

## Soal Unguided Kamis, 11 Mei 2023

### Praktikum Algoritma dan Pemrograman

#### 01. Simpan Sebanyak-banyaknya

Ada sekumpulan file yang diketahui ukurannya. Seluruh file tersebut akan dimasukkan dalam suatu media penyimpanan yang memiliki kapasitas tertentu. Jika total ukuran file yang ada melebihi kapasitas, maka ada beberapa file yang tidak dimasukkan ke dalam media penyimpanan.

Untuk menyelesaikan masalah tersebut, buatlah fungsi **simpan(list\_ukuran, kapasitas)** yang dapat menentukan ada berapa maksimum file yang terdaftar di dalam **list\_ukuran** untuk dimasukkan pada suatu media penyimpanan yang memiliki batasan **kapasitas** tertentu. Silahkan perhatikan contoh yang diberikan.

#### Contoh:

list\_ukuran = [4, 4, 4, 3, 3] => berarti ada 5 file, masing-masing berukuran 4 GB, 4 GB, 4 GB, 3 GB dan 3 GB.

kapasitas=12 => kapasitas maksimal adalah 12 GB

Maka yang bisa masuk adalah  $4 + 4 + 4 = 12$ . Yaitu 3 file saja. Maka fungsi anda harus menghasilkan jawaban 3.

list\_ukuran = [1, 2, 1, 1, 2, 1, 2, 4, 5], kapasitas 11, maka jawaban yang benar adalah 7 file, yaitu  $1 + 2 + 1 + 1 + 2 + 1 + 2 = 10$ . Tidak bisa ditambah lagi, karena ukuran berikutnya adalah 4 (total = 14, melebihi batas).

#### Perhatikan:

- Fungsi anda harus return jumlah file maksimal yang bisa dimasukkan sesuai kapasitas yang diberikan. Tidak ada print() di dalam fungsi.
- Lakukan perhitungan mulai dari index ke-0 (dari depan).
- Jika ternyata kapasitas terlalu kecil untuk menampung ukuran file pertama, maka jawabannya adalah 0.
- `len(list_ukuran) >= 1`

#### For example:

Test	Result
<code>print(simpan([4, 4, 4, 3, 3], 12))</code>	3
<code>print(simpan([1, 2, 1, 1, 2, 1, 2, 4, 5], 11))</code>	7

## 02. Total Tanpa 13

Buatlah fungsi **total\_tanpa\_tigabelas(list)** yang dapat menghitung jumlah dari isi **list** yang diberikan mulai dari depan (index ke-0) dan berhenti jika mencapai angka 13 di dalam list tersebut. Silahkan lihat contoh yang diberikan untuk lebih jelasnya.

Contoh:

Jika diberikan list = **[1, 2, 2, 1]**, maka total dari seluruh isinya adalah **1 + 2 + 2 + 1 = 4**

Jika diberikan list = **[2, 4, 1, 3, 13, 8]**, maka totalnya adalah **2 + 4 + 1 + 3 = 10**. Berhenti karena ada angka **13**.

Perhatikan:

- len(list) minimal adalah 1.
- Fungsi harus return, jangan lakukan print output di dalam fungsi.

**For example:**

Test	Result
print(total_tanpa_tigabelas([1, 2, 2, 1]))	6
print(total_tanpa_tigabelas([2, 4, 1, 3, 13, 8]))	10

### 03. Keliling Segitiga

Anda akan diberikan informasi 3 titik koordinat yang akan membentuk sebuah segitiga. Informasi tersebut akan diberikan dalam bentuk sebuah dictionary { "namaTitik": [x, y] }.

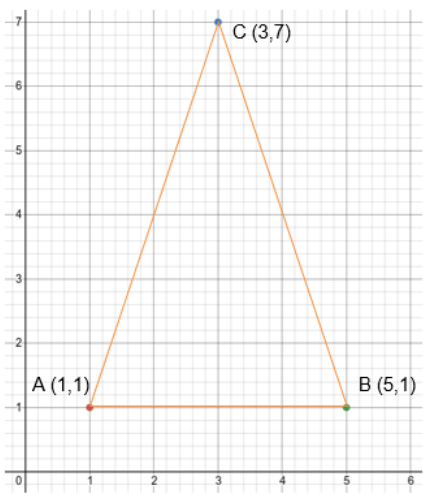
Tugas anda adalah membuat sebuah fungsi dengan nama **kelilingSegitiga(koordinat)** yang akan menampilkan nama dari segitiga tersebut beserta kelilingnya!

Contoh:

**Input:**

```
test = {"A" : [1,1], "B" : [5,1], "C" : [3,7] }
```

```
kelilingSegitiga(test)
```



**Output:**

Segitiga ABC memiliki keliling 16.6491

**For example:**

Test	Result
test = {"A" : [1,1], "B" : [5,1], "C" : [3,7] } kelilingSegitiga(test)	Segitiga ABC memiliki keliling 16.6491
test = {"J": [1,3], "K": [4,5], "L": [3,0]} kelilingSegitiga(test)	Segitiga JKL memiliki keliling 12.3101

Untuk menghitung jarak titik A (x1, y1) ke B (x2, y2) dapat menggunakan Euclidian Distance sebagai berikut:

$$\text{jarak } AB = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

#### 04. Hitung Nilai Akhir Berdasarkan n Nilai Terbaik

Buatlah fungsi `hitung_nilai_akhir(daftar_nilai, n)` yang dapat digunakan untuk menghitung nilai akhir berdasarkan rata-rata dari `n` nilai terbaik. Parameter `daftar_nilai` adalah sebuah dictionary dengan format seperti contoh berikut ini:

```
daftar_nilai = {  
'Udin' : [65, 74, 56, 80, 82, 94],  
'Atun' : [98, 84, 82, 88],  
'Tejo' : [85, 86]  
}
```

Udin memiliki 6 nilai yaitu 65, 74, 56, 80, 82 dan 94.

Atun memiliki 4 nilai yaitu 98, 84, 82, 88

Tejo memiliki 2 nilai, yaitu 85, 86

Contoh

##### Pemanggilan

```
daftar_nilai = {  
'Udin' : [65, 74, 56, 80, 82, 94],  
'Atun' : [98, 84, 82, 88],  
'Tejo' : [85, 86]  
}
```

`n = 2`

`hitung_nilai_akhir(daftar_nilai, n)`

##### Output

Atun 93.000000

Tejo 85.500000

Udin 88.000000

##### Penjelasan

`n = 2`, sehingga nilai akhir akan dihitung dari **2 nilai terbaik**.

Udin memiliki 6 nilai yaitu 65, 74, 56, 80, **82** dan **94**. Rata-rata =  $(94 + 82) / 2 = 88.000000$  (dalam format float %f)

Atun memiliki 4 nilai yaitu **98**, 84, 82, **88**. Rata-rata =  $(98 + 88) / 2 = 93.000000$  (dalam format float %f)

Tejo memiliki 2 nilai, yaitu **85**, **86**. Rata-rata =  $(85+86)/2 = 85.500000$  (dalam format float %f)

**Kemudian outputnya diurutkan berdasarkan nama, sehingga urutan outputnya adalah Atun, Tejo dan Udin.**

Batasan

- Fungsi anda tidak perlu return, lakukan `print()` di dalam fungsi.

- Dictionary daftar nilai minimal memuat 1 nama, kemudian jumlah nilai setiap nama minimal akan sama dengan n.
- Untuk menampilkan rata-rata gunakan format %f (float) supaya hasilnya seperti yang dicontohkan.

**For example:**

Test	Result
<pre>daftar_nilai = {     'Udin' : [65, 74, 56, 80, 82, 94],     'Atun' : [98, 84, 82, 88],     'Tejo' : [85, 86] } n = 2 hitung_nilai_akhir(daftar_nilai, n)</pre>	<pre>Atun 93.000000 Tejo 85.500000 Udin 88.000000</pre>