

Nama : Alfianto Andy P

NIM : L200180194

MODUL 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]
#NO 1
print("NO 1 =")
def cariKotaTinggal(list, target):
    a = []
    for i in list :
        if i.kotaTinggal == target:
            a.append(list.index(i))
    return a

a = cariKotaTinggal(Daftar, "Klaten")
print(a)
#NO 2
print("NO 2 =")
def cariUangSakuTerkecil(list):
    temp = list[0].uangSaku
    for i in list[1:]:
        if i.uangSaku < temp:
            temp = i.uangSaku
    return temp

a = cariUangSakuTerkecil(Daftar)
print(a)
#NO 3
print("NO 3 =")
```

```
#NO 3
print("NO 3 =")
def sakuKcl2(n):
    baru = n[0].uangSaku
    list = []
    for i in range(len(n)):
        if(n[i].uangSaku==baru):
            list.append(n[i].nama)
        elif(n[i].uangSaku<baru):
            baru = n[i].uangSaku
            list = []
            list.append(n[i].nama)
    return list
print(sakuKcl2(Daftar))
#NO 4
print("NO 4 =")
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)

#NO 5
print("NO 5 =")
def cariLinkedList(head, target):
    temp = head
    while temp.data != None:
        if temp.data == target:
            return temp
    return -1
#NO 6
print("NO 6 =")
def binSe(kumpulan, target):
    low = 0
    high = len(kumpulan)-1
    while low <= high:
        mid = (high+low)//2
        if kumpulan[mid] == target:
            return mid
```

```

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

kumpulan = [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
print(binSe(kumpulan, 5))
#NO 7
print("NO 7 ==")
def binSeMass(kumpulan, target):
    temp = []
    low = 0
    high = len(kumpulan)-1
    while low <= high :
        mid = (high+low)//2
        if kumpulan[mid] == target:
            midKiri = mid-1
            while kumpulan[midKiri] == target:
                temp.append(midKiri)
                midKiri = midKiri-1
            temp.append(mid)
            midKanan = mid+1
            while kumpulan[midKanan] == target:
                temp.append(midKanan)
                midKanan = midKanan+1
            return temp
        elif target < kumpulan[mid]:
            high = mid-1
        else:
            low = mid+1
    return False

kumpulan = [2, 4, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
print(binSeMass(kumpulan, 6))

```

```

print("NO 8 =")
#no 8
"""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!!!
"""

```

```

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: D:\Materi smester 4\Algo struktur data\Algo struktur data praktikum\Prak ASD modul 3 & 4\Modul 4\tugas.py
NO 1 =
[6, 8]
NO 2 =
230000
NO 3 =
['Budi']
NO 4 =
Ika
Budi
Chandra
Eka
Deni
Galuh
Janto
NO 5 =
NO 6 =
2
NO 7 =
[3, 4, 5]
NO 8 =
>>> |

```