

Nama : MuchFatan Rahmadan
NIM : L200180061
Kelas : C

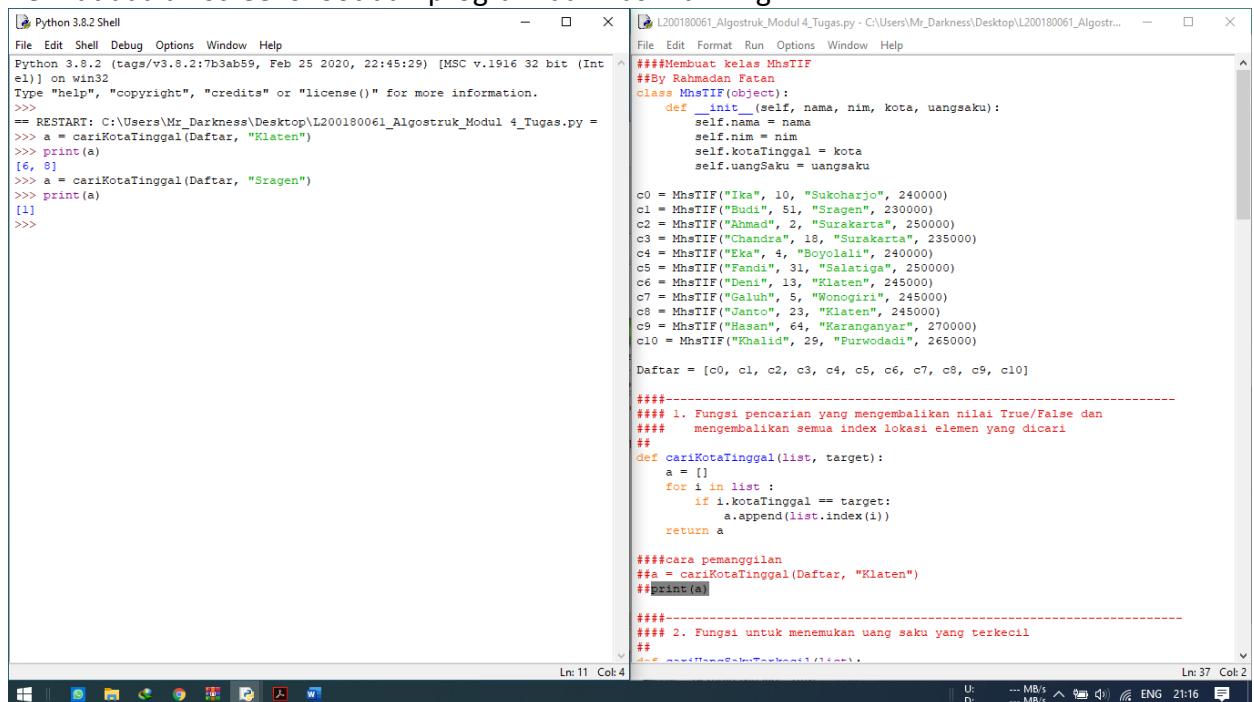
Modul 4 Pencarian

Tugas

Soal-soal untuk mahasiswa

1. Membuat suatu fungsi pencarian yang mengembalikan nilai True/False dan mengembalikan semua index lokasi elemen yang dicari pada kelas MhsTIF

Berikut adalah screenshoot dari program dan hasil running:



```
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 (Int
el)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 4_Tugas.py =
>>> a = cariKotaTinggal(Daftar, "Klaten")
>>> print(a)
[6, 8]
>>> a = cariKotaTinggal(Daftar, "Sragen")
>>> print(a)
[1]
>>>
```

```
#####Membuat kelas MhsTIF
##By Rahmadan Fatan
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 nilai True/False dan
### mengembalikan semua index lokasi elemen yang dicari
##
def cariKotaTinggal(list, target):
    a = []
    for i in list:
        if i.kotaTinggal == target:
            a.append(list.index(i))
    return a

####cara pemanggilan
#a = cariKotaTinggal(Daftar, "Klaten")
#print(a)

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

2. Membuat suatu fungsi untuk menemukan uang saku yang terkecil dari kelas MhsTIF

Berikut adalah screenshoot dari program dan hasil running:

The image shows two side-by-side windows from a Python IDE. The left window is a Python 3.8.2 Shell with the following code:

```
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: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 4_Tugas.py =
>>> a = cariUangSakuTerkecil(Daftar)
>>> print(a)
230000
>>>
```

The right window shows the source code for the program:

```
L200180061_Algostruk_Modul 4_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostr...
File Edit Format Run Options Window Help
#####
### 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

#####cara pemanggilan
#### a = cariUangSakuTerkecil(Daftar)
#### print(a)

#####
### 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

#####cara pemanggilan
#### a = cariUangSakuTerkecilObject(Daftar)
#### print(a)

#####
### 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

#####cara pemanggilan
#### a = cariUangSakuKurang250k(Daftar)
#### print(a)

#####
```

3. Mengubah program no 2 agar mengembalikan objek mahasiswa yang mempunyai uang saku terkecil

Berikut adalah screenshoot dari program dan hasil running:

The image shows two side-by-side windows from a Python IDE. The left window is a Python 3.8.2 Shell with the following code:

```
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: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 4_Tugas.py =
>>> a = cariUangSakuTerkecilObject(Daftar)
>>> print(a)
Budi
>>> |
```

The right window shows the modified source code:

```
L200180061_Algostruk_Modul 4_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostr...
File Edit Format Run Options Window Help
temp = i.uangSaku
return temp

#####cara pemanggilan
#### a = cariUangSakuTerkecil(Daftar)
#### print(a)

#####
### 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

#####cara pemanggilan
#### a = cariUangSakuTerkecilObject(Daftar)
#### print(a)

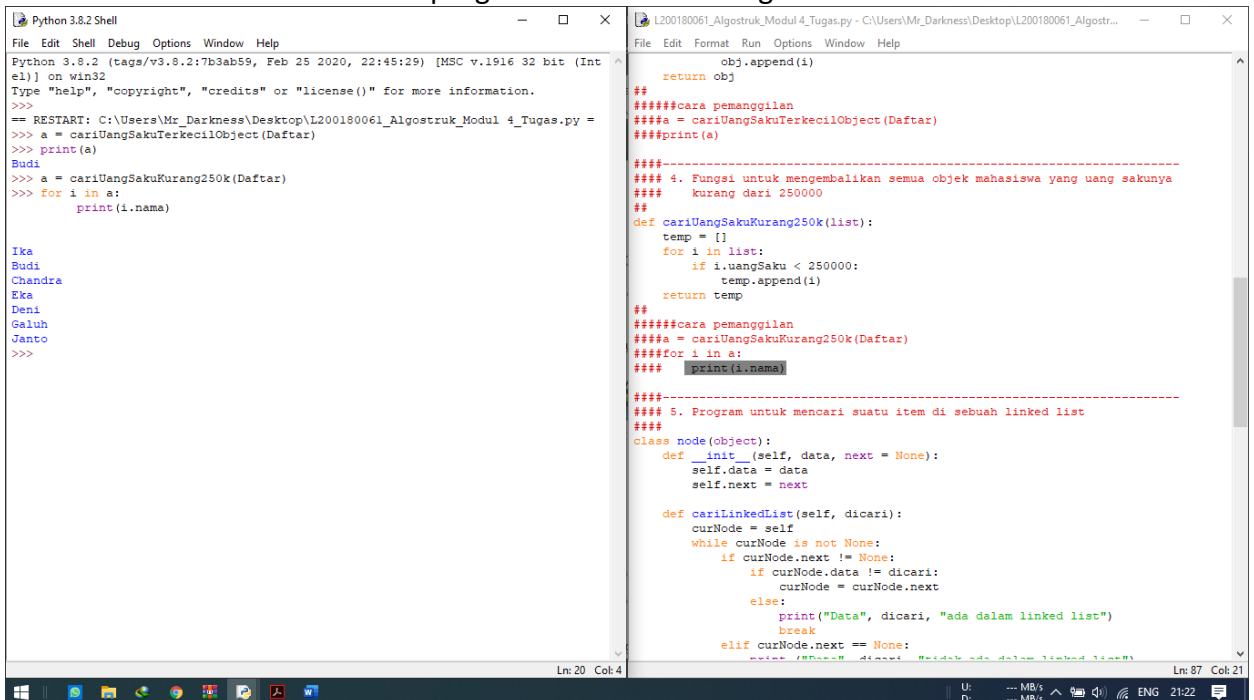
#####
### 4. Fungsi untuk mengembalikan semua objek mahasiswa yang uang sakunya kurang dari 250000
###
def cariUangSakuKurang250k(list):
    temp = []
    for i in list:
        if i.uangSaku < 250000:
            temp.append(i)
    return temp

#####cara pemanggilan
#### a = cariUangSakuKurang250k(Daftar)
#### print(a)

#####
```

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

Berikut adalah screenshot dari program dan hasil running:



```
Python 3.8.2 Shell
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: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 4_Tugas.py =
>>> a = cariUangSakuTerkecilObject(Daftar)
>>> print(a)
Budi
>>> a = cariUangSakuKurang250k(Daftar)
>>> for i in a:
>>>     print(i.nama)

Ika
Budi
Chandra
Eka
Deni
Galuh
Janto
>>>
```

```
L200180061_Algostruk_Modul 4_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostr...
File Edit Format Run Options Window Help

obj.append(i)
return obj

####cara pemanggilan
####a = cariUangSakuTerkecilObject(Daftar)
####print(a)

#####-----
#### 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

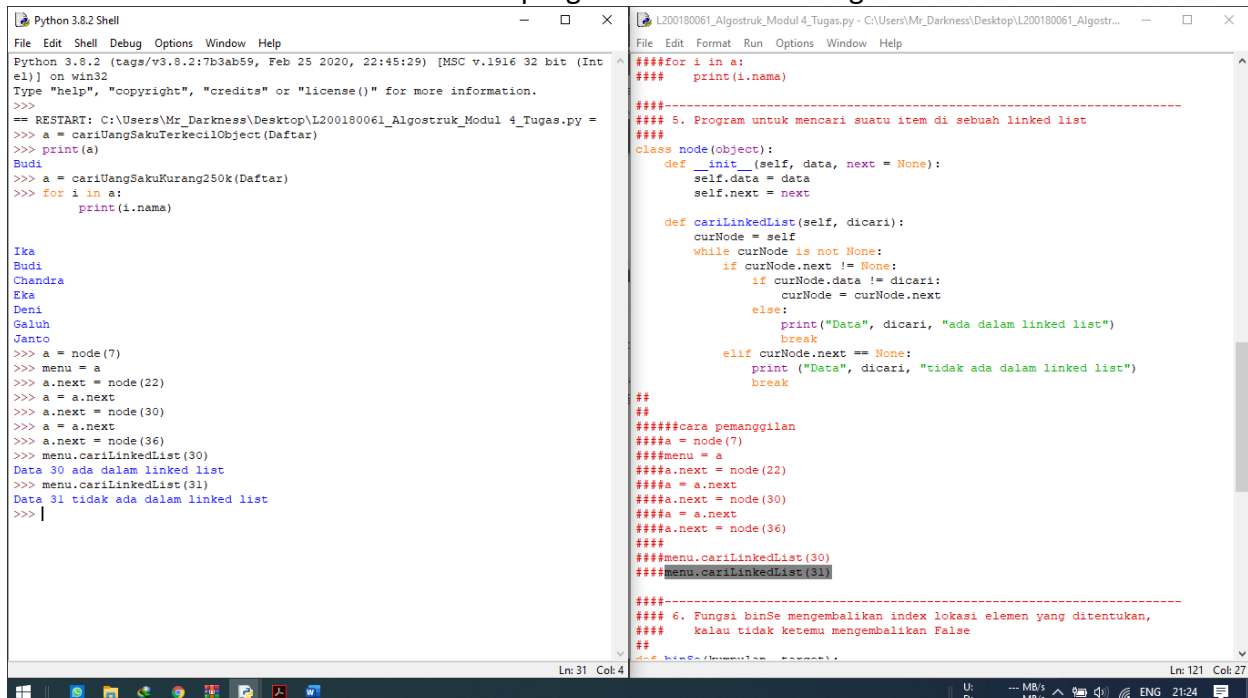
####
####cara pemanggilan
####a = cariUangSakuKurang250k(Daftar)
####for i in a:
####    print(i.nama)

#####-----
#### 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
```

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

Berikut adalah screenshot dari program dan hasil running:



```
Python 3.8.2 Shell
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: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul 4_Tugas.py =
>>> a = cariUangSakuTerkecilObject(Daftar)
>>> print(a)
Budi
>>> a = cariUangSakuKurang250k(Daftar)
>>> for i in a:
>>>     print(i.nama)

Ika
Budi
Chandra
Eka
Deni
Galuh
Janto
>>> a = node(7)
>>> menu = a
>>> a.next = node(22)
>>> a = a.next
>>> a.next = node(30)
>>> a = a.next
>>> a.next = node(36)
>>> menu.cariLinkedList(30)
Data 30 ada dalam linked list
>>> menu.cariLinkedList(31)
Data 31 tidak ada dalam linked list
>>>
```

```
L200180061_Algostruk_Modul 4_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostr...
File Edit Format Run Options Window Help

####for i in a:
####    print(i.nama)

#####-----
#### 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

####
####
####cara pemanggilan
####a = node(7)
####menu = a
####a.next = node(22)
####a = a.next
####a.next = node(30)
####a = a.next
####a.next = node(36)
####
####menu.cariLinkedList(30)
####menu.cariLinkedList(31)

#####-----
#### 6. Fungsi binSe mengembalikan index lokasi elemen yang ditentukan,
#### kalau tidak ketemu mengembalikan False
####
def binSe(menemukan, array):
```

6. Mengubah fungsi binSe pada halaman 43 agar mengembalikan index lokasi elemen yang ditentukan, kalau tidak ketemu mengembalikan False

Berikut adalah screenshoot dari program dan hasil running:

The screenshot shows two windows. The left window is a Python 3.8.2 Shell with the following code and output:

```
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: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul_4_Tugas.py =
>>> a = cariUangSakuTerkecilObject(Daftar)
>>> print(a)
Budi
>>> a = cariUangSakuKurang250k(Daftar)
>>> for i in a:
>>>     print(i.nama)

Ika
Budi
Chandra
Eka
Deni
Galuh
Janto
>>> a = node(7)
>>> menu = a
>>> a.next = node(22)
>>> a = a.next
>>> a.next = node(30)
>>> a = a.next
>>> a.next = node(36)
>>> menu.cariLinkedList(30)
Data 30 ada dalam linked list
>>> menu.cariLinkedList(31)
Data 31 tidak ada dalam linked list
>>> kumpulan = [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>> print(binSe(kumpulan, 5))
2
>>> |
```

The right window is an IDE showing the implementation of the `binSe` function:

```
L200180061_Algostruk_Modul_4_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul_4_Tugas.py
File Edit Format Run Options Window Help

#####cara pemanggilan
####a = node(7)
####menu = a
####a.next = node(22)
####a = a.next
####a.next = node(30)
####a = a.next
####a.next = node(36)
#####
#####menu.cariLinkedList(30)
#####menu.cariLinkedList(31)

#####
##### 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

####kumpulan = [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
####print(binSe(kumpulan, 5))

#####
##### 7. Fungsi binSe mengembalikan semua index lokasi elemen yg ditentukan
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. Mengubah fungsi binSe agar mengembalikan semua index lokasi elemen yang ditentukan

Berikut adalah screenshoot dari program dan hasil running:

The screenshot shows two windows. The left window is a Python 3.8.2 Shell with the following code and output:

```
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: C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul_4_Tugas.py =
>>> a = cariUangSakuTerkecilObject(Daftar)
>>> print(a)
Budi
>>> a = cariUangSakuKurang250k(Daftar)
>>> for i in a:
>>>     print(i.nama)

Ika
Budi
Chandra
Eka
Deni
Galuh
Janto
>>> a = node(7)
>>> menu = a
>>> a.next = node(22)
>>> a = a.next
>>> a.next = node(30)
>>> a = a.next
>>> a.next = node(36)
>>> menu.cariLinkedList(30)
Data 30 ada dalam linked list
>>> menu.cariLinkedList(31)
Data 31 tidak ada dalam linked list
>>> kumpulan = [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>> print(binSe(kumpulan, 5))
2
>>> kumpulan = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
>>> print(binSeAll(kumpulan, 6))
[3, 4, 5]
>>> |
```

The right window is an IDE showing the implementation of the `binSeAll` function:

```
L200180061_Algostruk_Modul_4_Tugas.py - C:\Users\Mr_Darkness\Desktop\L200180061_Algostruk_Modul_4_Tugas.py
File Edit Format Run Options Window Help

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

####kumpulan = [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
####print(binSe(kumpulan, 5))

#####
##### 7. Fungsi binSe mengembalikan semua index lokasi elemen yg ditentukan
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))
```

8. Pada permainan tebak angka yang sudah saya buat di modul 1 (soal nomor 12, halaman 16), kalau 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 ?

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”