

Nama : Lisa Dama Yanti

NPM : 21083010095

❖ Import semua library yang akan digunakan

- *getpid* digunakan untuk mengambil ID proses
- *time* digunakan untuk mengambil waktu(detik)
- *sleep* digunakan untuk memberi jeda waktu(detik)
- *cpu_count* digunakan untuk melihat jumlah CPU
- *Pool* merupakan sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada computer
- *Process* merupakan sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada komputer

```
GNU nano 6.2          tugas8.py
# import library
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process
```

❖ Buatlah fungsi yang dapat menentukan bilangan genap maupun ganjil.

❖ Untuk membuat jeda gunakan sleep

```
# inisialisasi fungsi
def tetapkan(i):
    if i % 2 == 0:
        print(i+1, "Genap - ID proses", getpid())
    else:
        print(i+1, "Ganjil - ID proses", getpid())
    sleep(1)
```

❖ Buatlah perintah yang dapat menerima input dan mencetak output

❖ Variabel a adalah batas perulangan

```
# input
a = int(input("Input:\n"))
print("\nOutput:")
```

❖ Fungsi sekuensial

- Buatlah sebuah perintah untuk mendapatkan waktu sebelum dieksekusi
- Perulangan diproses
- Untuk i di dalam variabel a, eksekusi fungsi lalu tetapkan(i)
- Lalu buatlah perintah untuk mendapatkan waktu setelah dieksekusi.

```
print("\nSekuensial")
sekuensial_awal = time()

for i in range (a):
    tetapkan(i)

sekuensial_akhir = time()
```

[Read 59 lines]

❖ Fungsi multiprocessing.Process

- Buatlah variabel yang dapat menampung sekumpulan proses.
- Buatlah perintah untuk mendapatkan waktu sebelum dieksekusi.
- Ketika proses berlangsung, maka akan terjadi penambahan nilai di dalam variabel yang menampung sekumpulan proses.
- Lalu, gabungkanlah sekumpulan proses tadi agar tidak loncat ke proses sebelumnya.
- Buatlah perintah untuk mendapatkan waktu setelah eksekusi.

```
GNU nano 6.2          tugas8.py
print("\nmultiprocessing.Process")
kumpulan_proses = []

process_awal = time()

for i in range(a):
    p = Process(target=tetapkan, args=(i,))
    kumpulan_proses.append(p)
    p.start()

for i in kumpulan_proses:
    p.join()

process_akhir = time()
```

❖ Fungsi multiprocessing.Pool

- Buatlah perintah untuk mendapatkan waktu sebelum dieksekusi.
- Proses berlangsung.
- Buatlah perintah untuk mendapatkan waktu setelah dieksekusi.

```
print("\nmultiprocessing.Pool")
pool_awal = time()

pool = Pool()
pool.map(tetapkan, range(a))
pool.close

pool_akhir = time()
```

❖ Perbandingan kecepatan waktu dari 3 metode

- menghitung selisih waktu setelah proses dan sebelum proses

```
#perbandingan waktu
print("\nWaktu eksekusi sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Waktu eksekusi multiprocessing.Process:", process_akhir - process_awal, "detik")
print("Waktu eksekusi multiprocessing.Pool:", pool_akhir - pool_awal, "detik")
```

❖ output

```
Waktu eksekusi multiprocessing.Pool: 3.050112670063818 detik
lisa@lisa-VirtualBox:~$ nano tugas8.py
lisa@lisa-VirtualBox:~$ python3 tugas8.py
Input:
3

Output:

Sekuensial
1 Genap - ID proses 6707
2 Ganjil - ID proses 6707
3 Genap - ID proses 6707

multiprocessing.Process
2 Ganjil - ID proses 6710
3 Genap - ID proses 6711
1 Genap - ID proses 6709

multiprocessing.Pool
1 Genap - ID proses 6712
2 Ganjil - ID proses 6712
3 Genap - ID proses 6712

Waktu eksekusi sekuensial : 3.00333833694458 detik
Waktu eksekusi multiprocessing.Process: 1.026702642440796 detik
Waktu eksekusi multiprocessing.Pool: 3.0517523288726807 detik
lisa@lisa-VirtualBox:~$
```