

LAPORAN
PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
(MODUL 4)
“ PENCARIAN ”



Disusun oleh :

NAMA : CINDI DILA APRILIANA

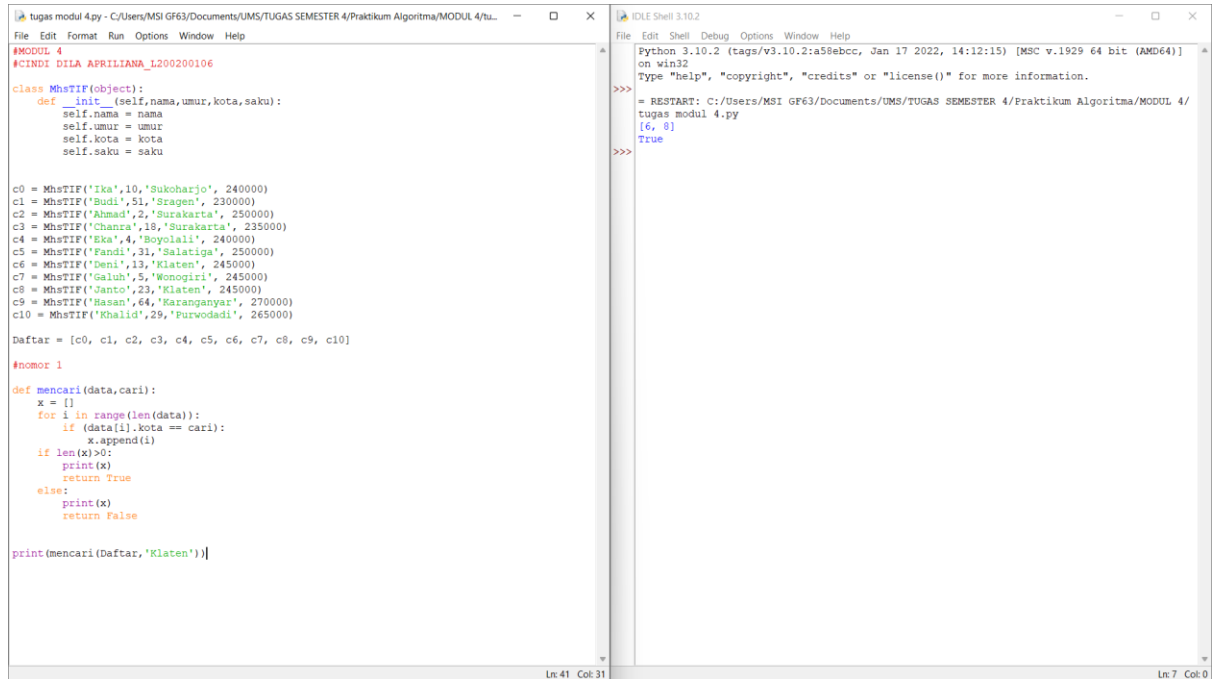
NIM : L200200106

KELAS : E

INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA

TAHUN 2021/2022

1. Buatlah suatu fungsi pencarian yang, alih-alih mengembalikan True/False, mengembalikan semua index lokasi elemen yang dicari. Jadi, misal pada list daftar mahasiswa di halaman 40 kita mencari mahasiswa yang berasal dari Klaten, kita akan mendapatkan [6, 8]. Kalau yang dicari tidak ditemukan, fungsi ini akan mengembalikan list kosong



The screenshot shows a Python IDE with two windows. The left window displays a Python script for a student class and a search function. The right window shows the output of the search function.

```
#MODUL 4
#CINDI DILA APRILIANA_L200200106

class MhsTIF(object):
    def __init__(self,nama,umur,kota,saku):
        self.nama = nama
        self.umur = umur
        self.kota = kota
        self.saku = saku

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

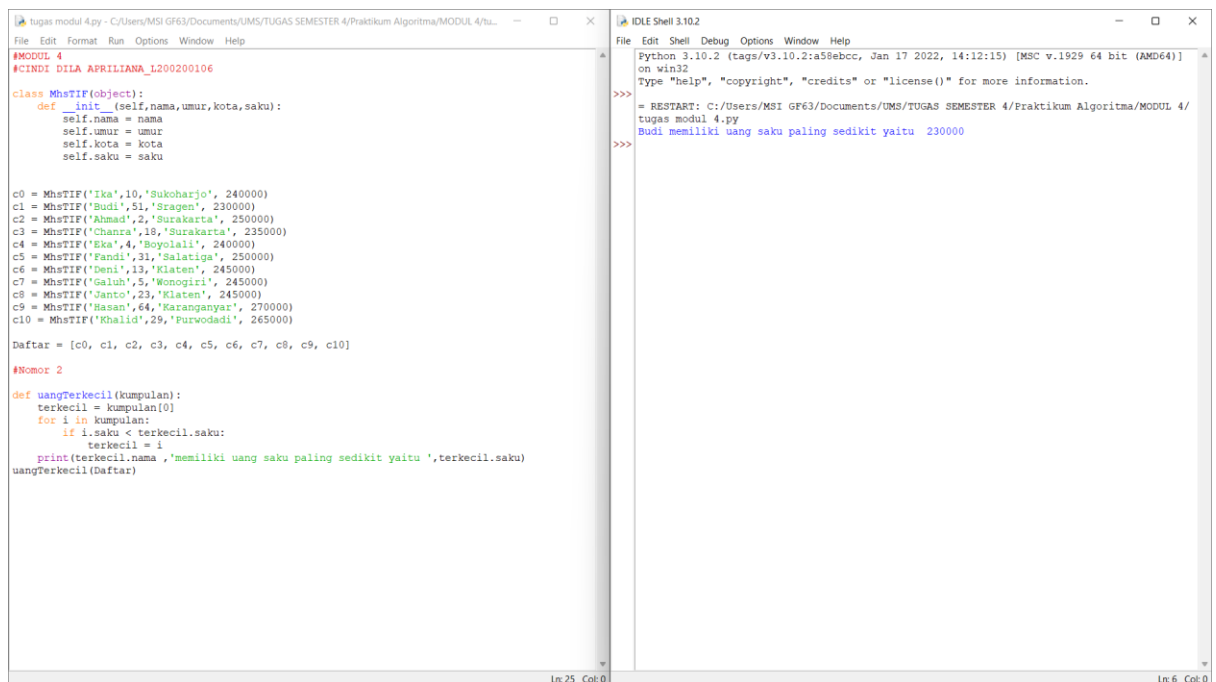
Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]

#nomor 1
def mencari(data,cari):
    x = []
    for i in range(len(data)):
        if (data[i].kota == cari):
            x.append(i)
    if len(x)>0:
        print(x)
        return True
    else:
        print(x)
        return False

print(mencari(Daftar,'Klaten'))
```

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 4/
tugas modul 4.py
[6, 8]
True
>>>
```

2. Dari list daftar mahasiswa di atas, buatlah fungsi untuk menemukan uang saku yang terkecil di antara mereka.



The screenshot shows a Python IDE with two windows. The left window displays a Python script for a student class and a function to find the minimum pocket money. The right window shows the output of the function.

```
#MODUL 4
#CINDI DILA APRILIANA_L200200106

class MhsTIF(object):
    def __init__(self,nama,umur,kota,saku):
        self.nama = nama
        self.umur = umur
        self.kota = kota
        self.saku = saku

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

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

#Nomor 2
def uangTerkecil(kumpulan):
    terkecil = kumpulan[0]
    for i in kumpulan:
        if i.saku < terkecil.saku:
            terkecil = i
    print(terkecil.nama , 'memiliki uang saku paling sedikit yaitu ',terkecil.saku)
uangTerkecil(Daftar)
```

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 4/
tugas modul 4.py
Budi memiliki uang saku paling sedikit yaitu 230000
>>>
```

3. Ubah program di atas agar mengembalikan objek mahasiswa yang mempunyai uang saku terkecil. Jika ada lebih dari satu mahasiswa yang uang sakunya terkecil, semua objek mahasiswa itu dikembalikan.

The screenshot shows a Python IDE with two windows. The left window, titled 'tugas modul 4.py', contains the following code:

```
#MODUL 4
#CINDI DILA APRILIANA_1200200106

class MhsTIF(object):
    def __init__(self,nama,umur,kota,saku):
        self.nama = nama
        self.umur = umur
        self.kota = kota
        self.saku = saku

c0 = MhsTIF('Ika',10,'Sukoharjo', 240000)
c1 = MhsTIF('Budi',51,'Sragen', 230000)
c2 = MhsTIF('Ahmad',2,'Surakarta', 250000)
c3 = MhsTIF('Chanra',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('Basan',64,'Karanganyar', 270000)
c10 = MhsTIF('Kholid',29,'Purwodadi', 245000)

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

#Nomor 3

def sakuTerkecil2(a):
    terkecil = Daftar[0].saku
    list = []
    for i in range(len(a)):
        if a[i].saku < terkecil:
            terkecil = a[i].saku
    for i in range(len(a)):
        if a[i].saku == terkecil:
            list.append(a[i].nama)
    return list
```

The right window, titled 'IDLE Shell 3.10.2', shows the execution output:

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 4/
tugas modul 4.py
>>> sakuTerkecil2(Daftar)
['Budi']
>>>
```

4. Buatlah suatu fungsi yang mengembalikan semua objek mahasiswa yang uang sakunya kurang dari 250000.

The screenshot shows a Python IDE with two windows. The left window, titled 'tugas modul 4.py', contains the following code:

```
#MODUL 4
#CINDI DILA APRILIANA_1200200106

class MhsTIF(object):
    def __init__(self,nama,umur,kota,saku):
        self.nama = nama
        self.umur = umur
        self.kota = kota
        self.saku = saku

c0 = MhsTIF('Ika',10,'Sukoharjo', 240000)
c1 = MhsTIF('Budi',51,'Sragen', 230000)
c2 = MhsTIF('Ahmad',2,'Surakarta', 250000)
c3 = MhsTIF('Chanra',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('Basan',64,'Karanganyar', 270000)
c10 = MhsTIF('Kholid',29,'Purwodadi', 245000)

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

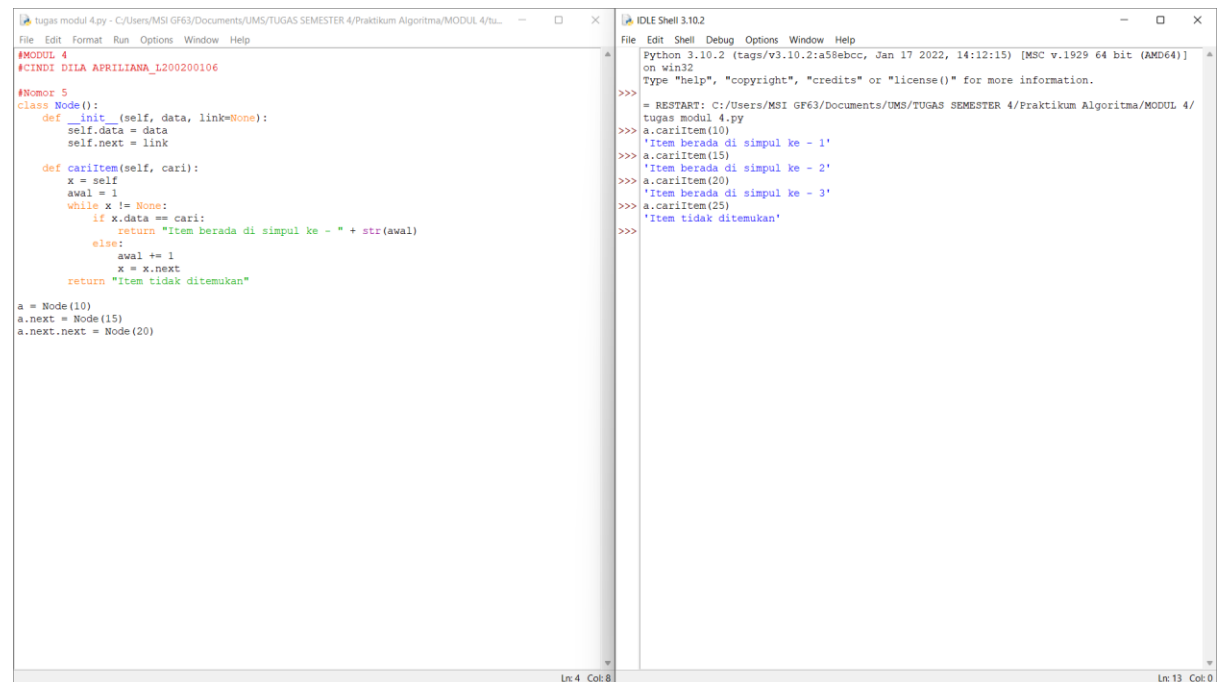
#Nomor 4

def sakuKurang():
    list = []
    for i in range(len(Daftar)):
        if Daftar[i].saku < 250000:
            list.append(Daftar[i].nama)
    for x in list:
        print(x, end=' ')
```

The right window, titled 'IDLE Shell 3.10.2', shows the execution output:

```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 4/
tugas modul 4.py
>>> sakuKurang()
Ika, Budi, Chanra, Eka, Deni, Galuh, Janto,
>>>
```

5. Buatlah suatu program untuk mencari suatu item di sebuah linked list.



The screenshot shows a Python IDE with two windows. The left window, titled 'tugas modul 4.py', contains the following code:

```
#MODUL 4
#CINDI DILA APRILIANA_L200200106

#Nomor 5
class Node():
    def __init__(self, data, link=None):
        self.data = data
        self.next = link

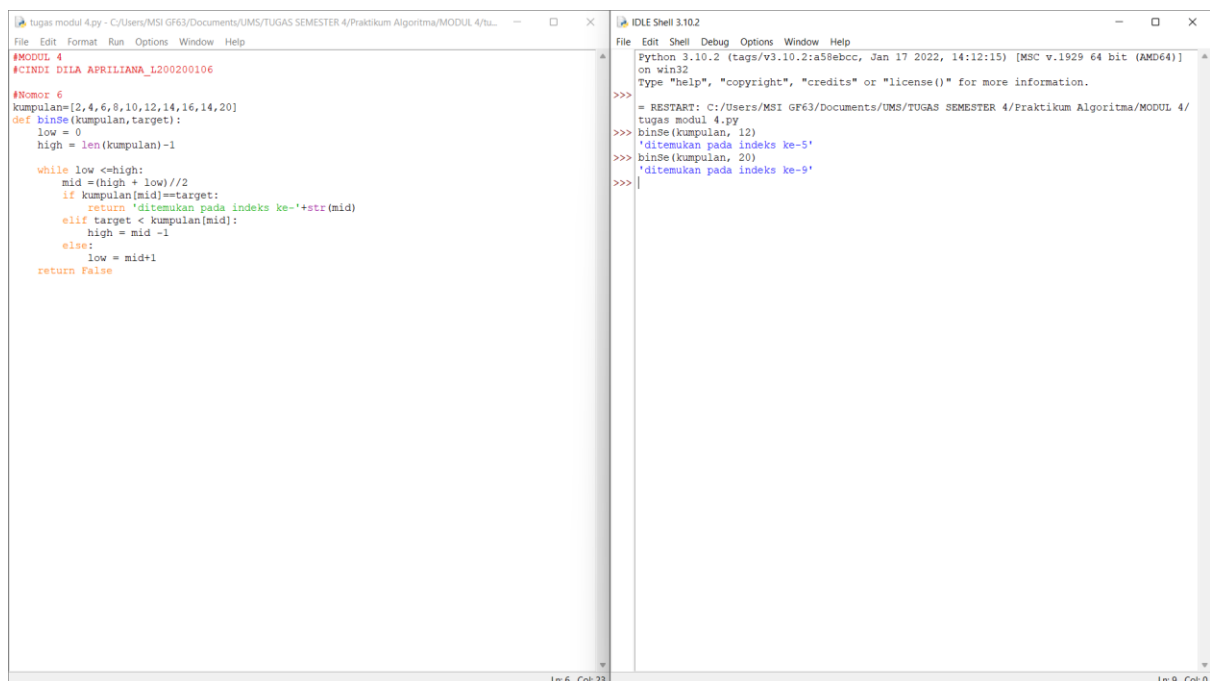
    def cariItem(self, cari):
        x = self
        awal = 1
        while x != None:
            if x.data == cari:
                return "Item berada di simpul ke - " + str(awal)
            else:
                awal += 1
                x = x.next
        return "Item tidak ditemukan"

a = Node(10)
a.next = Node(15)
a.next.next = Node(20)
```

The right window, titled 'IDLE Shell 3.10.2', shows the execution of the code:

```
>>> = RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 4/
tugas modul 4.py
>>> a.cariItem(10)
'Item berada di simpul ke - 1'
>>> a.cariItem(15)
'Item berada di simpul ke - 2'
>>> a.cariItem(20)
'Item berada di simpul ke - 3'
>>> a.cariItem(25)
'Item tidak ditemukan'
>>>
```

6. Binary search. Ubahlah fungsi binSe di halaman 43 agar mengembalikan index lokasi elemen yang ditemukan. Kalau tidak ketemu, akan mengembalikan False.



The screenshot shows a Python IDE with two windows. The left window, titled 'tugas modul 4.py', contains the following code:

```
#MODUL 4
#CINDI DILA APRILIANA_L200200106

#Nomor 6
kumpulan=[2,4,6,8,10,12,14,16,14,20]
def binSe(kumpulan, target):
    low = 0
    high = len(kumpulan)-1

    while low <= high:
        mid = (high + low) // 2
        if kumpulan[mid] == target:
            return 'ditemukan pada indeks ke-' + str(mid)
        elif target < kumpulan[mid]:
            high = mid - 1
        else:
            low = mid + 1
    return False
```

The right window, titled 'IDLE Shell 3.10.2', shows the execution of the code:

```
>>> = RESTART: C:/Users/MSI GF63/Documents/UMS/TUGAS SEMESTER 4/Praktikum Algoritma/MODUL 4/
tugas modul 4.py
>>> binSe(kumpulan, 12)
'ditemukan pada indeks ke-5'
>>> binSe(kumpulan, 20)
'ditemukan pada indeks ke-9'
>>>
```

7. Binary search. Ubahlah fungsi binSe itu agar mengembalikan semua index lokasi elemen yang ditemukan. Contoh: mencari angka 6 pada list [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14] akan mengembalikan [3, 4, 5]. Karena sudahurut, “tinggal melihat kiri dan kanannya”.

```
#MODUL 4
#CINDI DILA APRILIANA_L2002200106

#Nomor 7
data = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]

def binSe2(kumpulan, target):
    low = 0
    high = len(kumpulan)-1
    ketemu = False
    x = []
    while low <= high and not ketemu :
        mid = (high + low)//2
        if kumpulan[mid] == target:
            ketemu = True
        elif target < kumpulan[mid]:
            high = mid - 1
        else:
            low = mid + 1
    if not ketemu:
        print('Data tidak ditemukan')
    for i in range(len(kumpulan)):
        if kumpulan[i] == target:
            x.append(i)
            mid+=1
    return x

>>> binSe2(data,6)
[3, 4, 5]
>>> binSe2(data,13)
[11, 14]
>>> binSe2(data,11)
[10]
```

8. Pada permainan tebak angka yang sudah kamu buat di Modul 1 (soal nomer 12, halaman 16), kalau angka yang harus ditebak berada di antara 1 dan 100, seharusnya maksimal jumlah tebakan adalah 7. Kalau antara 1 dan 1000, maksimal jumlah tebakan adalah 10. Mengapa seperti itu? Bagaimanakah polanya?

Jawab :

Karena jika kita menebak dengan tepat membagi 2 range tersebut secara terus menerus sehingga memperkecil kemungkinan range yang dapat ditebak .misal ada angka 78(kita tidak tahu)

Tebakan pertama 50 : ternyata terlalu kecil

Tebakan kedua 75 : masih terlalu kecil

Tebakan ketiga 88 : terlalu besar

Tebakan keempat 82 : terlalu besar

Tebakan kelima 78 : benar