

Nama : Ismi Dzikrina

NIM : L200180010

Kelas : A

Matkul : Praktikum Algoritma Dan Struktur Data

MODUL 4

PENCARIAN

SOAL-SOAL UNTUK MAHASISWA

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

===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/1.py =====
>>> a = cariKotaTinggal(Daftar, "Klaten")
>>> print(a)
[6, 8]
>>> |
```

2.

```
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)

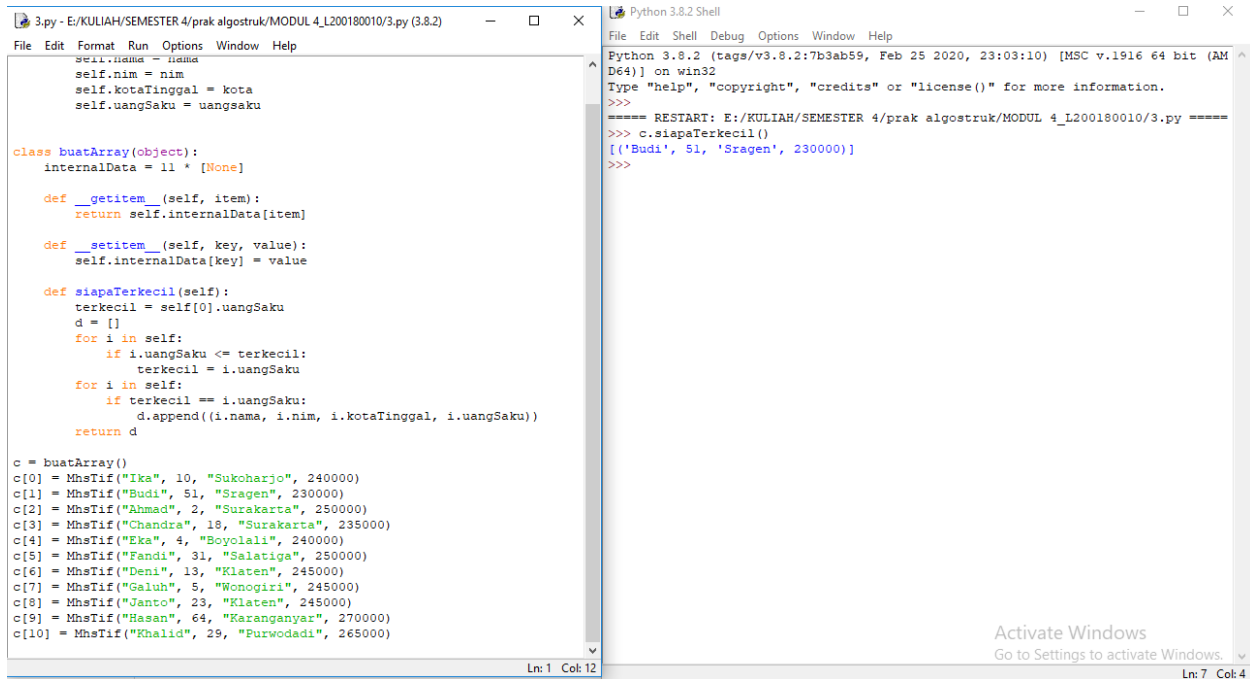
Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]

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

a = carisakuterkecil(Daftar)
print(a)

===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/2.py =====
230000
>>> |
```

3.



```
3.py - E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/3.py (3.8.2)
File Edit Format Run Options Window Help

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

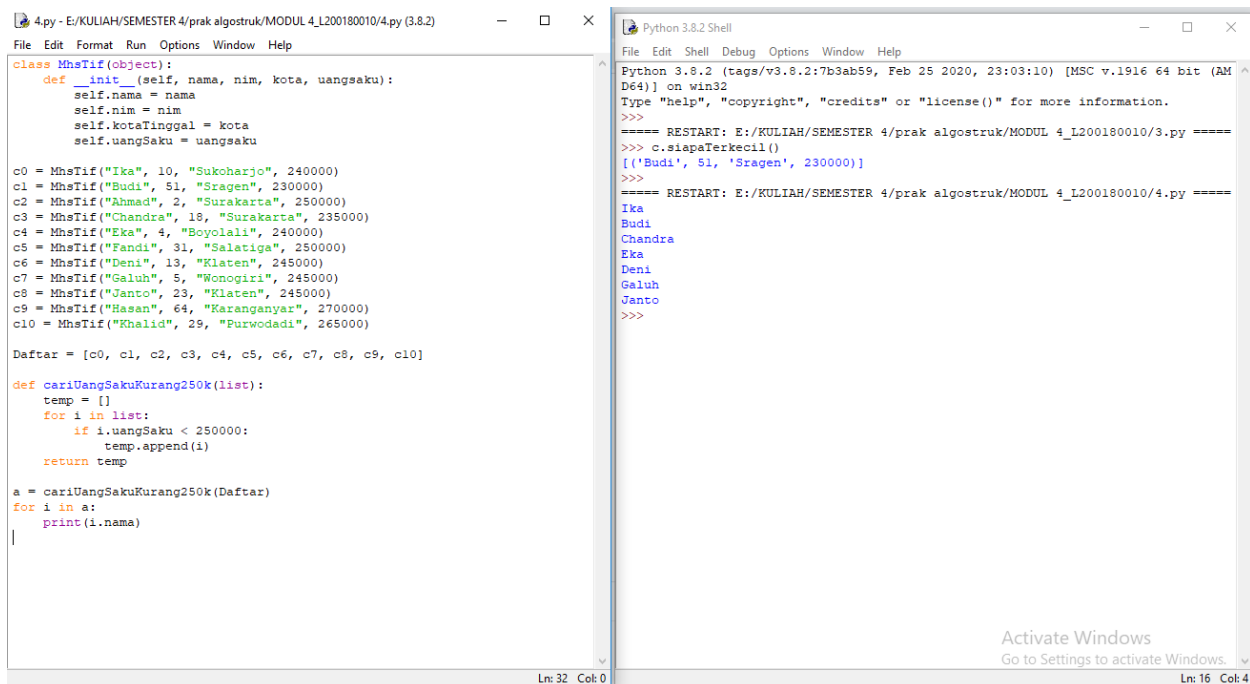
    def siapaTerkecil(self):
        terkecil = self[0].uangSaku
        d = []
        for i in self:
            if i.uangSaku <= terkecil:
                terkecil = i.uangSaku
        for i in self:
            if terkecil == i.uangSaku:
                d.append((i.nama, i.nim, i.kotaTinggal, i.uangSaku))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

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/3.py =====
>>> c.siapaTerkecil()
[('Budi', 51, 'Sragen', 230000)]
>>>
```

4. No 4



```
4.py - E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/4.py (3.8.2)
File Edit Format Run Options Window Help

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)
|

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/3.py =====
>>> c.siapaTerkecil()
[('Budi', 51, 'Sragen', 230000)]
>>>
===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/4.py =====
Ika
Budi
Chandra
Eka
Deni
Galuh
Janto
>>>
```

5.

```
5.py - E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/5.py (3...
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 is not None:
                if curNode.data != dicari:
                    curNode = curNode.next
            else:
                print ("Data ", dicari, "ada dalam linked list")
                break
        elif curNode.next is None:
            print ("Data ", dicari, "tidak ada dalam linked list")
            break

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/5.py =====
>>>
>>> a = node(12)
>>> menu = a
>>> a.next = node(11)
>>> a = a.next
>>> a.next = node(26)
>>> a = a.next
>>> a.next = node(43)
>>> menu.cariLinkedList(11)
Data 11 ada dalam linked list
>>> menu.cariLinkedList(9)
Data 9 tidak ada dalam linked list
>>>
```

6.

```
6.py - E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/6.py
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 "target berada di index " + str(mid)
            break

        elif target < kumpulan[mid]:
            high = mid - 1

        else:
            low = mid + 1
    return False

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/6.py =====
>>> list = [11, 34, 46, 57, 78, 100]
>>> binSe(list, 11)
'target berada di index 0'
>>> binSe(list, 55)
False
>>>
```

7.

```
7.py - E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/7.py
File Edit Format Run Options Window Help

def binSe(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

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/7.py =====
>>> a = [1,2,2,3,4,4,4, 5, 5,7, 8,8,9, 10]
>>> binSe(a,4)
[4, 5, 6]
>>> binSe(a, 8)
[10, 11]
>>>
```

8.

```
8.py - E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/8.py (3.8.2)
Python 3.8.2 Shell

File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help

print("""
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""")

print("""
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"
""")

===== RESTART: E:/KULIAH/SEMESTER 4/prak algostruk/MODUL 4_L200180010/8.py ==
=====
Soal :
    Pada permainan tebak angka, 1-100 dibutuhkan maksimal 7 kali tebakan untu
k
    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 dipaka
i
        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"
""")

>>>
Ln: 8 Col: 0
Ln: 1 Col: 94
```