>> cari("Klaten")
[6, 8]

>>> uskecil()
230000

>>> usterkecil()
['Budi']

>>> uskurang25k()
['Ika', 'Budi', 'Chandra', 'Eka', 'Deni', 'Galuh', 'Janto']
>>> uslebih25k()
['Hasan', 'Khalid']

>>> binSe(z,29)
'target pada index ke-7'
>>> binSe(z,3)
False

>>> binSe(b,8)
Traceback (most recent call last):
  File "pyshell#7>", line 1, in <module>
    binSe(b,8)
NameError: name 'binSe' is not defined

>>> binSe(b,8)
[6]
```

```
*MODUL4(TUGAS).py - E:\Praktikum AlgoPro\MODUL4(TUGAS).py (3.8.2)*
File Edit Format Run Options Window Help

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
menu.cari (17)
menu.cari (22)

##Nomer6##
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 index ke-'+str(mid)

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

##Nomer7##
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

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:\Praktikum AlgoPro\MODUL4(TUGAS).py =====
>>> cari ("Klaten")
[6, 8]
>>>
===== RESTART: E:\Praktikum AlgoPro\MODUL4(TUGAS).py =====
>>> uskecil()
230000
>>>
===== RESTART: E:\Praktikum AlgoPro\MODUL4(TUGAS).py =====
>>> usterkecil()
['Budi']
>>>
===== RESTART: E:\Praktikum AlgoPro\MODUL4(TUGAS).py =====
>>> uskurang25k()
['Ika', 'Budi', 'Chandra', 'Eka', 'Deni', 'Galuh', 'Janto']
>>> uslebih25k()
['Hasan', 'Khalid']
>>>
===== RESTART: E:\Praktikum AlgoPro\MODUL4(TUGAS).py =====
>>> Data 17 ada dalam Linked List
>>> Data 22 tidak ada dalam Linked List
>>>
===== RESTART: E:\Praktikum AlgoPro\MODUL4(TUGAS).py =====
>>> binSe(z,29)
'target pada index ke-7'
>>> binSe(z,3)
False
>>>
===== RESTART: E:\Praktikum AlgoPro\MODUL4(TUGAS).py =====
>>> binSe(b,8)
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    binSe(b,8)
NameError: name 'binSe' is not defined
>>> binse(b,8)
[6]
>>>
```

```
*MODUL4(TUGAS).py - E:\Praktikum AlgoPro\MODUL4(TUGAS).py (3.8.2)*
File Edit Format Run Options Window Help

low += 1
else:
    low += 1
return a

##Nomer8##
print(
    """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 =
    Di mana log berasal dari pangkat log berbasis 2. Dengan begitu dapat mengetahui
    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"
    """)

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

===== RESTART: E:\Praktikum AlgoPro\MODUL4(TUGAS).py =====
>>> binse(b,6)
[3, 4, 5]
>>>
===== RESTART: E:\Praktikum AlgoPro\MODUL4(TUGAS).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 =
>>> 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"
>>>
```