

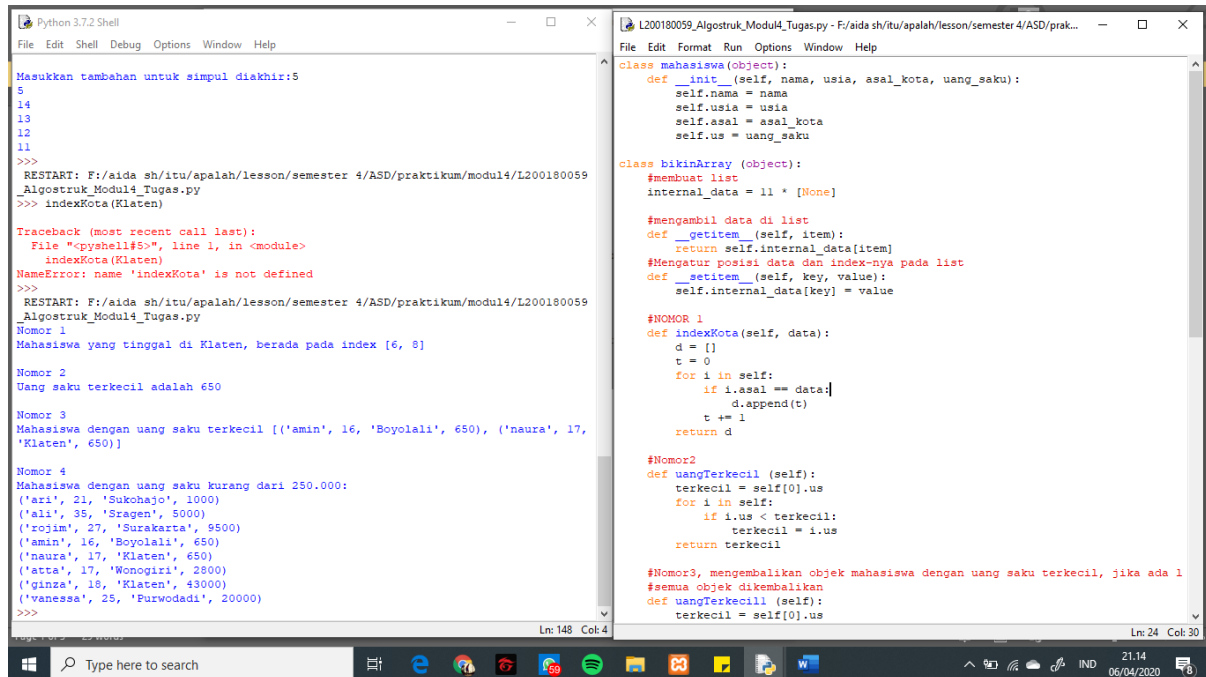
Nama : Abid Muhammad Taufiq

NIM : L200180059

Kelas : C

Modul 4

1. Tugas 1-4



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

Masukkan tambahan untuk simpul diakhir:5
5
14
13
12
11
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
>>> indexKota(Klaten)

Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    indexKota(Klaten)
NameError: name 'indexKota' is not defined
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
>>>
Nomor 1
Mahasiswa yang tinggal di Klaten, berada pada index [6, 8]

Nomor 2
Uang saku terkecil adalah 650

Nomor 3
Mahasiswa dengan uang saku terkecil [('amin', 16, 'Boyolali', 650), ('naura', 17, 'Klaten', 650)]

Nomor 4
Mahasiswa dengan uang saku kurang dari 250.000:
('ari', 21, 'Sukoharjo', 1000)
('ali', 35, 'Sragen', 5000)
('rojim', 27, 'Surakarta', 9500)
('amin', 16, 'Boyolali', 650)
('naura', 17, 'Klaten', 650)
('atta', 17, 'Wonogiri', 2800)
('ginza', 18, 'Klaten', 43000)
('vanessa', 25, 'Purwodadi', 20000)
>>>

L200180059_Algostruk_Modul4_Tugas.py - F:/aida sh/itu/apalah/lesson/semester 4/ASD/prak...
File Edit Format Run Options Window Help

class mahasiswa(object):
    def __init__(self, nama, usia, asal_kota, uang_saku):
        self.nama = nama
        self.usia = usia
        self.asal = asal_kota
        self.us = uang_saku

class bikinArray (object):
    #membuat list
    internal_data = 11 * [None]

    #mengambil data di list
    def __getitem__(self, item):
        return self.internal_data[item]

    #Mengatur posisi data dan index-nya pada list
    def __setitem__(self, key, value):
        self.internal_data[key] = value

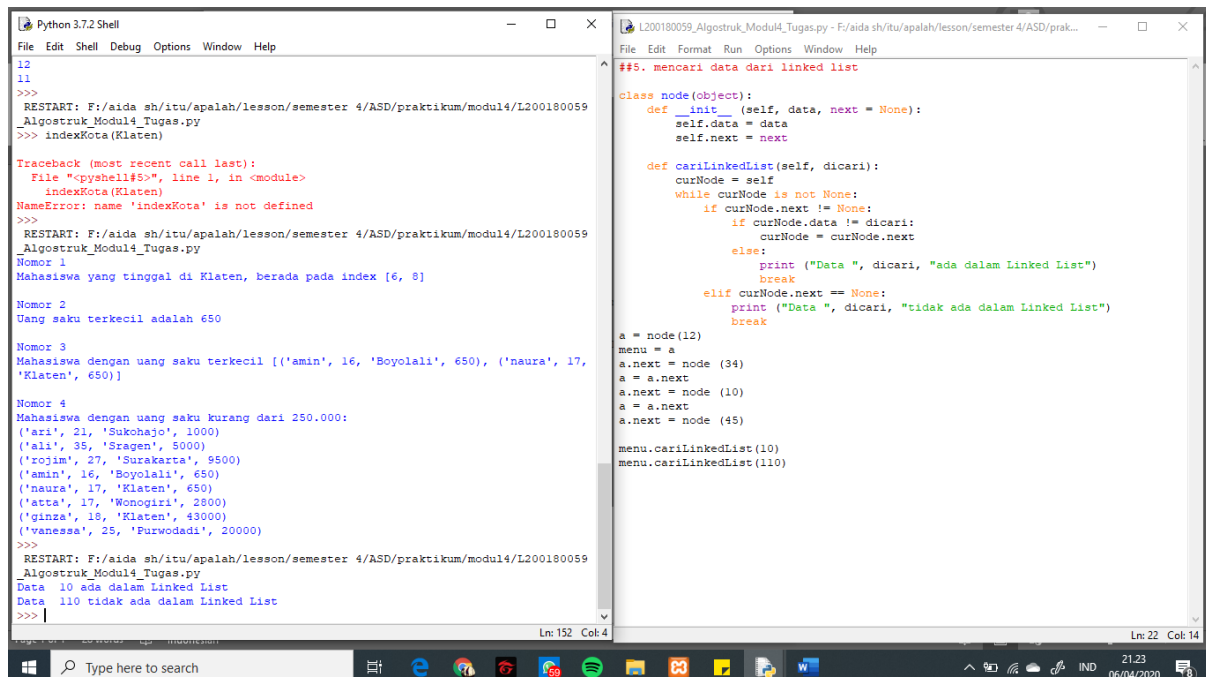
    #Nomor 1
    def indexKota(self, data):
        d = []
        t = 0
        for i in self:
            if i.asal == data:
                d.append(t)
            t += 1
        return d

    #Nomor2
    def uangTerkecil (self):
        terkecil = self[0].us
        for i in self:
            if i.us < terkecil:
                terkecil = i.us
        return terkecil

    #Nomor3, mengembalikan objek mahasiswa dengan uang saku terkecil, jika ada 1
    #semua objek dikembalikan
    def uangTerkecil (self):
        terkecil = self[0].us

Ln: 148 Col: 4
Ln: 24 Col: 30
```

2. Tugas 5



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

12
11
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
>>> indexKota(Klaten)

Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    indexKota(Klaten)
NameError: name 'indexKota' is not defined
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
>>>
Nomor 1
Mahasiswa yang tinggal di Klaten, berada pada index [6, 8]

Nomor 2
Uang saku terkecil adalah 650

Nomor 3
Mahasiswa dengan uang saku terkecil [('amin', 16, 'Boyolali', 650), ('naura', 17, 'Klaten', 650)]

Nomor 4
Mahasiswa dengan uang saku kurang dari 250.000:
('ari', 21, 'Sukoharjo', 1000)
('ali', 35, 'Sragen', 5000)
('rojim', 27, 'Surakarta', 9500)
('amin', 16, 'Boyolali', 650)
('naura', 17, 'Klaten', 650)
('atta', 17, 'Wonogiri', 2800)
('ginza', 18, 'Klaten', 43000)
('vanessa', 25, 'Purwodadi', 20000)
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
Data 10 ada dalam Linked List
Data 110 tidak ada dalam Linked List
>>>

L200180059_Algostruk_Modul4_Tugas.py - F:/aida sh/itu/apalah/lesson/semester 4/ASD/prak...
File Edit Format Run Options Window Help

##5. mencari data dari linked list

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

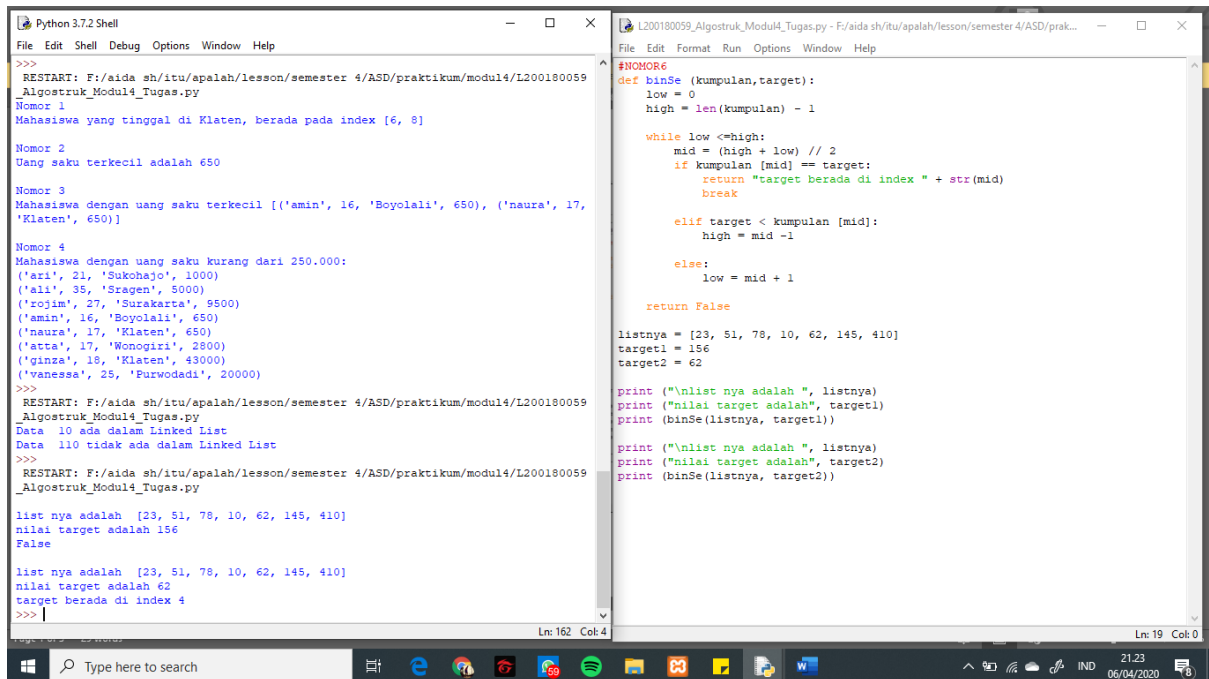
    def cariLinkedList(self, dicari):
        curNode = self
        while curNode is not None:
            if curNode.next != None:
                if curNode.data != dicari:
                    curNode = curNode.next
            else:
                print ("Data ", dicari, "ada dalam Linked List")
                break
        elif curNode.next == None:
            print ("Data ", dicari, "tidak ada dalam Linked List")
            break

a = node(12)
menu = a
a.next = node(34)
a = a.next
a.next = node(10)
a = a.next
a.next = node(45)

menu.cariLinkedList(10)
menu.cariLinkedList(110)

Ln: 152 Col: 4
Ln: 22 Col: 14
```

3. Tugas 6



The screenshot shows a Python 3.7.2 Shell window on the left and a code editor window on the right. The shell window displays the execution of a script named `_Algostruk_Modul4_Tugas.py`. The script defines a list of students with their names, IDs, and money amounts. It then performs a binary search for a target value of 156. The search fails, and the script prints the list and the target value. The code editor window shows the implementation of the `binSe` function, which takes a list and a target value as input and returns the index of the target value if it is found, or `-1` if it is not found.

```
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
Nomor 1
Mahasiswa yang tinggal di Klaten, berada pada index [6, 8]

Nomor 2
Uang saku terkecil adalah 650

Nomor 3
Mahasiswa dengan uang saku terkecil [('amin', 16, 'Boyolali', 650), ('naura', 17,
'Klaten', 650)]

Nomor 4
Mahasiswa dengan uang saku kurang dari 250.000:
('ari', 21, 'Sukoharjo', 1000)
('ali', 35, 'Sragen', 5000)
('rojim', 27, 'Surakarta', 9500)
('amin', 16, 'Boyolali', 650)
('naura', 17, 'Klaten', 650)
('atta', 17, 'Wonogiri', 2800)
('ginza', 18, 'Klaten', 43000)
('vanessa', 25, 'Purwodadi', 20000)
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
Data 10 ada dalam Linked List
Data 110 tidak ada dalam Linked List
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py

list nya adalah [23, 51, 78, 10, 62, 145, 410]
nilai target adalah 156
False

list nya adalah [23, 51, 78, 10, 62, 145, 410]
nilai target adalah 62
target berada di index 4
>>>
Ln: 162 Col: 4
```

```
#NOMOR 6
def binSe (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

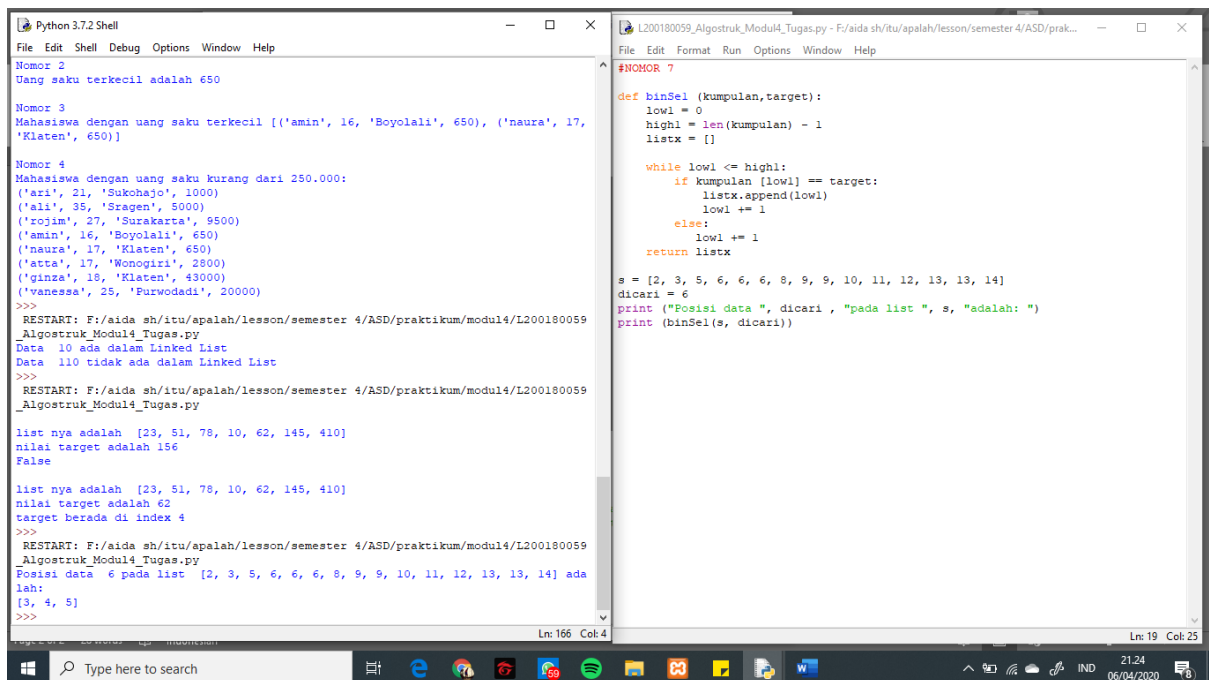
    return False

listnya = [23, 51, 78, 10, 62, 145, 410]
target1 = 156
target2 = 62

print ("\nlist nya adalah ", listnya)
print ("nilai target adalah", target1)
print (binSe(listnya, target1))

print ("\nlist nya adalah ", listnya)
print ("nilai target adalah", target2)
print (binSe(listnya, target2))
Ln: 19 Col: 0
```

4. Tugas 7



The screenshot shows a Python 3.7.2 Shell window on the left and a code editor window on the right. The shell window displays the execution of a script named `_Algostruk_Modul4_Tugas.py`. The script defines a list of students with their names, IDs, and money amounts. It then performs a binary search for a target value of 156. The search fails, and the script prints the list and the target value. The code editor window shows the implementation of the `binSel` function, which takes a list and a target value as input and returns the index of the target value if it is found, or `-1` if it is not found.

```
Nomor 2
Uang saku terkecil adalah 650

Nomor 3
Mahasiswa dengan uang saku terkecil [('amin', 16, 'Boyolali', 650), ('naura', 17,
'Klaten', 650)]

Nomor 4
Mahasiswa dengan uang saku kurang dari 250.000:
('ari', 21, 'Sukoharjo', 1000)
('ali', 35, 'Sragen', 5000)
('rojim', 27, 'Surakarta', 9500)
('amin', 16, 'Boyolali', 650)
('naura', 17, 'Klaten', 650)
('atta', 17, 'Wonogiri', 2800)
('ginza', 18, 'Klaten', 43000)
('vanessa', 25, 'Purwodadi', 20000)
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
Data 10 ada dalam Linked List
Data 110 tidak ada dalam Linked List
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py

list nya adalah [23, 51, 78, 10, 62, 145, 410]
nilai target adalah 156
False

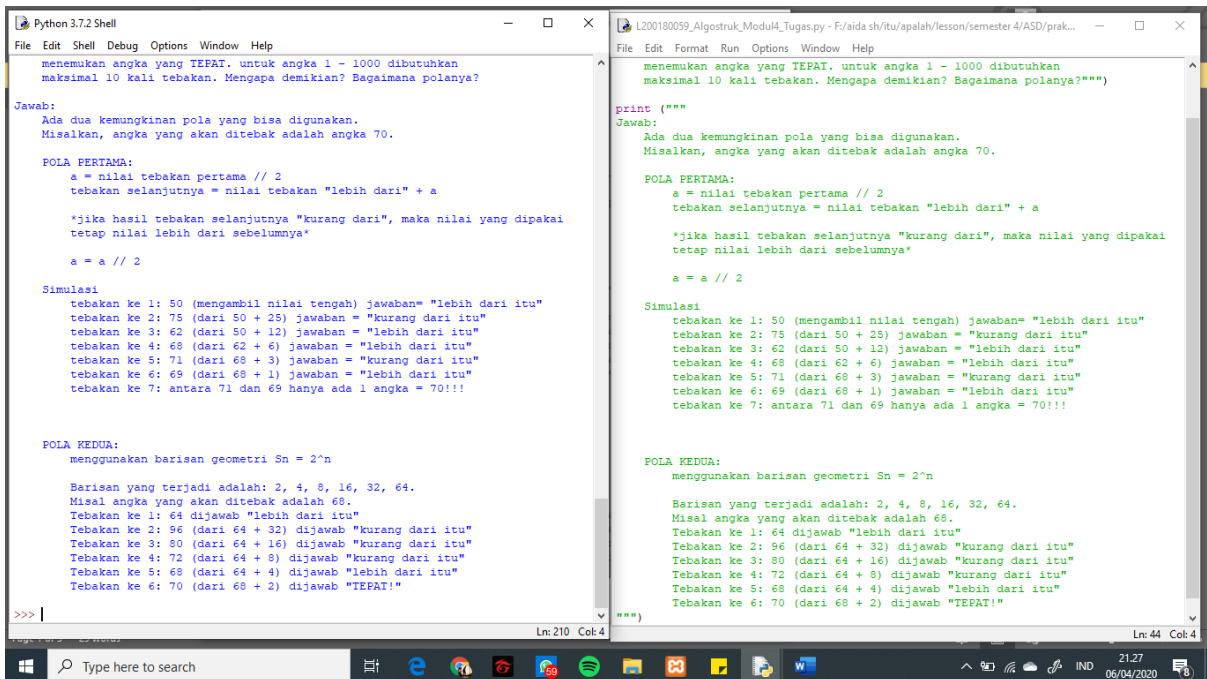
list nya adalah [23, 51, 78, 10, 62, 145, 410]
nilai target adalah 62
target berada di index 4
>>>
RESTART: F:/aida sh/itu/apalah/lesson/semester 4/ASD/praktikum/modul4/L200180059
_Algostruk_Modul4_Tugas.py
Posisi data 6 pada list [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14] ada
lah:
[3, 4, 5]
>>>
Ln: 166 Col: 4
```

```
#NOMOR 7
def binSel (kumpulan,target):
    low1 = 0
    high1 = len(kumpulan) - 1
    listx = []

    while low1 <= high1:
        if kumpulan [low1] == target:
            listx.append(low1)
            low1 += 1
        else:
            low1 += 1
    return listx

s = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
dicari = 6
print ("Posisi data ", dicari , "pada list ", s, "adalah: ")
print (binSel(s, dicari))
Ln: 19 Col: 25
```

5. Tugas 8



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

menemukan angka yang TEPAT. untuk angka 1 - 1000 dibutuhkan
maksimal 10 kali tebakan. Mengapa demikian? Bagaimana polanya?

Jawab:
Ada dua kemungkinan pola yang bisa digunakan.
Misalkan, angka yang akan ditebak adalah angka 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  $S_n = 2^n n$ 
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!"

Lrn: 210 Col: 4

L200180059_Algostruk_Modul4_Tugas.py - F:\aida sh\itu/apalah\lesson/semester 4/ASD/prak...
File Edit Format Run Options Window Help

menemukan angka yang TEPAT. untuk angka 1 - 1000 dibutuhkan
maksimal 10 kali tebakan. Mengapa demikian? Bagaimana polanya?"""

print("""
Jawab:
Ada dua kemungkinan pola yang bisa digunakan.
Misalkan, angka yang akan ditebak adalah angka 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  $S_n = 2^n n$ 
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!"

""")

Lrn: 44 Col: 4
```