

Nama : Rahmat Beny Susanto

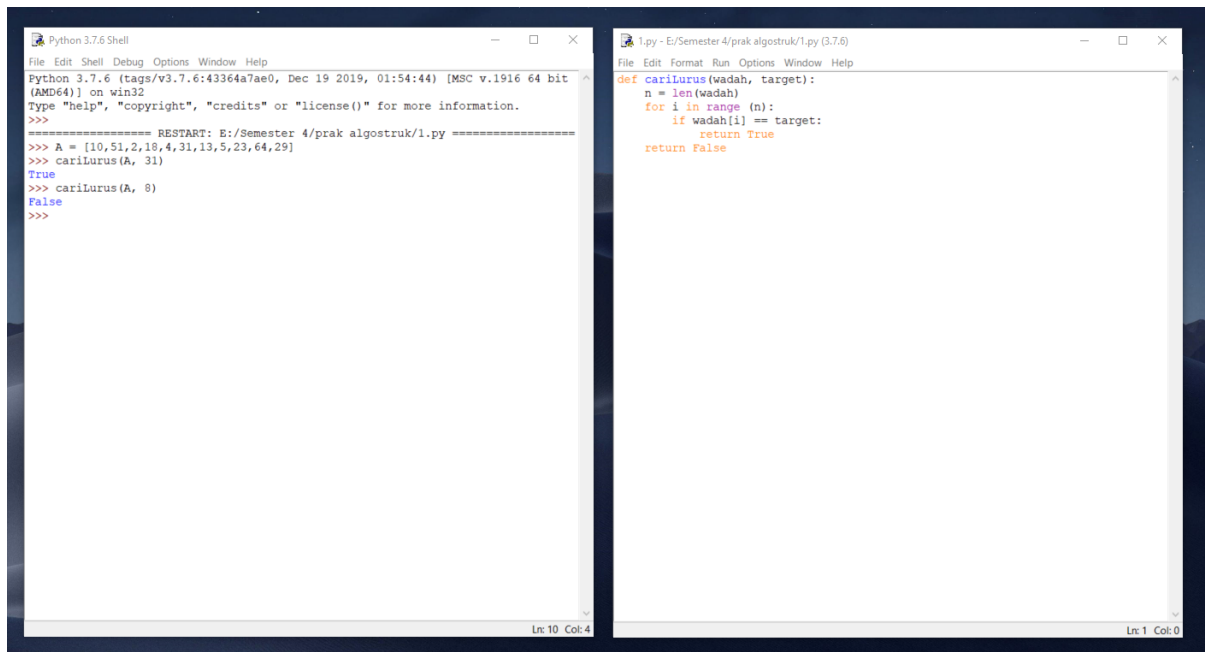
NIM : L200180079

Kelas : C

## MODUL 4

## LATIHAN

### 4.1 Linear Search

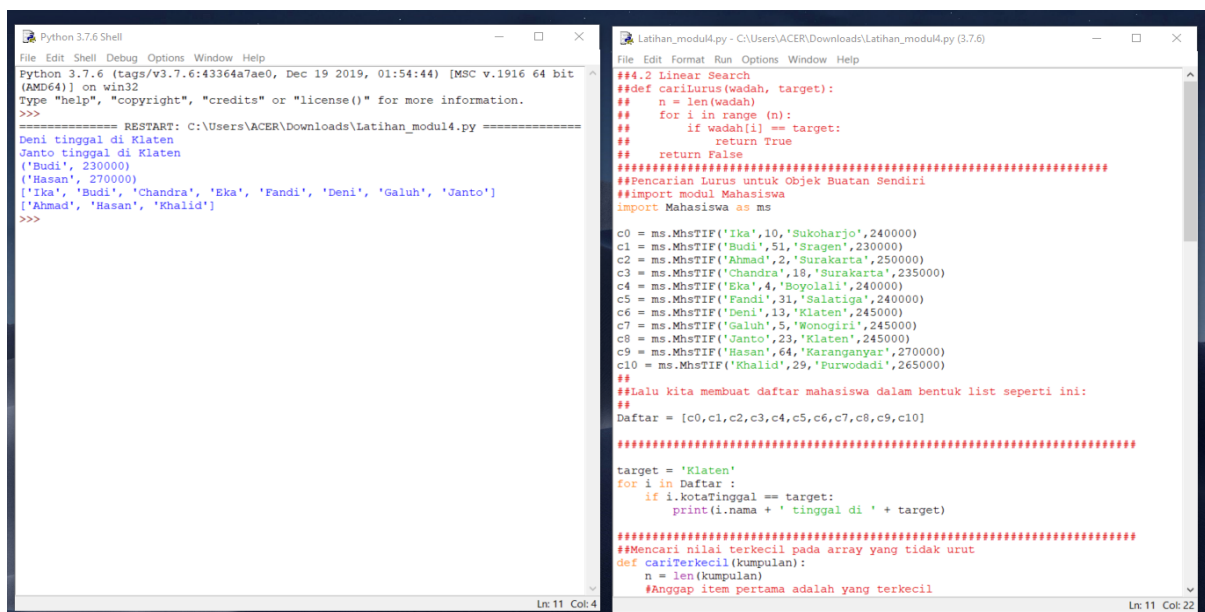


```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 01:54:44) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/Semester 4/prak algostruk/1.py =====
>>> A = [10,51,2,18,4,31,13,5,23,64,29]
>>> carilurus(A, 31)
True
>>> carilurus(A, 8)
False
>>>
```

```
1.py - E:/Semester 4/prak algostruk/1.py (3.7.6)
File Edit Format Run Options Window Help
def carilurus(wadah, target):
    n = len(wadah)
    for i in range(n):
        if wadah[i] == target:
            return True
    return False
```

Pencarian lurus untuk objek buatan sendiri

Mencari mahasiswa yang beralamat di 'Klaten'



```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 01:54:44) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ACER\Downloads\Latihan_modul4.py =====
Deni tinggal di Klaten
Janto tinggal di Klaten
('Budi', 230000)
('Hasan', 270000)
('Eka', 'Budi', 'Chandra', 'Eka', 'Fandi', 'Deni', 'Galuh', 'Janto')
['Ahmad', 'Hasan', 'Khalid']
>>>
```

```
Latihan_modul4.py - C:\Users\ACER\Downloads\Latihan_modul4.py (3.7.6)
File Edit Format Run Options Window Help
def carilurus(wadah, target):
    n = len(wadah)
    for i in range(n):
        if wadah[i] == target:
            return True
    return False

#####
##4.2 Linear Search
##
n = len(wadah)
for i in range(n):
    if wadah[i] == target:
        return True
return False

#####
##Pencarian Lurus untuk Objek Buatan Sendiri
##Import modul Mahasiswa
import Mahasiswa as ms

c0 = ms.MhsTIF('Ika',10,'Sukoharjo',240000)
c1 = ms.MhsTIF('Budi',51,'Sragen',230000)
c2 = ms.MhsTIF('Ahmad',2,'Surakarta',250000)
c3 = ms.MhsTIF('Chandra',18,'Surakarta',235000)
c4 = ms.MhsTIF('Eka',4,'Boyolali',240000)
c5 = ms.MhsTIF('Fandi',31,'Salatiga',240000)
c6 = ms.MhsTIF('Deni',13,'Klaten',245000)
c7 = ms.MhsTIF('Galuh',5,'Wonogiri',245000)
c8 = ms.MhsTIF('Janto',23,'Klaten',245000)
c9 = ms.MhsTIF('Hasan',64,'Karanganyar',270000)
c10 = ms.MhsTIF('Khalid',29,'Purwodadi',265000)

##
##Lalu kita membuat daftar mahasiswa dalam bentuk list seperti ini:
##
Daftar = [c0,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10]

#####
target = 'Klaten'
for i in Daftar:
    if i.kotaTinggal == target:
        print(i.nama + ' tinggal di ' + target)

#####
##Mencari nilai terkecil pada array yang tidak urut
def cariTerkecil(kumpulan):
    n = len(kumpulan)
    Anggap item pertama adalah yang terkecil
```

## Pencarian lurus linked-list

- Mencari nilai terkecil pada array yang tidak urut

```
def cariTerkecil(kumpulan):
    n = len(kumpulan)
    #Anggap item pertama adalah yang terkecil
    terkecil = kumpulan[0]
    #tentukan apakah item lain lebih kecil
    for i in range(1,n):
        if kumpulan[i] < terkecil:
            terkecil = kumpulan[i]

    return terkecil #kembalikan yang terkecil

>>> A = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
>>> cariTerkecil(A)
2
```

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

```
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))
```

```
('Budi', 230000)
```

- Bagaimana kalau yang terbesar?

```
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))
```

```
('Hasan', 270000)
```

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

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

```
['Ika', 'Budi', 'Chandra', 'Eka', 'Fandi', 'Deni', 'Galuh', 'Janto']
```

- Yang lebih dari 250 ribu?

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

```
['Ahmad', 'Hasan', 'Khalid']
```

## 4.2 Binary search

```
def binSe(kumpulan, target):
    #mulai dari seluruh runtutan elemen
    low = 0
    high = len(kumpulan) - 1

    #secara berulang belah runtutan itu menjadi separuhnya
    # sampai targetnya ditemukan
    while low <= high:
        #temukan pertengahan runtut itu
        mid = (high + low) // 2
        #Apakah pertengahanya semua target?
        if kumpulan[mid] == target:
            return True
        #ataukah targetnya di sebelah kirinya?
        elif target < kumpulan[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
```

```
>>> kumpulan = [2,3,5,6,6,6,8,9,9,10,11,12,13,13,14]
>>> target = 6
>>> print(binSe(kumpulan,target))
True
>>>
>>> kumpulan = [2,3,5,6,6,6,8,9,9,10,11,12,13,13,14]
>>> target = 7
>>> print(binSe(kumpulan ,target))
False
```