Nama : Salmaa Khoirun Nisaa’

NIM : L200180019

Kelas : A

Modul 4

Pencarian

1. Nomer 1

```
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]

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

```
 'target'
>>> cariKotaTinggal(Daftar, target)
[6, 8]
>>> target = 'Klaten'
>>> cariKotaTinggal(Daftar, target)
[6, 8]
>>> target = 'Sukoharjo'
>>> cariKotaTinggal(Daftar, target)
[0]
>>> target = 'Jepara'
>>> cariKotaTinggal(Daftar, target)
[]
>>>
```

2. Nomer 2

```
def carisakuterkecil(list):
    terkecil = list[0].uangsaku
    for i in list[1:]:
        if i.uangsaku < terkecil :
            terkecil = i.uangsaku
    return terkecil

a = carisakuterkecil(Daftar)
print(a)
```

```
----- RESTART:
230000
>>>
```

3. Nomer 3

```
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku


class buatArray(object):
    internalData = 11 * [None]

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

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

    def siapaTerkecil(self):
        terkecil = self[0].uangSaku
        d = []
        for i in self:
            if i.uangSaku <= terkecil:
                terkecil = i.uangSaku
        for i in self:
            if terkecil == i.uangSaku:
                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("Khalid", 29, "Purwodadi", 265000)
```

```
>>> c.siapaTerkecil()
[('Budi', 51, 'Sragen', 230000)]
>>>
```

4. Nomer 4

```
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]

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)
```

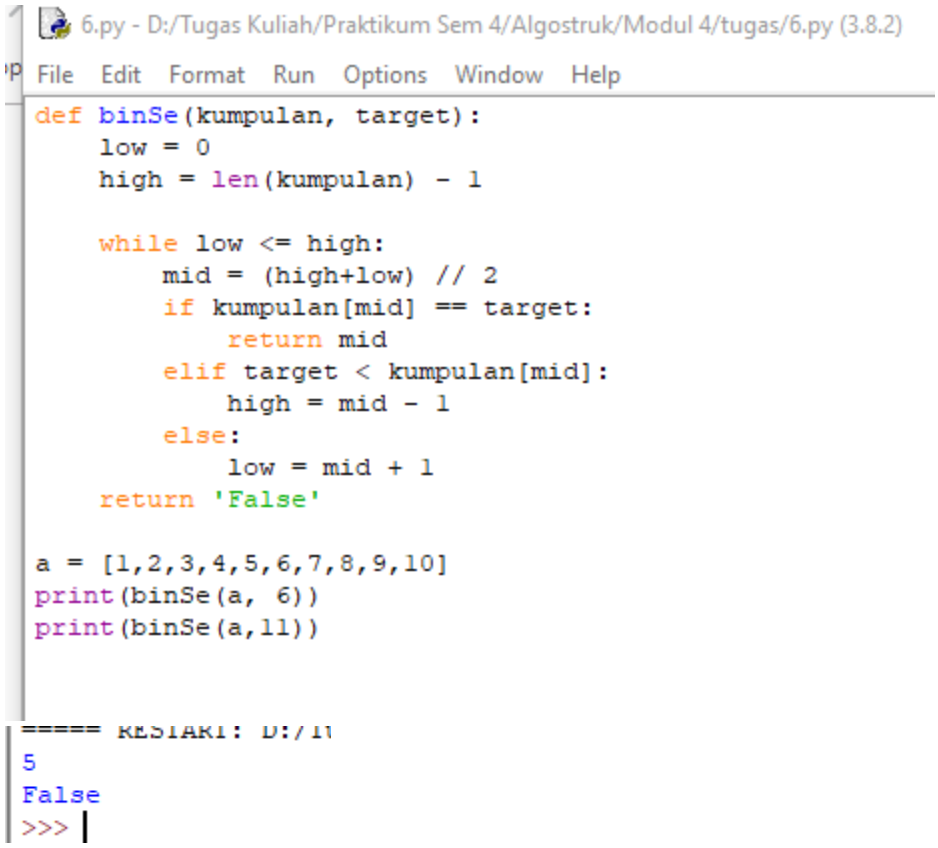
```
===== RESTART: D:/Tugas Kuliah/Praktikum S
Ika
Budi
Chandra
Eka
Deni
Galuh
Janto
>>>
```

5. Nomer 5

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

class LinkedList:
    def __init__(self, head = None):
        self.head = head

    def cari(self, head, yang_dicari):
        while head is not None:
            if head.data == yang_dicari:
                return True
            head = head.next
        return False

a = Node(11)
b = Node(12)
c = Node(13)
d = Node(15)
e = Node(16)

a.next = b
b.next = c
c.next = d
d.next = e

llist = LinkedList(a)

print(llist.cari(a,13))
print(llist.cari(a,12))
```

```
----- RESTART
True
True
>>>
```

6. Nomer 6


6.py - D:/Tugas Kuliah/Praktikum Sem 4/Algostruk/Modul 4/tugas/6.py (3.8.2)
File Edit Format Run Options Window Help
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'
a = [1,2,3,4,5,6,7,8,9,10]
print(binSe(a, 6))
print(binSe(a,11))
5
False
>>>

7. Nomer 7

```
def binSe(kumpulan, target):
    low = 0
    high = len(kumpulan) - 1

    index = []

    while low <= high:
        mid = (high+low) // 2
        if kumpulan[mid] == target:
            while kumpulan[mid - 1] == target:
                mid -= 1
            while kumpulan[mid] == target:
                index.append(mid)
                mid += 1
                if mid > len(kumpulan)-1:
                    break
            return index
        elif target < kumpulan[mid]:
            high = mid - 1
        else:
            low = mid + 1
    return 'False'

a = [1,1,1,2,3,4]
b = [1,2,4,4,4,4]
c = [2,3,5,6,6,6,8,9,9,10]
print(binSe(a,1))
print(binSe(b,4))
print(binSe(c, 6))
```

```
===== RESTART: D:/Tu
[0, 1, 2]
[2, 3, 4, 5]
[3, 4, 5]
>>>
```

8. Nomer 8

```python
from random import randint

def log2n(n):
    return 1 + log2n(n/2) if (n > 1) else 0

def quiz(angka):
    quiz = randint(1, angka)
    jawab = 0
    count = 1
    maks = log2n(angka)

    print('Saya menyimpan angka bulat antara 1 sampai {}. anda punya {}x kesempa
    while jawab != quiz and count <= maks:
        jawab = int(input('Masukkan tebakan ke-{}:>'.format(count)))
        if jawab == quiz:
            print('Ya. Anda benar')
        elif jawab < quiz:
            print('Itu terlalu kecil. Coba lagi')
        else:
            print('Itu terlalu besar. Coba lagi')
        count += 1

quiz(10000)
```

```
===== RESTART: D:/Tugas Kuliah/Praktikum Sem 4/Algostruk/Modul 4/tugas/8.py ====
Saya menyimpan angka bulat antara 1 sampai 10000. anda punya 14x kesempatan. coba tebak
Masukkan tebakan ke-1:>100
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-2:>600
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-3:>700
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-4:>800
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-5:>900
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-6:>999
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-7:>8000
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-8:>6000
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-9:>5000
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-10:>4000
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-11:>1000
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-12:>2000
Itu terlalu besar. Coba lagi
Masukkan tebakan ke-13:>1111
Itu terlalu kecil. Coba lagi
Masukkan tebakan ke-14:>1211
Itu terlalu kecil. Coba lagi
>>>
```