

Nama : Muhammad Riza Radyaka Susanto

NIM : L200180136

Kelas = E

Latihan

```
>>> A = [10,51,2,18,4,31,13,5,23,64,29]
>>> cariLurus(A,31)
True
>>> cariLurus(A,8)
False
>>> target = 'Klaten'
>>> for i in Daftar:
    if i.kotaTinggal == target:
        print(i.nama + ' tinggal di ' + target)

Deni tinggal di Klaten
Janto tinggal di Klaten
```

```
>>> c = [2,4,5,10,13,18,23,29,31,51,64]
>>> binSe(c,10)
True
```

```
>>> a = [1,10,23,51,3,5]
>>> cariTerkecil(a)
1
```

Tugas

```
>>> cariIndex(Daftar,'Klaten')
[6]
[6, 8]
[6, 8]
>>> cariUangSakuTerkecil(Daftar)
230000
>>> uangSakuTerkecil()
['Budi']
>>> uangSakuTerkecil()
['Budi']
```

```

>>> n1 = node(11)
>>> n1 = node(12)
>>> n1.search(11)
Data 11 tidak ada dalam Linked List
>>> n1.search(12)
Data 12 tidak ada dalam Linked List
>>> n1.search(1)
Data 1 tidak ada dalam Linked List
>>> n1.search(10)
Data 10 tidak ada dalam Linked List
>>> n1.next = node(12)
>>> n1.search(11)
Data 11 tidak ada dalam Linked List
>>> n1.search(12)
Data 12 ada dalam Linked List
>>> a = [1,2,3,4,5,6]
>>> binSe(a,5)
4
>>> binSe(7)
Traceback (most recent call last):
  File "<pyshell#27>", line 1, in <module>
    binSe(7)
TypeError: binSe() missing 1 required positional argument: 'target'
>>> binSe(a,7)
False
>>> b = [1,2,5,6,6,8,9,9,10,11,12,13,13,14]
>>> binSel(b,6)
[3, 4]
>>> b [1,2,5,6,6,6,8,9,9,10,11,12,13,13,14]
Traceback (most recent call last):
  File "<pyshell#31>", line 1, in <module>
    b [1,2,5,6,6,6,8,9,9,10,11,12,13,13,14]
TypeError: list indices must be integers or slices, not tuple
>>> b = [1,2,5,6,6,6,8,9,9,10,11,12,13,13,14]
>>> binSel(b,6)
[3, 4, 5]

```

Nomer 8

Ada dua pola

Pertama 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!!!

Kedua menggunakan barisan geometri $S_n = 2^n$

barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64

Misal angka yang akan diebak adalah 68

Tebakan ke-1 : 64 dijawab lebih dari itu

Tebakan ke-2 : 96(dari 64 + 32) dijawab "Kurang dari itu"

Tebakan ke-3 : 80(dari 64 + 16) dijawab "Kurang dari itu"

Tebakan ke-4 : 72(dari 64 + 8) dijawab "Kurang dari itu"

Tebakan ke-5 : 68(dari 64 + 4) dijawab "Lebih dari itu"

Tebakan ke-6 : 70(dari 68 + 2) dijawab "TEPAT"