# **Tugas Week 1**

## **Soal 1**

Petani memiliki uang 100.000 lalu barang yang ada di point eFishery sebagai berikut ini

| ID | Barang | Harga |
| --- | --- | --- |
| 1 | Benih Lele | 50.000 |
| 2 | Pakan lele cap menara | 25.000 |
| 3 | Probiotik A | 75.000 |
| 4 | Probiotik Nila B | 10.000 |
| 5 | Pakan Nila | 20.000 |
| 6 | Benih Nila | 20.000 |
| 7 | Cupang | 5.000 |
| 8 | Benih Nila | 30.000 |
| 9 | Benih Cupang | 10.000 |
| 10 | Probiotik B | 10.000 |

1. Dengan data berikut ini buatlah sebuah algoritma agar petani dapat membeli barang paling banyak dengan jumlah uang 100.000. Petani hanya boleh membeli 1 item di masing-masing barang.
2. Buat function untuk mencari barang paling murah dan barang paling mahal. ***Tidak boleh menggunakan function sort***
3. Buatlah function untuk menampilkan barang dengan harga 10.000

## **Contoh hasil Soal 1:**

Total produk dengan harga dibawah Rp 100.000 :

Harga total 100000

Cupang - 5000

Probiotik Nila B - 10000

Benih Cupang - 10000

Probiotik B - 10000

Benih Nila 1 - 20000

Pakan Nila - 20000

Pakan lele cap menara - 25000

------------------------------------------------

Daftar produk dengan harga Rp 10.000 :

Probiotik Nila B - 10000

Benih Cupang - 10000

Probiotik B - 10000

------------------------------------------------

Daftar produk termurah: Cupang Rp 5000

Daftar produk termahal: Probiotik A Rp 75000

## **Soal 2**

eFishery memiliki website job search yang dapat di kunjungi di <https://efishery.com/job-search/> . Asumsikan konten yang ada pada halaman website diambil dari 1 API milik eFishery, yang dibelakang layar API tersebut mengambil data dari 2 API yang berbeda, yaitu API job description <https://workspace-rho.vercel.app/api/description> dan API job list <https://workspace-rho.vercel.app/api/jobs>

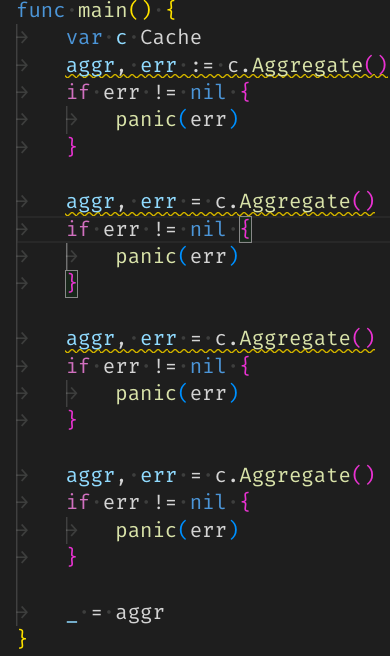
## 

Buatlah sebuah program yang mengambil data dari 2 *fake* API sebelumnya, dan aggregate menjadi output yang ditampilkan pada *console* / *terminal*. Program memiliki spesifikasi:

1. Simpan hasil aggregate pada sebuah *cache* (teknologi atau trik yang digunakan bebas), yang mana hasil aggregate akan kadaluarsa setiap dipakai 1x, sehingga setiap penggunaan yang ke-angka ganjil harus melakukan aggregate ulang, dan seterusnya.

## **Contoh hasil Soal 2:**

Contoh program:



Hasilnya:

dari calculate 2022-09-23 11:15:59.60781893 +0700 WIB m=+0.000310919

took 422.182651ms

dari calculate 2022-09-23 11:16:00.030006964 +0700 WIB m=+0.422498950

took 4.426µs

dari calculate 2022-09-23 11:16:00.030013258 +0700 WIB m=+0.422505242

took 261.648828ms

dari calculate 2022-09-23 11:16:00.291665521 +0700 WIB m=+0.684157503

took 4.675µs

Bisa dilihat pada pada daftar log

1. Pertama (ganjil) waktu yang diperlukan 422ms karena harus melakukan fetch ke server
2. Kedua (genap) 4 micro second, karena memakai nilai dari Cache
3. Ketiga (ganjil) waktu yang diperlukan 261ms karena harus melakukan fetch ke server
4. Ketiga (genap) waktu yang diperlukan 4 micro second karena harus melakukan fetch ke server