

**ALGORITMA DAN STRUKTUR DATA
MODUL 4
PENCARIAN**



Disusun oleh:
Muhammad Ferizal Fadhli
L200210119
D

**TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2022/2023**

Tugas Mahasiswa:

1. Buatlah suatu fungsi pencarian allih alih mengembalikan true/false, mengeembalikana semua index lokaasi elemeen yang dicari. Jadi, missal padada list daftara mahasiswa di halaman 40 kita mencari yang berasal dari Klaten, kita akaan mendapatkan [6,8]. Kalua yang dicari tidak ditemukan, fungsi ini akan mengembalikan list kosong.

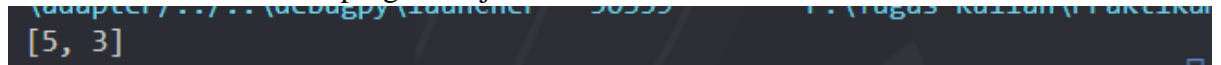
Berikut adalah kode program :

```
c0 = ['bang ram',10, "Sukoharjo"]
c1 = ['budi',121, "Sragen"]
c2 = ['Ahmad',121, "Surakarta"]
c3 = ['Chandra',121, "Surakarta"]
c4 = ['Fandi',121, "Boyolali"]
c5 = ['Deni',121, "Klaten"]
c6 = ['Galuh',121, "Wonogiri"]
c7 = ['Janto',121, "rembang"]
c8 = ['Hasan',121, "rembang"]
c9 = ['Khalid',121, "rembang"]
c10 = ['Muhammad Sumbul',121, "rembang"]

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9 ,c10]
lokasi = "Klaten"
def search_students(Daftar, lokasi):
    indexes = []
    for i in range(len(Daftar)):
        if Daftar[i][2] == lokasi:
            indexes.append(i)
            indexes.append(len(Daftar[i]))
    return indexes

print(search_students(Daftar, lokasi))
```

Berikut adalah SS Ketika program dijalankan :



```
[5, 3]
```

2. Dari list daftar mahasiswa diatas, buatlah fungsi untuk menemukan uang saku yang terkecil diantara mereka.

Berikut adalah kode program :

```
c0 = Mahasiswa('bang ram',10, "Sukoharjo", 234234234, [])
c1 = Mahasiswa('budi',121, "Sragen", 2222222222220, [])
c2 = Mahasiswa('Ahmad',121, "Surakarta", 111111, [])
c3 = Mahasiswa('Chandra',121, "Surakarta", 666666, [])
c4 = Mahasiswa('Fandi',121, "Boyolali", 666, [])
c5 = Mahasiswa('Deni',121, "Klaten", 1000000000000000000, [])
```

```

c6 = Mahasiswa('Galuh',121, "Wonogiri", 10000000000000000, [])
c7 = Mahasiswa('Janto',121, "rembang", 10000000000000000, [])
c8 = Mahasiswa('Hasan',121, "rembang", 10000000000000000, [])
c9 = Mahasiswa('Khalid',121, "rembang", 10000000000000000, [])
c10 = Mahasiswa('Muhammad Sumbul',121, "rembang", 10000000000000000,
[])

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

```

```

def cariUangSakuTerkecil(Daftar):
    terkecil = Daftar[0].uangSaku
    for i in Daftar:
        if i.uangSaku < terkecil:
            terkecil = i.uangSaku
    return terkecil
print(cariUangSakuTerkecil(Daftar))

```

Berikut SS ketikaa program dijalankan :

666

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

Berikut adalah kode program :

```

def namauangsakuterkecil(Daftar):
    terkecil = Daftar[0].uangSaku
    for i in Daftar:
        if i.uangSaku < terkecil:
            terkecil = i.uangSaku
            terkecilNama = i.nama
    return terkecilNama
print("Yang mempunyai uang saku terkecil  
adalah",namauangsakuterkecil(Daftar))

```

Berikut SS program Ketika dijalankan :

Yang mempunyai uang saku terkecil adalah Fandi

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

Berikut adalah kode program ;

```

def uangsakukurangdari250(Daftar):
    for i in Daftar:
        if i.uangSaku < 250000:
            print(i.nama + ' Uang saku kurang dari 250000')

```

```
uangsakukurangdari250(Daftar)
```

Berikut SS program ketika dijalankan :

```
Ahmad Uang saku kurang dari 250000  
Eandi Uang saku kurang dari 250000
```

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

Berikut kode program :

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def push(self, data):
        new_node = Node(data)
        new_node.next = self.head
        self.head = new_node

    def search(self, item):
        current = self.head
        index = 0
        while current is not None:
            if current.data == item:
                return index
            current = current.next
            index += 1
        return None

linked_list = LinkedList()
linked_list.push(3)
linked_list.push(5)
linked_list.push(7)

item_to_search = 5
result = linked_list.search(item_to_search)
if result is not None:
    print(f"Item {item_to_search} ditemukan pada indeks ke-{result}")
else:
    print(f"Item {item_to_search} tidak ditemukan")
```

Berikut SS Ketika programa dijalankan :

```
\\adapter7\..\..\debugpy\launcher 50598 -- F:\Tugas Kuliah\Praktikum\A  
Item 5 ditemukan pada indeks ke-1  
PS F:\Tugas Kuliah\Praktikum\Algoritma dan Struktur Data\Pertemuan 4>
```