

# SOAL PRAKTIKUM OOP (THREAD)

## SOAL NO 1

Intro :

Anda diminta untuk membuat aplikasi yang digunakan untuk meload beberapa data, namun agar prosesnya dapat terjadi secara paralel anda memutuskan menggunakan thread.

Alur :

1. <Skenario> terdapat 4 data dari sumber berbeda yang masing-masing akan di load
2. Buat **Ui Thread** yang bertugas untuk memantau apakah semua proses load data telah selesai.
3. Buat **Background Thread** yang ceritanya bertugas untuk meload data. jadi jumlah Background Threadnya nanti sama dengan jumlah sumber data yang ingin di load.

Kriteria :

1. <Skenario> karena jumlah processor yang tidak memadai, maka anda memutuskan hanya membuat 3 thread pool untuk mengeksekusi Background Thread Sebelumnya.
2. Setiap Background Thread diasumsikan memiliki waktu eksekusi yang berbeda-beda, oleh karena itu buat sebuah class bernama **TaskTimeHelper**

yang memiliki method static yang mana akan mengembalikan angka random dari 1-6. Nah, kembalian dari method inilah yang nantinya akan digunakan sebagai waktu eksekusi masing-masing background thread.

3. <Skenario> Sebagai skenario tambahan saat ada suatu background thread yang waktu eksekusinya melebihi 4 sekon, maka program akan menganggap proses load datanya gagal, dan menampilkan Request Timeout di console.
4. UI thread seakan-akan dikhususkan untuk tampilan. Pada saat program pertama kali jalan, UI Thread akan menampilkan :

**Start load {d} Data.**

***d = jumlah\_data\_sumber***

Kemudian akan dilakukan pengecekan berkelanjutan apakah semua data telah berhasil di-load, jika belum maka akan ditampilkan,

**Loading... ({d}s)**

***d = waktu eksekusi dalam detik***

Namun, jika semua data telah berhasil di-load, maka tampilkan

**Task Finish.**

**Time Execution : {a}s**

**{b} Data Successfully Loaded & {c} Data failed to load.**

**a = waktu yang dibutuhkan untuk meload semua data,**

**b = jumlah data yang berhasil di load**

**c = jumlah data yang gagal di load**

Contoh Output :

Saat semua data berhasil di load

```
Start load 4 Data
Loading... (1s)
Loading... (2s)
Loading... (3s)
Loading... (4s)
Loading... (5s)
Request Timeout
Loading... (6s)

Task Finish.
Time Execution : 6s
3 Data Successfully loaded & 1 Data failed to load
```

Saat ada data yang gagal di load

```
Start load 4 Data
Loading... (1s)
Loading... (2s)
Loading... (3s)
Loading... (4s)
Loading... (5s)

Task Finish.
Time Execution : 5s
All data is successfully loaded
```