

Nama: Yayang Dimas Saputra

NPM: 21083010102

Kelas: Sistem Operasi B

## Laporan Tugas 8

Soal latihan :

Dengan menggunakan pemrosesan paralel buatlah program yang dapat menentukan sebuah bilangan itu ganjil atau genap!

Jawaban:

### Script

```
GNU nano 6.2                                     Tugas 8.py
from os import getpid
from time import time, sleep
from multiprocessing import Pool, Process

def cetak(i):
    bil = i % 2
    if bil == 0:
        print(i, "Genap - ID proses", getpid())
    else:
        print(i, "Ganjil - ID proses", getpid())
        sleep(1)

n = int(input("Angka batasan: "))

# Sekuensial
print("\nSekuensial")
sekuensial_awal = time()
for i in range(1, n + 1):
    cetak(i)
sekuensial_akhir = time()

# Kelas Process
print("\nmultiprocessing.Process")
kumpulan_proses = []
process_awal = time()
for i in range(1, n + 1):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()

for i in kumpulan_proses:
    p.join()
process_akhir = time()

# Kelas Pool
print("\nmultiprocessing.Pool")
pool_awal = time()
pool = Pool()
pool.map(cetak, range(1, n + 1))
pool.close()
pool_akhir = time()

# Perbandingan waktu eksekusi
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")
```

## Penjelasan

```
from os import getpid
from time import time, sleep
from multiprocessing import Pool, Process
```

- Import built-in libraries yang akan digunakan

### 1. getpid

Merupakan sebuah function yang berfungsi untuk mendapatkan ID proses dari proses saat ini.

### 2. time

Merupakan sebuah function yang berfungsi untuk mengambil waktu dalam satuan detik.

### 3. sleep

Merupakan sebuah function yang berfungsi untuk menangguhkan eksekusi perintah dalam jumlah waktu (detik) yang diberikan.

### 4. Process

Merupakan sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada komputer

### 5. Pool

Merupakan sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU yang terdapat pada komputer.

```
def cetak(i):
    bil = i % 2
    if bil == 0:
        print(i, "Genap - ID proses", getpid())
    else:
        print(i, "Ganjil - ID proses", getpid())
        sleep(1)

n = int(input("Angka batasan: "))
```

- Buat fungsi ber inisial 'cetak' yang digunakan untuk menandai apakah angka yang diproses ganjil atau genap. Dengan proses bilangan  $i \% 2$ , lalu jika hasilnya 0(habis) maka angka tersebut genap dan jika menghasilkan angka selain 0, maka angka tersebut berarti ganjil.

- Lalu user diminta meng inputkan angka bulat digunakan sebagai angka batasan
- Lalu script melakukan processing Sekuensial, multiprocessing.Process, dan multiprocessing.Pool

## Output

```
yayang@yayang-VirtualBox:~$ python3 Tugas_8.py
Angka batasan: 4

Sekuensial
1 Ganjil - ID proses 73160
2 Genap - ID proses 73160
3 Ganjil - ID proses 73160
4 Genap - ID proses 73160

multiprocessing.Process
1 Ganjil - ID proses 73161
2 Genap - ID proses 73162
4 Genap - ID proses 73164
3 Ganjil - ID proses 73163

multiprocessing.Pool
1 Ganjil - ID proses 73165
2 Genap - ID proses 73166
3 Ganjil - ID proses 73166
4 Genap - ID proses 73165

waktu eksekusi sekuensial           : 4.064723491668701 detik
waktu eksekusi multiprocessing.Process : 1.0162863731384277 detik
waktu eksekusi multiprocessing.Pool   : 2.0372729301452637 detik
```

- Disini user meng isi kan angka 4 sebagai angka batasan
- Pada sekuensial, terlihat bahwa setiap ID processnya sama. ini menandakan proses sekuensial mengeksekusi pada pemroses yang sama(satu proses yang sama).
- Pada multiprocessing dengan kelas process, terlihat bahwa setiap ID processnya itu berbeda dan beruntun. Hal ini menunjukkan bahwa tiap pemanggilan fungsi cetak ditangani oleh setiap satu proses saja
- Pada multiprocessing dengan kelas pool, terlihat ada 2 ID process yang berbeda dan berulang di 2 ID process itu saja dan tidak apa tidak urut,. 2 ID process saja ini terjadi karena laptop yang digunakan untuk menjalankan script hanya

memiliki 2 CPU maka dipetakan menjadi 3 pemrosesan di tiap CPU yang berbeda.

- Dapat dilihat bahwa multiprocessing dengan kelas Process memiliki durasi waktu yang paling singkat dibandingkan dengan proses lainnya dan pemrosesan sekuensial menjadi proses yang memiliki durasi waktu terlama