

Nama : Aulia Yogatama

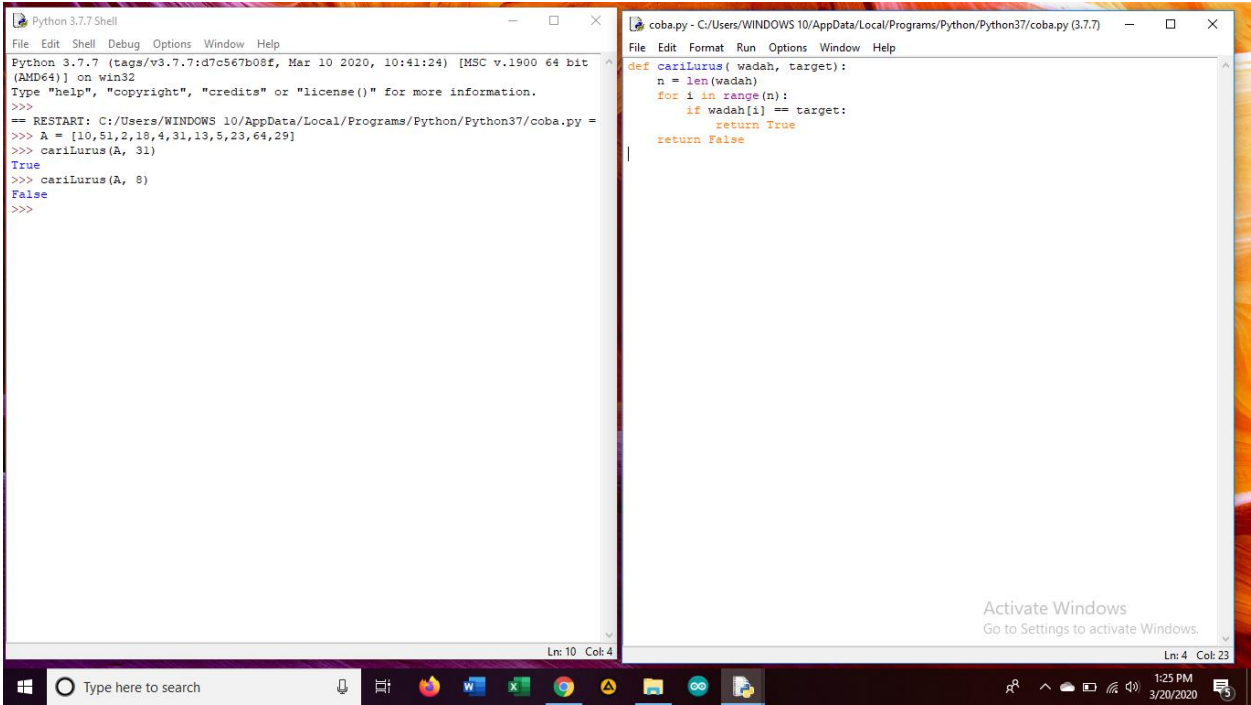
NIM : L200180081

Kelas : C

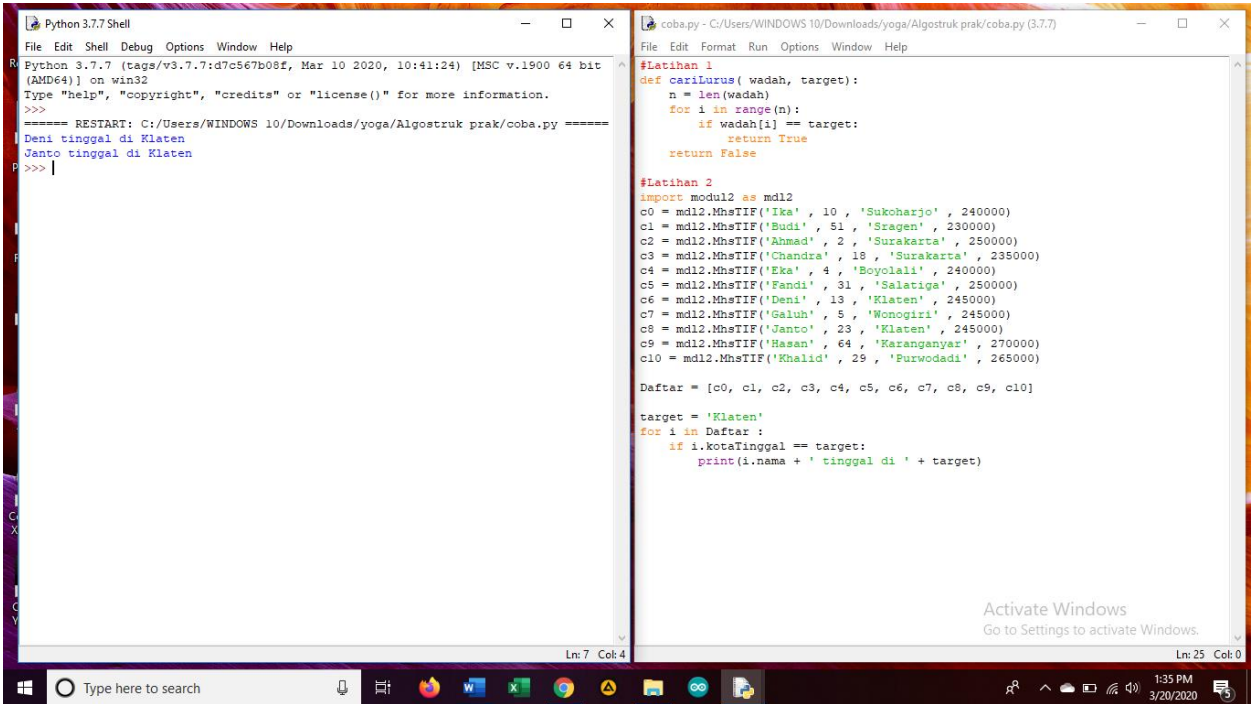
Modul 4

Pencarian

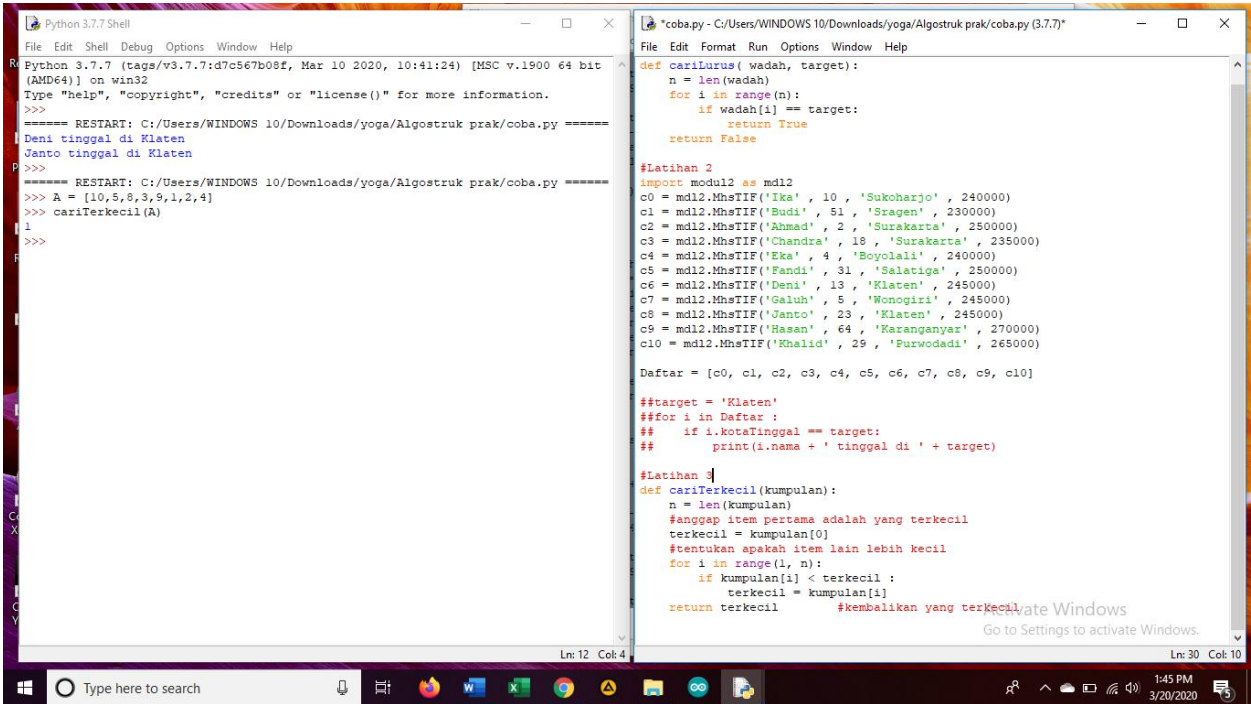
Latihan 1



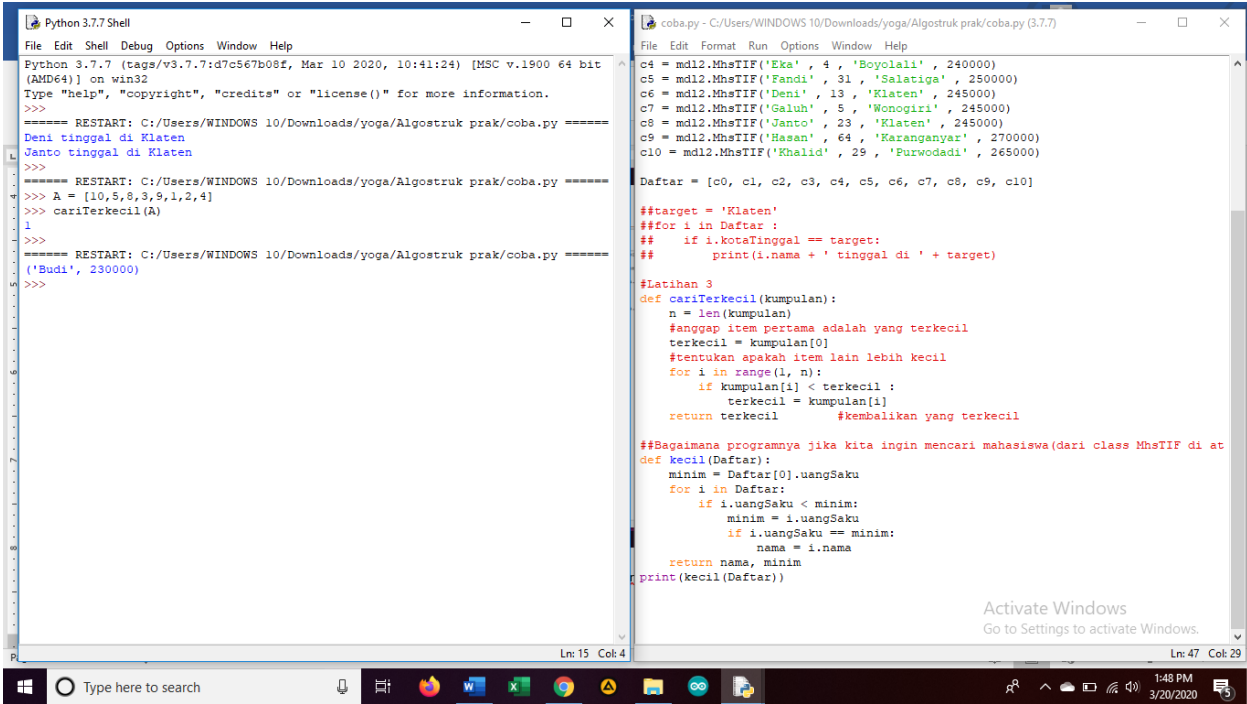
Latihan 2



• Latihan 3



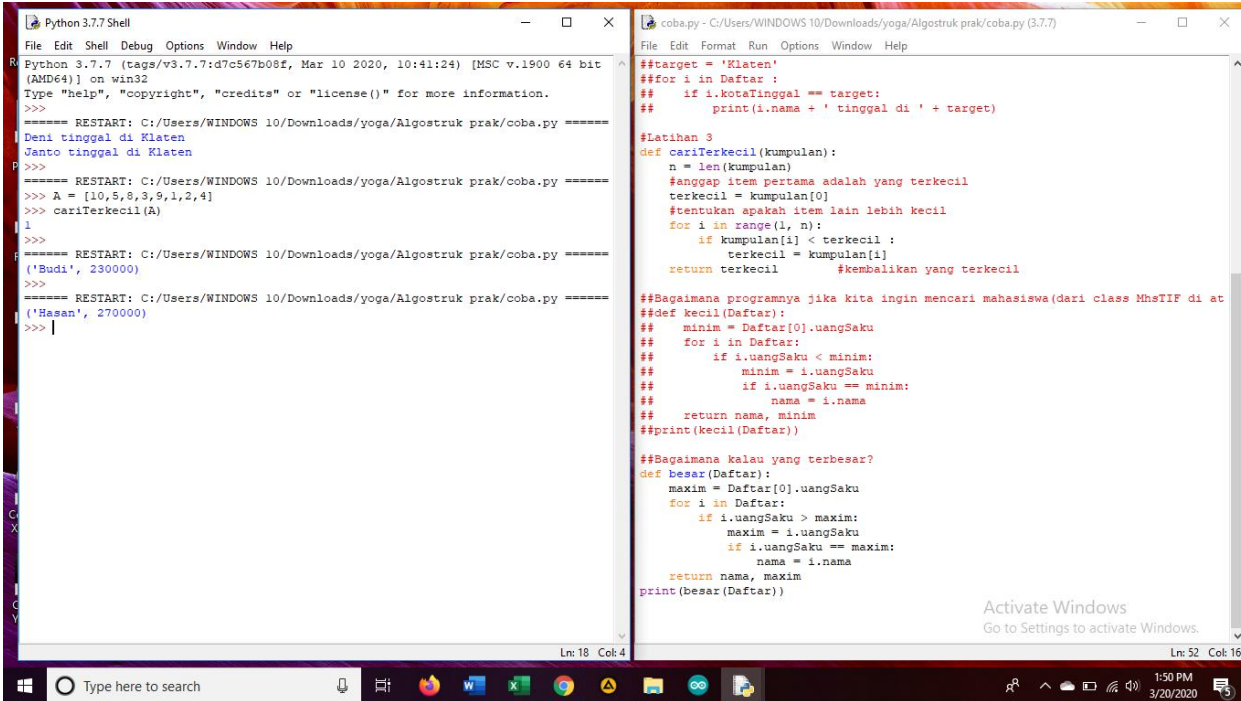
➤ Bagaimana programnya jika kita ingin mencari mahasiswa(dari class MhsTIF di atas) yang uang sakunya terkecil?



Program :

```
def kecil(Daftar):
    minim = Daftar[0].uangSaku
    for i in Daftar:
        if i.uangSaku < minim:
            minim = i.uangSaku
            if i.uangSaku == minim:
                nama = i.nama
    return nama, minim
print(kecil(Daftar))
```

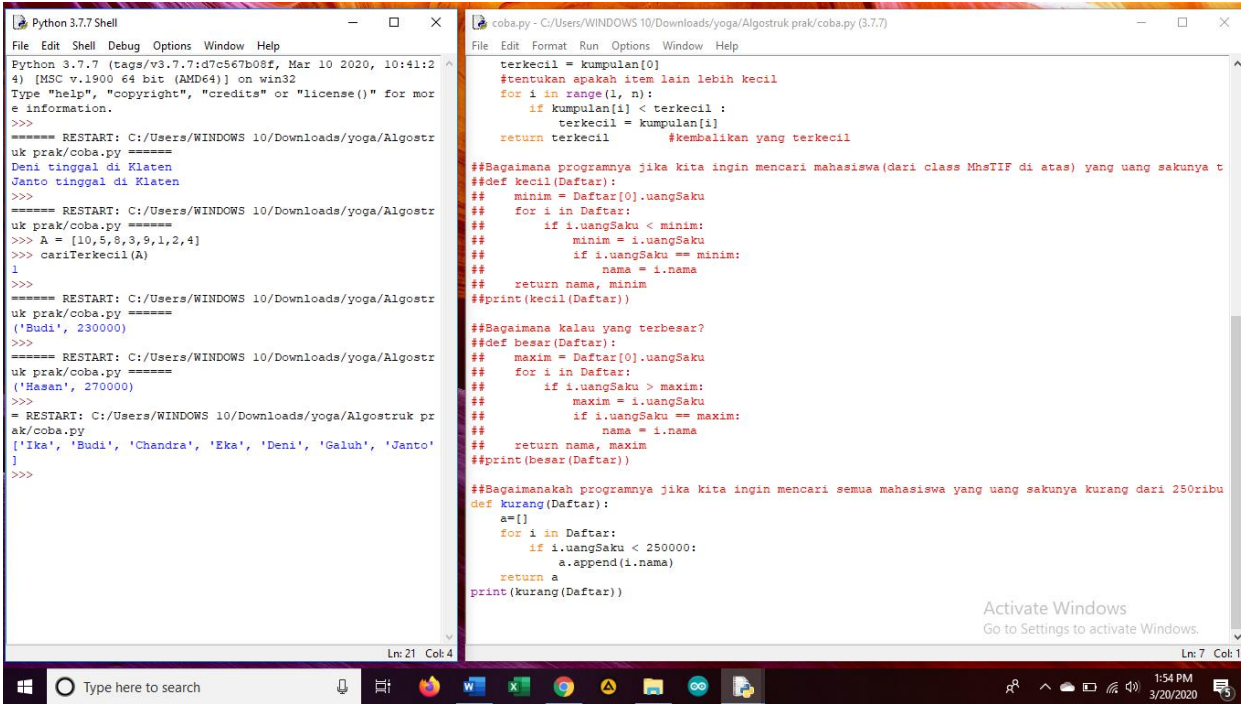
➤ Bagaimana kalau yang terbesar?



Program :

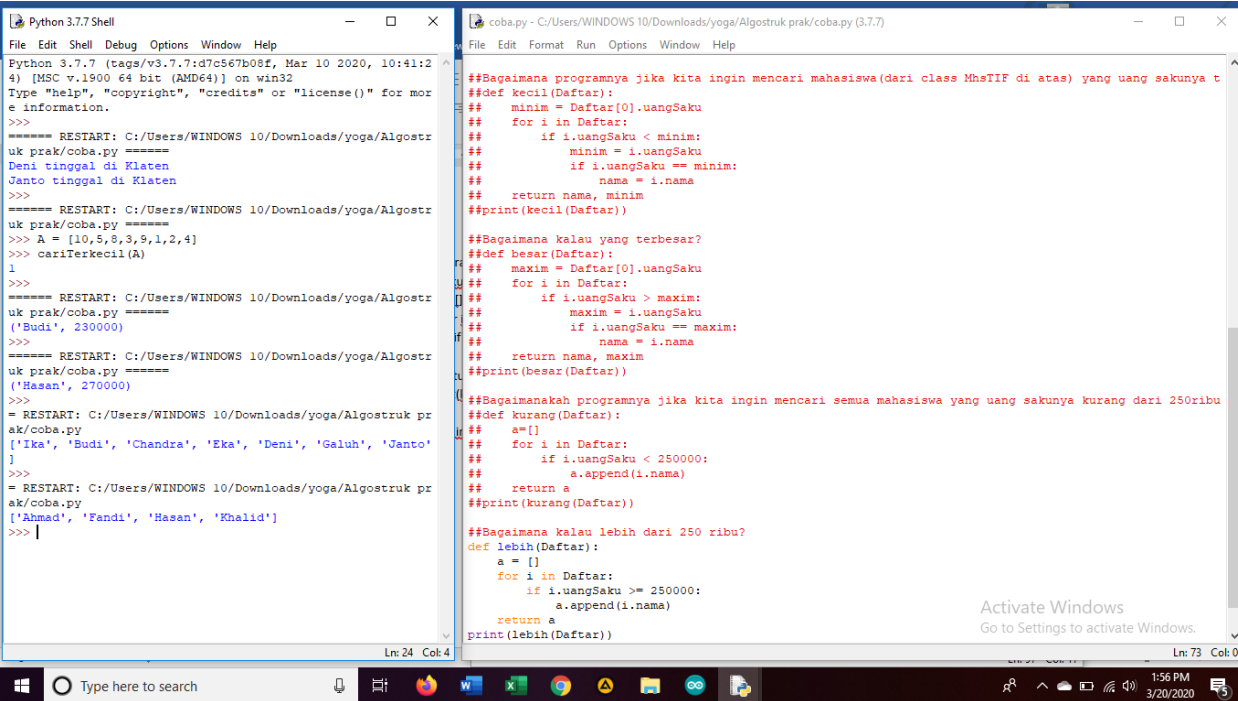
```
def besar(Daftar):
    maxim = Daftar[0].uangSaku
    for i in Daftar:
        if i.uangSaku > maxim:
            maxim = i.uangSaku
        if i.uangSaku == maxim:
            nama = i.nama
    return nama, maxim
print(besar(Daftar))
```

➤ Bagaimanakah programnya jika kita ingin mencari semua mahasiswa yang uang sakunya kurang dari 250ribu?



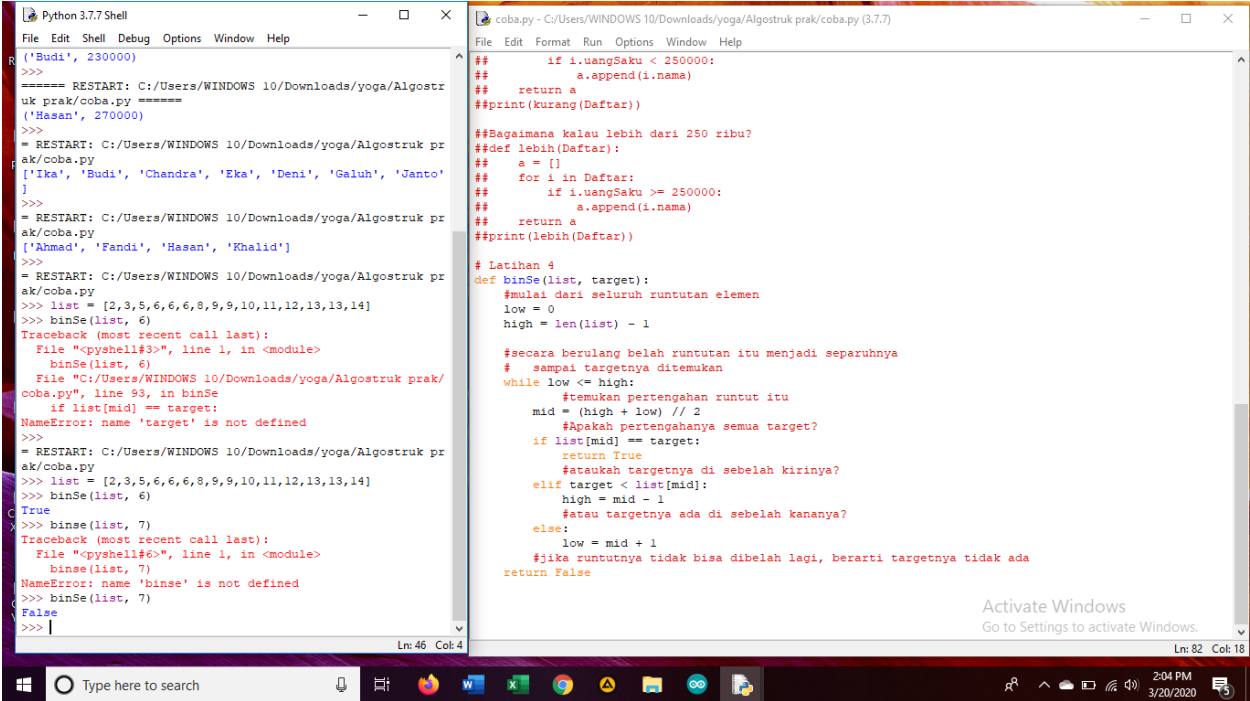
```
Program:
def kurang(Daftar):
    a=[]
    for i in Daftar:
        if i.uangSaku < 250000:
            a.append(i.nama)
    return a
print(kurang(Daftar))
```

➤ Bagaimana kalau lebih dari 250 ribu?

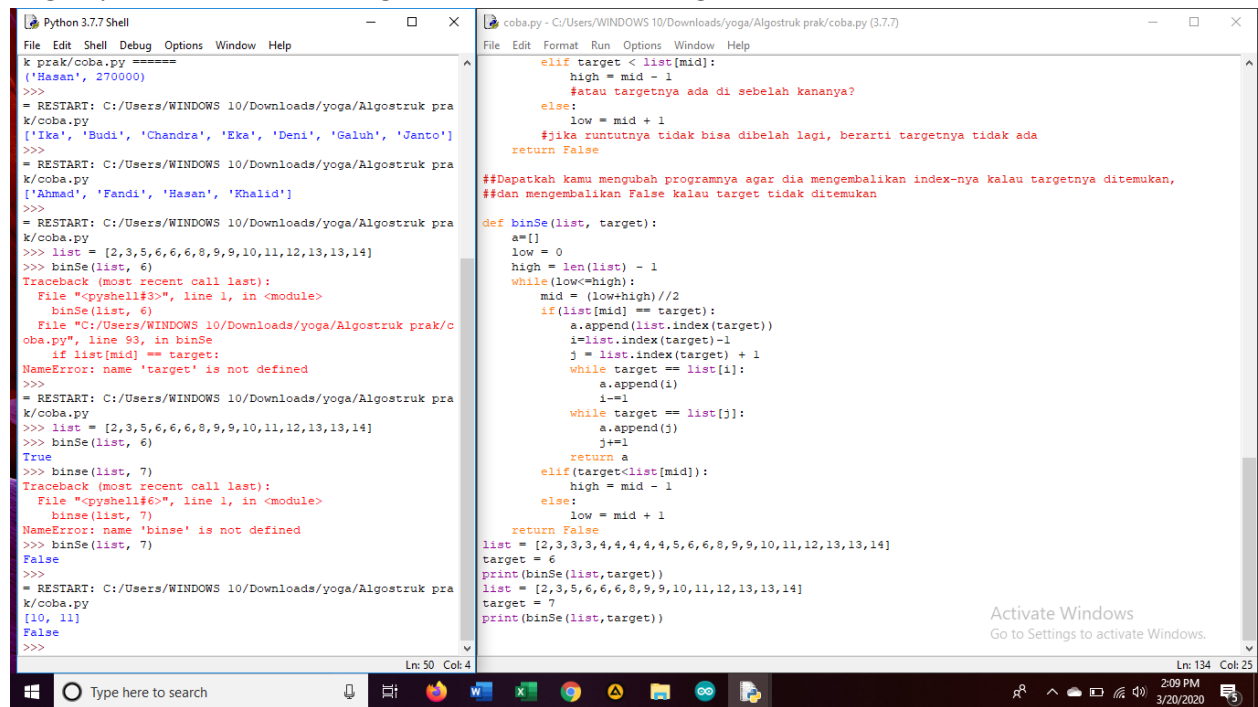


```
Program :
def lebih(Daftar):
    a = []
    for i in Daftar:
        if i.uangSaku >= 250000:
            a.append(i.nama)
    return a
print(lebih(Daftar))
```

• Latihan 4



- Dapatkah kamu mengubah programnya agar dia mengembalikan index-nya kalau targetnya ditemukan, dan mengembalikan False kalau target tidak ditemukan



The screenshot shows two windows. The left window is a Python 3.7.7 Shell with the following code and output:

```
k prak/coba.py =====
('Hasan', 270000)
>>>
= RESTART: C:/Users/WINDOWS 10/Downloads/yoga/Algostruk pra
k/coba.py
['Ika', 'Budi', 'Chandra', 'Eka', 'Deni', 'Galuh', 'Janto']
>>>
= RESTART: C:/Users/WINDOWS 10/Downloads/yoga/Algostruk pra
k/coba.py
['Ahmad', 'Fandi', 'Hasan', 'Khalid']
>>>
= RESTART: C:/Users/WINDOWS 10/Downloads/yoga/Algostruk pra
k/coba.py
>>> list = [2,3,5,6,6,6,6,8,9,9,10,11,12,13,13,14]
>>> binSe(list, 6)
Traceback (most recent call last):
  File "<pyshe11#>", line 1, in <module>
    binSe(list, 6)
  File "C:/Users/WINDOWS 10/Downloads/yoga/Algostruk prak/c
oba.py", line 93, in binSe
    if list[mid] == target:
NameError: name 'target' is not defined
>>>
= RESTART: C:/Users/WINDOWS 10/Downloads/yoga/Algostruk pra
k/coba.py
>>> list = [2,3,5,6,6,6,6,8,9,9,10,11,12,13,13,14]
>>> binSe(list, 6)
True
>>> binSe(list, 7)
Traceback (most recent call last):
  File "<pyshe11#>", line 1, in <module>
    binSe(list, 7)
NameError: name 'binse' is not defined
>>> binSe(list, 7)
False
>>>
= RESTART: C:/Users/WINDOWS 10/Downloads/yoga/Algostruk pra
k/coba.py
[10, 11]
False
>>>
```

The right window is a Python script editor for 'coba.py' with the following code:

```
elif target < list[mid]:
    high = mid - 1
    #atau targetnya ada di sebelah kananya?
else:
    low = mid + 1
    #jika runtutnya tidak bisa dibelah lagi, berarti targetnya tidak ada
    return False

##Dapatkah kamu mengubah programnya agar dia mengembalikan index-nya kalau targetnya ditemukan,
##dan mengembalikan False kalau target tidak ditemukan

def binSe(list, target):
    a=[]
    low = 0
    high = len(list) - 1
    while (low<=high):
        mid = (low+high)//2
        if (list[mid] == target):
            a.append(list.index(target))
            i=list.index(target)-1
            j = list.index(target) + 1
            while target == list[i]:
                a.append(i)
                i-=1
            while target == list[j]:
                a.append(j)
                j+=1
            return a
        elif (target<list[mid]):
            high = mid - 1
        else:
            low = mid + 1
    return False

list = [2,3,3,3,4,4,4,4,4,5,6,6,8,9,9,10,11,12,13,13,14]
target = 6
print(binSe(list,target))
list = [2,3,5,6,6,6,6,8,9,9,10,11,12,13,13,14]
target = 7
print(binSe(list,target))
```

Program :

```
def binSe(list, target):
    a=[]
    low = 0
    high = len(list) - 1
    while (low<=high):
        mid = (low+high)//2
        if (list[mid] == target):
            a.append(list.index(target))
            i=list.index(target)-1
            j = list.index(target) + 1
            while target == list[i]:
                a.append(i)
                i-=1
            while target == list[j]:
                a.append(j)
                j+=1
            return a
        elif (target<list[mid]):
            high = mid - 1
        else:
            low = mid + 1
    return False
```