

Nama : Amartya Bintang Wijat Ranti
NIM : L200180193
Kelas : G
Mata Kuliah : Praktikum Algoritma dan Struktur Data

Modul 4

```
No. 1-4.py - E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul4\No....
File Edit Format Run Options Window Help

class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', nim ' + str(self.nim)\
            + '. Tinggal di ' + self.kota\
            + '. Uang saku Rp ' + str(self.uangSalu)\
            + '. tiap bulannya.'
        return s

c0 = MhsTIF("Bintang", 10, "Sukoharjo", 240000)
c1 = MhsTIF("Baity", 51, "Sragen", 230000)
c2 = MhsTIF("Ceyl", 2, "Surakarta", 250000)
c3 = MhsTIF("Alfian", 18, "Surakarta", 235000)
c4 = MhsTIF("Riska", 4, "Boyolali", 240000)
c5 = MhsTIF("Inung", 31, "Salatiga", 250000)
c6 = MhsTIF("Amar", 13, "Klaten", 245000)
c7 = MhsTIF("Ulin", 5, "Wonogiri", 245000)
c8 = MhsTIF("Aviza", 23, "Klaten", 245000)
c9 = MhsTIF("Danang", 64, "Karanganyar", 270000)
c10 = MhsTIF("Tika", 29, "Purwodadi", 265000)

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

#No. 1
def cariAsal():
    target = input("Masukkan kota: ")
    x = []
    for i in range (len(Daftar)):
        if target == Daftar[i].kota:
            x.append(i)
    print (x)

#No. 2
def cariUSTerkecil():
    terkecil = Daftar[0].uangSaku
```

```

    for i in range (len(Daftar)):
        if terkecil > Daftar[i].uangSaku:
            terkecil = Daftar[i].uangSaku
    return terkecil

#No. 3
def cariUSTerkecil2():
    terkecil = Daftar[0].uangSaku
    x = []
    a = cariUSTerkecil
    for i in range (len(Daftar)):
        if terkecil > Daftar[i].uangSaku:
            terkecil = Daftar[i].uangSaku

    for i in range (len(Daftar)):
        if Daftar[i].uangSaku == terkecil:
            x.append(Daftar[i].nama)
    return x

#No. 4
def carikurang25():
    a = 250000
    x = []
    for i in range(len(Daftar)):
        if Daftar[i].uangSaku < a:
            x.append(Daftar[i].nama)
    return x

```

No. 5.py - E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul4\No. 5... □ ×

File Edit Format Run Options Window Help

```

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(45)
menu = a
a.next = node (9)
a = a.next
a.next = node (17)
a = a.next
a.next = node (23)

menu.cariLinkedList(9)
menu.cariLinkedList(22)

```

No. 6 -7.py - E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul4\No...

File Edit Format Run Options Window Help

```
A = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22]
```

#No. 6

```
def binSe(target):
    low = 0
    high = len(A)

    while low < high:
        mid = (high + low) // 2
        if A[mid] == target:
            return "Target pada indeks " + str(mid)
        elif target < A[mid]:
            high = mid - 1
        else:
            low = mid + 1
    return False
```

#No. 7

```
B = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
```

```
def binSe2(target):
    low = 0
    high = len(B)
    x = []

    while low < high:
        if B[low] == target:
            x.append(low)
            low+=1
        else:
            low+=1
    return x
```

*No. 8.py - E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul4\No. ...

File Edit Format Run Options Window Help

```
print("""Karena 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!!!
```

```
""")
```

Hasil Run

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul4\No. 1-4.py
>>> cariAsal()
Masukkan kota: Klaten
[6, 8]
>>> cariUSTerkecil()
230000
>>> cariUSTerkecil2()
['Baity']
>>> carikurang25()
['Bintang', 'Baity', 'Alfian', 'Riska', 'Amar', 'Ulin', 'Aviza']
>>>
RESTART: E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul4\No. 5.py
Data 9 ada dalam Linked List
Data 22 tidak ada dalam Linked List
>>>
RESTART: E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul4\No. 6 -7.py
>>> binSe(6)
'Target pada indeks 2'
>>> binSe2(20)
[]
>>> binSe2(10)
[9]
>>>
RESTART: E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul4\No. 8.py
Karena menggunakan konsep Big-O. Dimana yang dipakai
adalah rumus  $O(\log n)$  dengan rincian  $1 = 1$ ,  $2 = 2$ ,  $4 = 3$ ,  $10 = 4$ ,  $100 = 7$ ,  $1000 = 10$ 
.
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!!!
```