

Nama : Defa Raffy Z.R

NIM : L200180068

Kelas : C

## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

### MODUL 4

#### Soal - Soal untuk Mahasiswa

1. Buatlah suatu fungsi pencarian yang alih-alih mengembalikan True/False, mengembalikan semua index lokasi elemen yang dicari. Jadi, misal pada list daftar mahasiswa di halaman 40 kita mencari mahasiswa yang berasal dari Klaten, kita akan mendapatkan [6,8]. Kalau yang dicari tidak ditemukan, fungsi ini akan mengembalikan list kosong.

Jawab:

```
L200180068_Algoritma_Modul4_Tugas.py - D:\AHQ\Semester 4\Laporan praktikum\ASD\Modul 4\L200180068_Algoritma_Modul4_Tugas.py (3.7.0)
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("Ika", 10, "Sukoharjo", 240000)
c1 = MhsTIF("Budi", 51, "Sragen", 230000)
c2 = MhsTIF("Adnan", 2, "Surakarta", 250000)
c3 = MhsTIF("Chandra", 18, "Surakarta", 245000)
c4 = MhsTIF("Eka", 4, "Boyolali", 240000)
c5 = MhsTIF("Fandi", 31, "Salatiga", 250000)
c6 = MhsTIF("Hendi", 13, "Klaten", 245000)
c7 = MhsTIF("Galuh", 5, "Wonogiri", 245000)
c8 = MhsTIF("Jessa", 23, "Klaten", 245000)
c9 = MhsTIF("Nessa", 64, "Purwokerto", 270000)
c10 = MhsTIF("Khaidi", 29, "Purwodadi", 245000)

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

##: Fungsi pencarian yang mengembalikan semua index lokasi elemen yang dicari dan
## mengembalikan list kosong jika tidak ditemukan

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

a = cariKotaTinggal(Daftar, "Klaten")
print(a)
```

```
class MhsTIF(object):
```

```
    def __init__(self, nama, nim, kota, uangsaku):
```

```
        self.nama = nama
```

```
        self.nim = nim
```

```
        self.kotaTinggal = kota
```

```
        self.uangSaku = uangsaku
```

```
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]
```

## 1. Fungsi pencarian yang mengembalikan semua index lokasi elemen yang dicari  
 ##   mengembalikan list kosong bila tidak ditemukan

```
def cariKotaTinggal(list, target):
```

```
    a = []
```

```
    for i in list :
```

```
        if i.kotaTinggal == target:
```

```
            a.append(list.index(i))
```

```
    return a
```

```
a = cariKotaTinggal(Daftar, "Klaten")
```

```
print(a)
```



```

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/3.7.0:01b2f90000, Jun 27 2019, 06:06:47) [AMD64] on win32
Type "copyright", "credits()" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ARD\Modul 4\1200180080_Algoritma Modul 4_Tugas.py
[6, 8]
>>> a = cariKotaTinggal (Daftar, "Surakarta")
>>> print (a)
[2, 3]
>>> a = cariKotaTinggal (Daftar, "Klaten")
>>> print (a)
[6, 8]
>>>

```

2. Dari list daftar mahasiswa diatas, buatlah fungsi untuk menemukan uang saku yang terkecil di antara mereka

**Jawab:**

```

L20018000_Algoritma_Modul4_Tugas.py - ID:AFYQ Semester 4 Laporan praktikum ASD/Modul 4/L20018000_Algoritma_Modul4_Tugas.py (2.7.0)
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("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]

### 2. Fungsi untuk menemukan uang saku yang terkecil

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

a = cariUangSakuTerkecil(Daftar)
print(a)

```

class MhsTIF(object):

def \_\_init\_\_(self, nama, nim, kota, uangsaku):

self.nama = nama

self.nim = nim

self.kotaTinggal = kota

self.uangSaku = uangsaku

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]

#### 2. Fungsi untuk menemukan uang saku yang

terkecil def cariUangSakuTerkecil(list):

```
temp = list[0].uangSaku
```

```
for i in list[1:]:
```

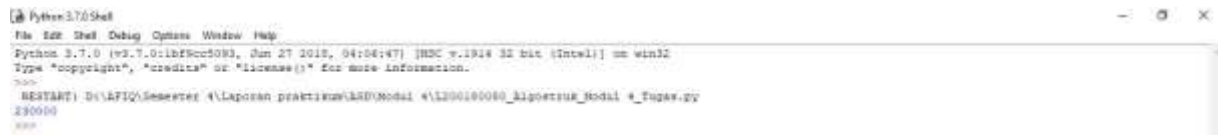
```
    if i.uangSaku < temp:
```

```
        temp = i.uangSaku
```

```
return temp
```

```
a = cariUangSakuTerkecil(Daftar)
```

```
print(a)
```



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/v3.7.0:bfecf50b3, Jun 27 2018, 04:06:47) [AMD64] on win32
Type "copyright", "credits()" or "license()" for more
>>>
RESTART: C:\AFQ\Semester 4\Laporan praktikum\ASD\Modul 4\L200100000_Algoritma_Modul 4_Tugas.py
240000
>>>
```

3. Ubah program diatas agar mengembalikan objek mahasiswa yang mempunyai uang saku terkecil. Jika ada lebih dari satu mahasiswa yang uang sakunya terkecil, semua objek objek mahasiswa itu dikembalikan

**Jawab:**



```
L200100000_Algoritma_Modul 4_Tugas.py - C:\AFQ\Semester 4\Laporan praktikum\ASD\Modul 4\L200100000_Algoritma_Modul 4_Tugas.py (3.7.0)
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("Ika", 10, "Sukoharjo", 240000)
c1 = MhsTIF("Budi", 51, "Bojonegara", 230000)
c2 = MhsTIF("Ahmad", 2, "Durenkerta", 250000)
c3 = MhsTIF("Chandra", 16, "Durenkerta", 135000)
c4 = MhsTIF("Ika", 4, "Korpri", 240000)
c5 = MhsTIF("Rani", 31, "Korpri", 280000)
c6 = MhsTIF("Rani", 13, "Korpri", 240000)
c7 = MhsTIF("Rani", 9, "Korpri", 240000)
c8 = MhsTIF("Rani", 23, "Korpri", 240000)
c9 = MhsTIF("Rani", 44, "Korpri", 270000)
c10 = MhsTIF("Rani", 29, "Korpri", 280000)

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

#### 1. Program mengembalikan objek mahasiswa yang mempunyai uang saku
#### terkecil

def cariUangSakuTerkecilObjek(list):
    temp = list[0].uangSaku
    obj = list[0].nama
    for i in list[1:]:
        if i.uangSaku < temp:
            temp = i.uangSaku
            obj = i.nama
        elif i.uangSaku == temp:
            temp.append(i)
            obj.append(i)
    return obj

a = cariUangSakuTerkecilObjek(Daftar)
print(a)
```

```
class MhsTIF(object):
```

```
    def __init__(self, nama, nim, kota, uangsaku):
```

```
        self.nama = nama
```

```
        self.nim = nim
```

```
        self.kotaTinggal = kota
```

```
        self.uangSaku = uangsaku
```

```
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]
```

#### 3. Program mengembalikan objek mahasiswa yang mempunyai uang saku

#### terkecil

```
def cariUangSakuTerkecilObject(list):
```

```
    temp = list[0].uangSaku
```

```
    obj = list[0].nama
```

```
    for i in list[1:]:
```

```
        if i.uangSaku < temp:
```

```
            temp = i.uangSaku
```

```
            obj = i.nama
```

```
        elif i.uangSaku == temp:
```

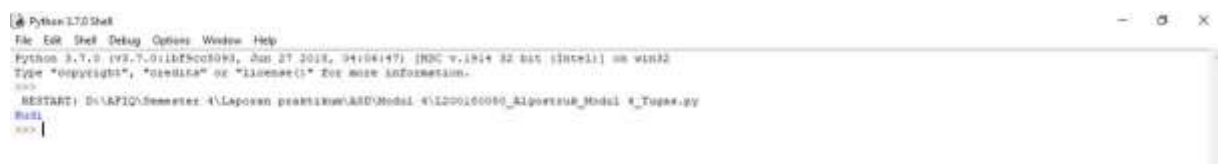
```
            temp.append(i)
```

```
            obj.append(i)
```

```
    return obj
```

```
a = cariUangSakuTerkecilObject(Daftar)
```

```
print(a)
```



4. Buatlah suatu fungsi yang mengembalikan semua objek mahasiswa yang uang sakunya kurang dari 250000

**Jawab:**

```
"L20180080_Algoritma_Modul4_Tugas.py" - D:\APQ\Semester 4\Laporan praktikum\ASD\Modul 4\L20180080_Algoritma_Modul4_Tugas.py (3.7.0)
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

m0 = MhsTIF("Ika", 10, "Sukoharjo", 240000)
m1 = MhsTIF("Budi", 51, "Sragen", 230000)
m2 = MhsTIF("Ahmad", 2, "Surakarta", 250000)
m3 = MhsTIF("Chandra", 18, "Surakarta", 235000)
m4 = MhsTIF("Eka", 4, "Boyolali", 240000)
m5 = MhsTIF("Fandi", 31, "Salatiga", 250000)
m6 = MhsTIF("Deni", 13, "Klaten", 245000)
m7 = MhsTIF("Galuh", 5, "Wonogiri", 245000)
m8 = MhsTIF("Janto", 23, "Klaten", 245000)
m9 = MhsTIF("Hasan", 64, "Karanganyar", 270000)
m10 = MhsTIF("Khalid", 29, "Purwodadi", 265000)

Daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]

### 1. Fungsi untuk mengembalikan semua objek mahasiswa yang uang sakunya
### kurang dari 250000

def cariUangSakuKurang250k(lis):
    temp = []
    for i in lis:
        if i.uangsaku < 250000:
            temp.append(i)
    return temp

x = cariUangSakuKurang250k(Daftar)
for i in x:
    print(i.nama)
```

class MhsTIF(object):

def \_\_init\_\_(self, nama, nim, kota, uangsaku):

self.nama = nama

self.nim = nim

self.kotaTinggal = kota

self.uangsaku = uangsaku

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]



#### 4. Fungsi untuk mengembalikan semua objek mahasiswa yang uang sakunya  
#### kurang dari 250000

```
def cariUangSakuKurang250k(list):
```

```
    temp = []
```

```
    for i in list:
```

```
        if i.uangSaku < 250000:
```

```
            temp.append(i)
```

```
    return temp
```

```
a = cariUangSakuKurang250k(Daftar)
```

```
for i in a:
```

```
    print(i.nama)
```



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/3.7.0:0000000, Jun 27 2019, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 4\120180080_Algoritma_Modul 4_Tugas.py
Eka
Budi
Chandre
Eka
Irena
Galuh
Janto
>>>
```

5. Buatlah suatu program untuk mencari suatu item di sebuah linked list

**Jawab:**



```
120180080_Algoritma_Modul 4_Tugas.py - D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 4\120180080_Algoritma_Modul 4_Tugas.py (3.7.0)
File Edit Shell Debug Options Window Help
## 5. PROGRAM UNTUK MENCAI SUATU ITEM DI SEBUAH LINKED LIST
class Node(object):
    def __init__(self, data, next = None):
        self.data = data
        self.next = next

    def cariLinkedList(self, dicari):
        curNode = self
        while curNode != None:
            if curNode.data == dicari:
                curNode = curNode.next
            else:
                print("Data", dicari, "ada dalam linked list")
                return
        print("Data", dicari, "tidak ada dalam linked list")
        return

a = Node(12)
b = a
a.next = Node(34)
a = a.next
a.next = Node(10)
a = a.next
a.next = Node(45)

meny.cariLinkedList(10)
meny.cariLinkedList(110)
```

#### 5. Program untuk mencari suatu item di sebuah linked

list class node(object):

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

```
self.next = next
```

```
def cariLinkedList(self, dicari):
```

```
    curNode = self
```

```
    while curNode is not None:
```

```
        if curNode.next != None:
```

```
            if curNode.data != dicari:
```

```
                curNode = curNode.next
```

```
            else:
```

```
                print("Data", dicari, "ada dalam linked list")
```

```
                break
```

```
        elif curNode.next == None:
```

```
            print ("Data", dicari, "tidak ada dalam linked list")
```

```
            break
```



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/3.7.0:01b1880000, Jun 27 2018, 04:04:47) [AMD64 (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 4\1200180080_Algoritma_Modul 4_Tugas.py
>>> a = node(10)
>>> mada = a
>>> a.next = node(20)
>>> a = a.next
>>> a.next = node(30)
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    a.next = node(10)
NameError: name 'node' is not defined
>>> a.next = node(10)
>>> a = a.next
>>> a.next = node(40)
Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
    a.next = node(40)
TypeError: cannot unpack non-iterable node object
>>> a.next = node(40)
>>> mada.cariLinkedList()
Data 10 ada dalam linked list
>>> mada.cariLinkedList(110)
Data 110 tidak ada dalam linked list
>>> |
```

6. Binary search. Ubahlah fungsi binSe di halaman 43 agar mengembalikan index lokasi elemen yang ditemukan. Kalau tidak ketemu, akan mengembalikan False

**Jawab:**



```
"1200180080_Algoritma_Modul 4_Tugas.py" - D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 4\1200180080_Algoritma_Modul 4_Tugas.py (3.7.0)
File Edit Format Run Options Window Help
## 6. Fungsi binSe mengembalikan index lokasi elemen yang ditentukan,
## kalau tidak ketemu mengembalikan False

def binSe(himpunan, target):
    low = 0
    high = len(himpunan)-1
    while low <= high:
        mid = (high+low)//2
        if himpunan[mid] == target:
            return mid
        elif target < himpunan[mid]:
            high = mid-1
        else:
            low = mid+1
    return False

himpunan = [2, 4, 5, 10, 15, 18, 21, 29, 31, 33, 35]
print(binSe(himpunan, 35))
```

**## 6.** Fungsi binSe mengembalikan index lokasi elemen yang ditentukan,

**##** kalau tidak ketemu mengembalikan False

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



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/v3.7.0:0b3800093, Jun 27 2019, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits()" or "license()" for more information.
>>>
RESTART: D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 4\1200100080_Algoritma_Modul 4_Tugas.py
>>> kumpulan = [2, 4, 3, 10, 13, 18, 23, 29, 31, 32, 44]
>>> print(binSe(kumpulan, 2))
2
>>> print(binSe(kumpulan, 0))
0
>>> print(binSe(kumpulan, 6))
6
>>> print(binSe(kumpulan, 30))
False
>>> print(binSe(kumpulan, 44))
False
>>> |
```

7. Binary search. Ubahlah fungsi binSe itu agar mengembalikan semua index lokasi elemen yang ditemukan. Contoh : mencari angka 6 pada list [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14] akan mengembalikan [3, 4, 5]. Karena sudah urut, tinggal melihat kiri dan kanannya.

**Jawab:**



```
"1200100080_Algoritma_Modul 4_Tugas.py" - D:\AFIQ\Semester 4\Laporan praktikum\ASD\Modul 4\1200100080_Algoritma_Modul 4_Tugas.py (3.7.0)
File Edit Format Run Options Window Help
## 7. Fungsi binSeAll mengembalikan semua index lokasi elemen yg ditemukan

def binSeAll(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
    return False

kumpulan = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
print(binSeAll(kumpulan, 6))
|
```

#### 7. Fungsi binSe mengembalikan semua index lokasi elemen yg ditemukan

```

def binSeAll(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
    return False

```

```

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/b3c5093, Jun 27 2018, 04:06:47) [AMD64] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: In:\AFIQ\Semester 4\Laporan praktikum\AIIM\Modul 4\1200180060_Algortrak_Modul 4_Tugas.py
>>> kumpulan = [2, 3, 8, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 14]
>>> print(binSeAll (kumpulan, 8))
[8, 9]
>>> print(binSeAll (kumpulan, 9))
[9]
>>> print(binSeAll (kumpulan, 12))
[12, 13]
>>> print(binSeAll (kumpulan, 2))
[]
>>>

```

8. Pada permainan tebak angka yang sudah kamu buat di modul 1 (soal nomor 12, halaman 16), kalua angka yang harus ditebak berada di antara 1 dan 100, seharusnya maksimal jumlah tebakan adalah 7. Kalau antara 1 dan 1000, maksimal jumlah tebakan adalah 10. Mengapa seperti itu ? Bagaimanakah polanya ?

**Jawab:**

Ada dua kemungkinan pola bisa digunakan.

Misalkan, angka yang akan ditebak adalah 70.



Pola pertama

$a = \text{nilai tebakan pertama} // 2$

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  $S_n = 2^n$  Barisan

yang terjadi adalah 2, 4, 8, 16, 32, 64 Misal

angka yang akan ditebak adalah 70 Tebakan ke-

1 : 64 => Jawaban = "Lebih dari itu"

Tebakan ke-2 : 96 (dari 64 + 32) => Jawaban = "Kurang dari itu"

Tebakan ke-3 : 80 (dari 64 + 16) => Jawaban = "Kurang dari itu"

Tebakan ke-4 : 72 (dari 64 + 8) => Jawaban = "Kurang dari itu"

Tebakan ke-5 : 68 (dari 64 + 4) => Jawaban = "Lebih dari itu"

Tebakan ke-6 : 70 (dari 68 + 2) => Jawaban = "TEPAT"