

Nama : Riyan Sutantio Bangkit Nugroho

NIM : L200180180

Modul 4

Praktikum Algoritma dan Struktur Data

```
praktikum modul 4.py - C:/Users/ssus/AppData/Local/Programs/Python/Python38-32/praktikum modul 4.py (3...
File Edit Format Run Options Window Help
class MhsTIF(object):
    def __init__(self, nama, angka, alamat, uangsaku):
        self.Nama = nama
        self.Angka = angka
        self.Alatamat = alamat
        self.Uangsaku = uangsaku

c0 = MhsTIF('Ika', 10, 'Sukoharjo', 240000)
c1 = MhsTIF('Budi', 51, 'Surabaya', 230000)
c2 = MhsTIF('Ahmad', 2, 'Surakarta', 250000)
c3 = MhsTIF('Chandra', 18, 'Surakarta', 235000)
c4 = MhsTIF('Eka', 4, 'Bojonegara', 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('Kholid', 29, 'Purwodadi', 265000)

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]
S = [1, 2, 3, 10, 15, 18, 73, 87, 89, 151, 164]
C = [2, 4, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]

#1
def carialamat(list, Kota):
    a = []
    for i in list:
        if i.Alatamat == Kota:
            a.append(list.index(i))
    return print("data berada pada nomor indeks ke : ", a, end=' ')

#2
def terkecil(list):
    nilai = list[0].Uangsaku
    for i in list[1:]:
        if i.Uangsaku < nilai:
            nilai = i.Uangsaku
    return nilai

#3
def uangSakuTerkecil(list):
    nilai = list[0].Uangsaku
    for i in list[1:]:
        if i.Uangsaku < nilai:
            nilai = i
        elif i.Uangsaku == nilai:
            nilai.append(i.Nama)
    return nilai

#4
def uangSakuKurang(list):
    nilai = []

praktikum modul 4.py - C:/Users/ssus/AppData/Local/Programs/Python/Python38-32/praktikum modul 4.py (3...
File Edit Format Run Options Window Help
#4
def uangSakuKurang(list):
    nilai = []
    for i in list:
        if i.Uangsaku < 250000:
            nilai.append(i.Nama)
    return nilai

#5
def carilinkedList(head, target):
    nilai = head
    while nilai.data != None:
        if nilai.data == target:
            return nilai
        return nilai

#6
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

#7
def binSeMass(kumpulan, target):
    nilai = []
    low = 0
    high = len(kumpulan)-1
    while low <= high:
        mid = (high+low)//2
        if kumpulan[mid] == target:
            midLeft = mid - 1
            while kumpulan[midLeft] == target:
                nilai.append(midLeft)
                midLeft = midLeft - 1
            midRight = mid + 1
            while kumpulan[midRight] == target:
                nilai.append(midRight)
                midRight = midRight + 1
            return nilai
        elif target < kumpulan[mid]:
            high = mid - 1
        else:
            low = mid + 1
```

```
praktikum modul 4.py - C:/Users/ssus/AppData/Local/Programs/Python/Python38-32/praktikum modul 4.py (3...
File Edit Format Run Options Window Help
#7
def binSeMass(kumpulan, target):
    nilai = []
    low = 0
    high = len(kumpulan)-1
    while low <= high:
        mid = (high+low)//2
        if kumpulan[mid] == target:
            midLeft = mid - 1
            while kumpulan[midLeft] == target:
                nilai.append(midLeft)
                midLeft = midLeft - 1
            midRight = mid + 1
            while kumpulan[midRight] == target:
                nilai.append(midRight)
                midRight = midRight + 1
            return nilai
        elif target < kumpulan[mid]:
            high = mid - 1
        else:
            low = mid + 1
    return False

#8
def cetak():
    print("""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
    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 3.8.0 Shell
File Edit Shell Debug Options Window Help
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!!!!

>>> carialamat(Daftar, "Surakarta")
data berada pada nomor indeks ke : [2, 3]
>>> terkecil(Daftar)
230000
>>> uangSakuTerkecil(Daftar)
[<_main_.MhsTIF object at 0x032C3040>]
>>> uangSakuKurang(Daftar)
['Ika', 'Budi', 'Chandra', 'Eka', 'Deni', 'Galuh', 'Janto']
>>> binSe(S, 151)
9
>>> binSeMass(C, 6)
[3, 4, 5]
>>> cetak()
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!!!!

>>>
```