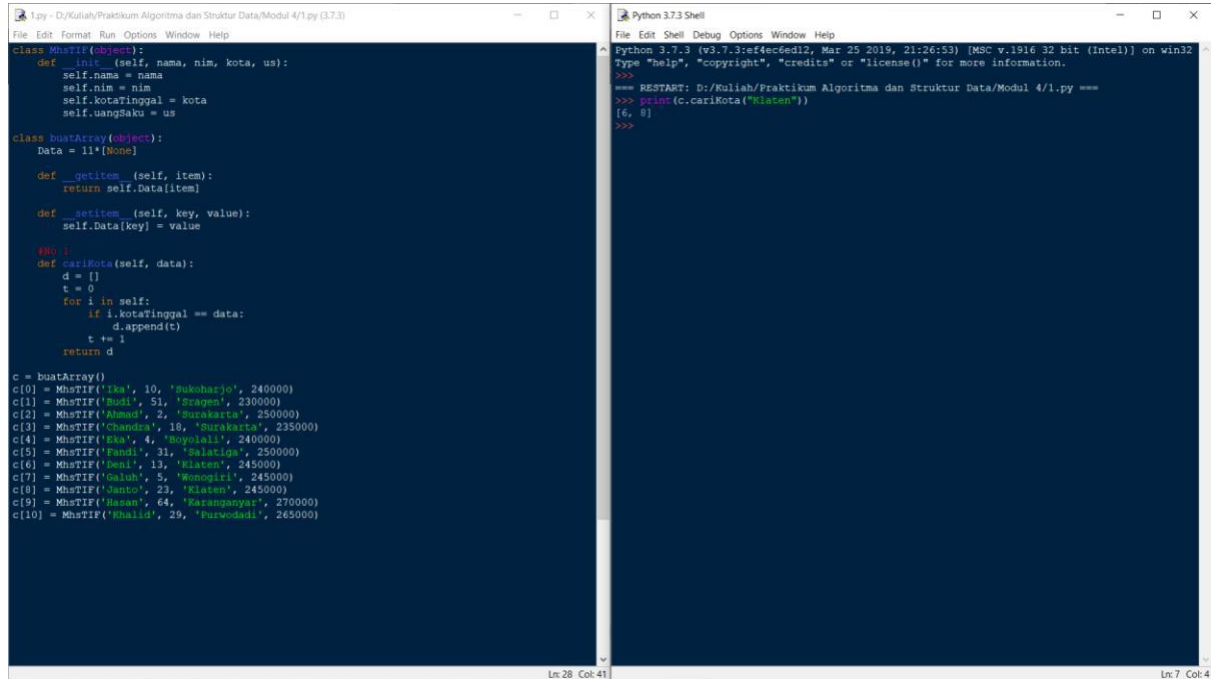


LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
MODUL 4
Fawwaz Haidar A.K
L200183143
KELAS H

Nomer 1



```
File Edit Format Run Options Window Help
class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = us

class buatArray(object):
    Data = []
    def __getitem__(self, item):
        return self.Data[item]

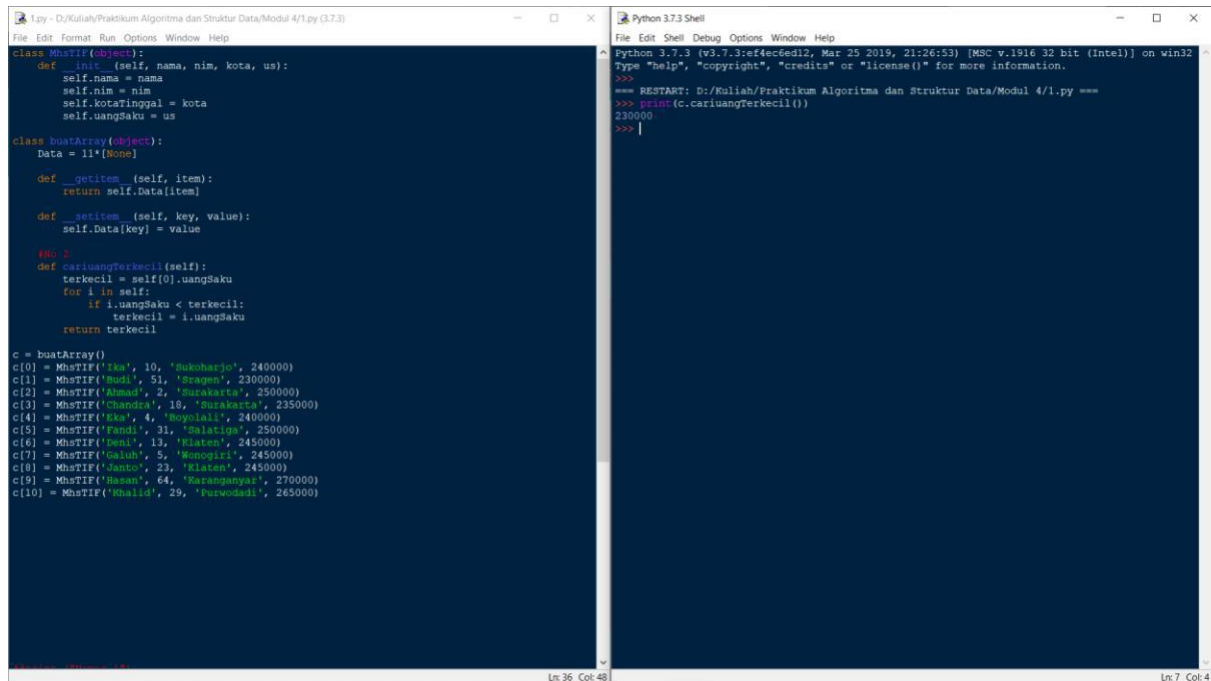
    def __setitem__(self, key, value):
        self.Data[key] = value

PBO 1
def cariKota(self, data):
    d = []
    t = 0
    for i in self:
        if i.kotaTinggal == data:
            d.append(t)
            t += 1
    return d

c = buatArray()
c[0] = MhsTIF('Ika', 10, 'Bukharjo', 240000)
c[1] = MhsTIF('Budi', 51, 'Sragen', 230000)
c[2] = MhsTIF('Ahmad', 2, 'Surakarta', 250000)
c[3] = MhsTIF('Chandra', 18, 'Surakarta', 235000)
c[4] = MhsTIF('Eka', 4, 'Boyolali', 240000)
c[5] = MhsTIF('Fandi', 31, 'Salatiga', 250000)
c[6] = MhsTIF('Demi', 13, 'Klaten', 245000)
c[7] = MhsTIF('Galuh', 5, 'Wonogiri', 245000)
c[8] = MhsTIF('Janto', 23, 'Klaten', 245000)
c[9] = MhsTIF('Hasan', 64, 'Karanganyar', 270000)
c[10] = MhsTIF('Khalid', 29, 'Purwodadi', 265000)

Python 3.7.3 Shell
Python 3.7.3 (v3.7.3:ef4ecded12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 4/1.py ===
>>> print(c.cariKota("Klaten"))
[6, 8]
>>>
```

Nomer 2



```
File Edit Format Run Options Window Help
class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = us

class buatArray(object):
    Data = []
    def __getitem__(self, item):
        return self.Data[item]

    def __setitem__(self, key, value):
        self.Data[key] = value

PBO 2
def cariuangTerkecil(self):
    terkecil = self[0].uangSaku
    for i in self:
        if i.uangSaku < terkecil:
            terkecil = i.uangSaku
    return terkecil

c = buatArray()
c[0] = MhsTIF('Ika', 10, 'Bukharjo', 240000)
c[1] = MhsTIF('Budi', 51, 'Sragen', 230000)
c[2] = MhsTIF('Ahmad', 2, 'Surakarta', 250000)
c[3] = MhsTIF('Chandra', 18, 'Surakarta', 235000)
c[4] = MhsTIF('Eka', 4, 'Boyolali', 240000)
c[5] = MhsTIF('Fandi', 31, 'Salatiga', 250000)
c[6] = MhsTIF('Demi', 13, 'Klaten', 245000)
c[7] = MhsTIF('Galuh', 5, 'Wonogiri', 245000)
c[8] = MhsTIF('Janto', 23, 'Klaten', 245000)
c[9] = MhsTIF('Hasan', 64, 'Karanganyar', 270000)
c[10] = MhsTIF('Khalid', 29, 'Purwodadi', 265000)

Python 3.7.3 Shell
Python 3.7.3 (v3.7.3:ef4ecded12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 4/1.py ===
>>> print(c.cariuangTerkecil())
230000
>>>
```

Nomer 3

```
1.py - D:\Kuliah\Praktikum Algoritma dan Struktur Data\Modul 4\1.py (3.7.3)
File Edit Format Run Options Window Help

class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = us

class buatArray(object):
    Data = []

    def __getitem__(self, item):
        return self.Data[item]

    def __setitem__(self, key, value):
        self.Data[key] = value

#03
def cariKecil(self):
    terkecil = self[0].uangSaku
    for i in self:
        if i.uangSaku < terkecil:
            terkecil = i.uangSaku
    d = []
    for i in self:
        if i.uangSaku == terkecil:
            d.append((i.nama, i.nim, i.kotaTinggal, i.uangSaku))
    return d

c = buatArray()
c[0] = MhsTIF('Ika', 10, 'Sukoharjo', 240000)
c[1] = MhsTIF('Budi', 51, 'Sragen', 230000)
c[2] = MhsTIF('Ahmad', 2, 'Surakarta', 250000)
c[3] = MhsTIF('Chandra', 18, 'Surakarta', 235000)
c[4] = MhsTIF('Eka', 4, 'Boyolali', 240000)
c[5] = MhsTIF('Fandi', 31, 'Salatiga', 250000)
c[6] = MhsTIF('Deni', 13, 'Klaten', 245000)
c[7] = MhsTIF('Galuh', 5, 'Wonogiri', 245000)
c[8] = MhsTIF('Janto', 23, 'Klaten', 245000)
c[9] = MhsTIF('Hasan', 64, 'Karanganyar', 270000)
c[10] = MhsTIF('Khaidi', 29, 'Purwodadi', 265000)

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

Python 3.7.3 (v3.7.3:ef4ecded12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: D:\Kuliah\Praktikum Algoritma dan Struktur Data\Modul 4\1.py ====
>>> print(c.cariKecil())
[('Budi', 51, 'Sragen', 230000)]
>>>
```

Nomer 4

```
1.py - D:\Kuliah\Praktikum Algoritma dan Struktur Data\Modul 4\1.py (3.7.3)
File Edit Format Run Options Window Help

class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = us

class buatArray(object):
    Data = []

    def __getitem__(self, item):
        return self.Data[item]

    def __setitem__(self, key, value):
        self.Data[key] = value

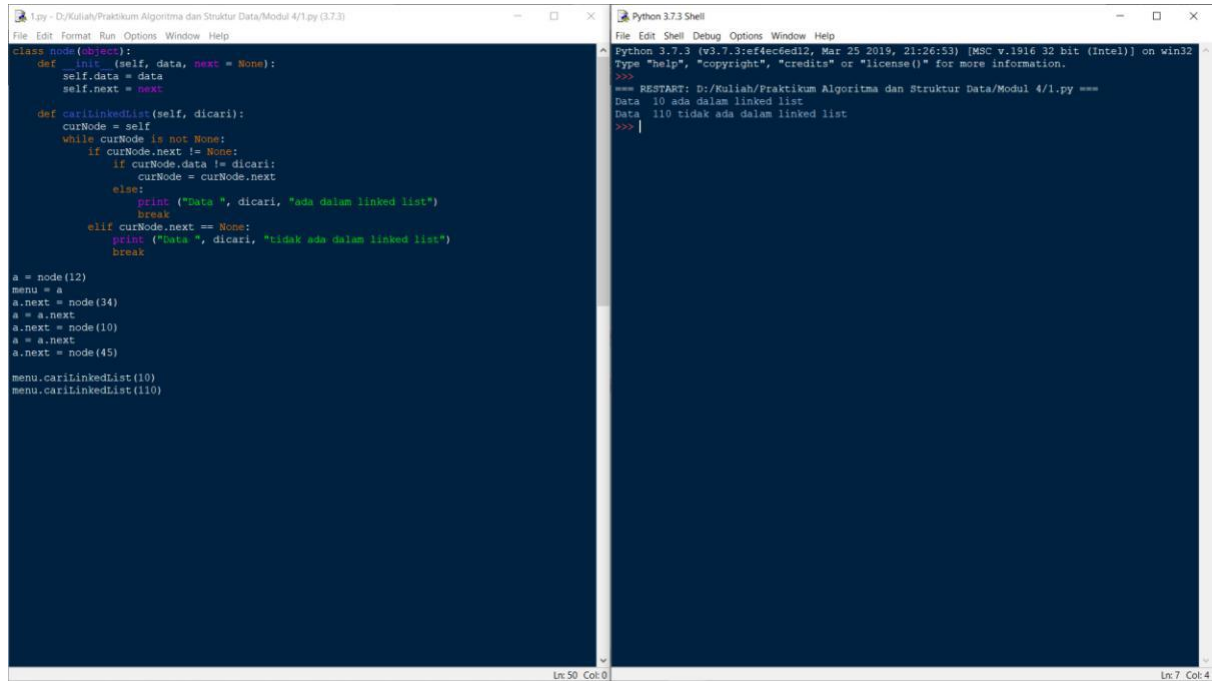
#04
def below250k(self):
    terkecil = 250000
    d = []
    for i in self:
        if i.uangSaku < terkecil:
            d.append((i.nama, i.nim, i.kotaTinggal, i.uangSaku))
    for i in d:
        print(i)

c = buatArray()
c[0] = MhsTIF('Ika', 10, 'Sukoharjo', 240000)
c[1] = MhsTIF('Budi', 51, 'Sragen', 230000)
c[2] = MhsTIF('Ahmad', 2, 'Surakarta', 250000)
c[3] = MhsTIF('Chandra', 18, 'Surakarta', 235000)
c[4] = MhsTIF('Eka', 4, 'Boyolali', 240000)
c[5] = MhsTIF('Fandi', 31, 'Salatiga', 250000)
c[6] = MhsTIF('Deni', 13, 'Klaten', 245000)
c[7] = MhsTIF('Galuh', 5, 'Wonogiri', 245000)
c[8] = MhsTIF('Janto', 23, 'Klaten', 245000)
c[9] = MhsTIF('Hasan', 64, 'Karanganyar', 270000)
c[10] = MhsTIF('Khaidi', 29, 'Purwodadi', 265000)

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

Python 3.7.3 (v3.7.3:ef4ecded12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: D:\Kuliah\Praktikum Algoritma dan Struktur Data\Modul 4\1.py ====
>>> c.below250k()
('Ika', 10, 'Sukoharjo', 240000)
('Budi', 51, 'Sragen', 230000)
('Chandra', 18, 'Surakarta', 235000)
('Eka', 4, 'Boyolali', 240000)
('Deni', 13, 'Klaten', 245000)
('Galuh', 5, 'Wonogiri', 245000)
('Janto', 23, 'Klaten', 245000)
>>>
```

Nomer 5



The screenshot shows a Python IDE with two windows. The left window displays a Python script for a linked list. The right window shows the output of the script executed in a Python 3.7.3 Shell.

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

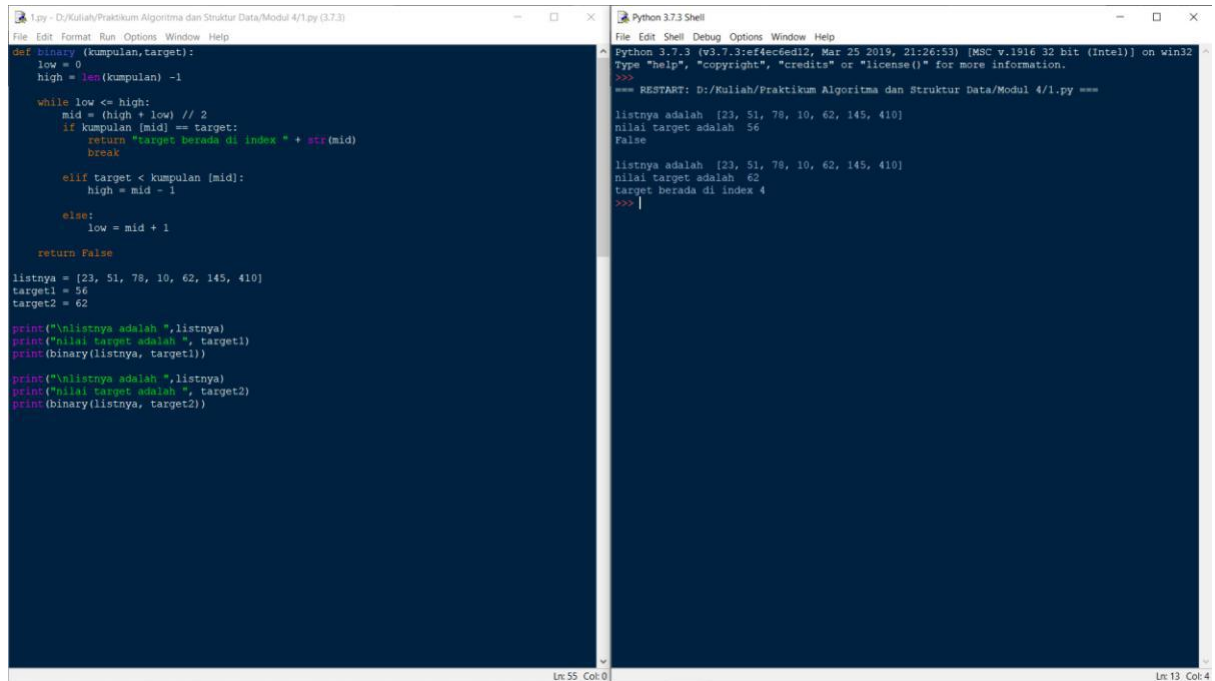
    def cariInkedList(self, dicari):
        curNode = self
        while curNode is not None:
            if curNode.next != None:
                if curNode.data != dicari:
                    curNode = curNode.next
                else:
                    print("Data ", dicari, "ada dalam linked list")
                    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.cariInkedList(10)
menu.cariInkedList(110)
```

```
Python 3.7.3 (v3.7.3:ef4ecded12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 4/1.py ===
Data 10 ada dalam linked list
Data 110 tidak ada dalam linked list
>>>
```

Nomer 6



The screenshot shows a Python IDE with two windows. The left window displays a Python script for a binary search algorithm. The right window shows the output of the script executed in a Python 3.7.3 Shell.

```
def binary(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 = 56
target2 = 62

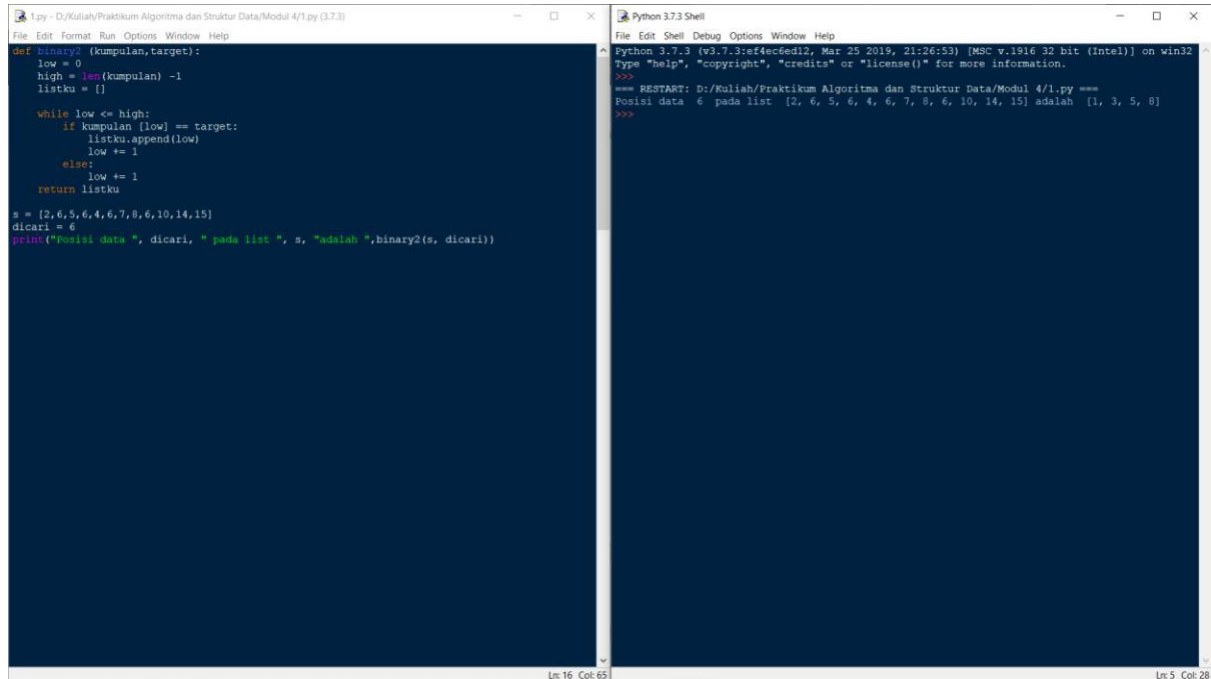
print("listnya adalah ", listnya)
print("nilai target adalah ", target1)
print(binary(listnya, target1))

print("listnya adalah ", listnya)
print("nilai target adalah ", target2)
print(binary(listnya, target2))
```

```
Python 3.7.3 (v3.7.3:ef4ecded12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 4/1.py ===
listnya adalah [23, 51, 78, 10, 62, 145, 410]
nilai target adalah 56
False

listnya adalah [23, 51, 78, 10, 62, 145, 410]
nilai target adalah 62
target berada di index 4
>>>
```

Nomer 7



```
1.py - D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 4/1.py (3.7.3)
File Edit Format Run Options Window Help
def binary2(kumpulan, target):
    low = 0
    high = len(kumpulan) - 1
    listku = []
    while low <= high:
        if kumpulan[low] == target:
            listku.append(low)
            low += 1
        else:
            low += 1
    return listku
s = [2, 6, 5, 6, 4, 6, 7, 8, 6, 10, 14, 15]
dicari = 6
print("Posisi data ", dicari, " pada list ", s, " adalah ", binary2(s, dicari))

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ecded12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: D:/Kuliah/Praktikum Algoritma dan Struktur Data/Modul 4/1.py ===
Posisi data 6 pada list [2, 6, 5, 6, 4, 6, 7, 8, 6, 10, 14, 15] adalah [1, 3, 5, 8]
>>>
```

Nomer 8

Soal :

Pada permainan tebak angka, 1-100 dibutuhkan maksimal 7 kali tebakan untuk 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 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 $S_n = 2^n$

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"