

Nama : Edina Alana Nabila  
NPM : 21083010022  
Kelas : Sistem Operasi A

## TUGAS 8

1. Pertama-tama kita buat dulu file pythonnya dengan command nano. Jangan lupa karena ini membuat file python, maka type filenya adalah .py

```
edina@edina-VirtualBox:~/Tugas8$ nano Tugas_8.py
```

2. Kita tuliskan script yang pertama yaitu import library

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

3. Definisikan variabel x sebagai angka yang nantinya akan diinput sebagai angka batasan

```
#Input angka batasan yang diinginkan
x = int(input("Masukkan angka batasan: "))
```

4. Selanjutnya kita menginisiasi fungsi yang akan digunakan dengan fungsi bernama cetak

```
#Inisiasi fungsi yang akan digunakan
def cetak(i):
    print("Cetak angka", i+1,"- punya ID proses", getpid())
    sleep(1)
```

5. Masuk ke proses sekuensial kita memasukkan script seperti berikut.

```
#Pemrosesan sekuensial
print("Sekuenisial")
sekuensial_awal = time()

for i in range(x):
    cetak(i)

sekuensial_akhir = time()
```

Pada bagian untuk i pada range(x), x nya disini adalah angka batasan sembarang yang akan kita inputkan, lalu akan dicetak i hingga batasan tersebut.

6. Multiprocessing dengan kelas process kita menuliskan script berikut.

```
#Multiprocessing dengan kelas Process
print("multiprocessing.Process")
kumpulan_proses = []

process_awal = time()

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

for i in kumpulan_proses:
    p.join()

process_akhir = time()
```

Yang masih sama adalah batasannya yaitu angka x.

Pada proses ini, Kumpulan proses harus ditampung dan digabung menjadi satu(p.join())agar tidak merambah ke proses selanjutnya.

7. Lalu multiprocessing dengan kelas Pool kita menuliskan script berikut.

```
#Multiprocessing dengan kelas Pool
print("multiprocessing.Pool")
pool_awal = time()

pool = Pool()
pool.map(cetak, range(0,x))
pool.close()

pool_akhir = time()
```

Fungsi map() itu memetakan pemanggilan fungsi cetak ke dalam 4 CPU sebanyak 10 kali.

8. Terakhir, kita tuliskan script berikut agar kita dapat membandingkan waktu eksekusi.

```
#Bandingkan waktu eksekusi
print("Waktu 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")
```

9. Simpan file python dengan CTRL-X lalu CTRL-Y
10. Run atau jalankan file tersebut dengan comman python3 Tugas\_8.py  
Maka output yang dikeluarkan akan seperti ini

```

edina@edina-VirtualBox:~/Tugas8$ python3 Tugas_8.py
Masukkan angka batasan: 3
Sekuensial
Cetak angka 1 - punya ID proses 3608
Cetak angka 2 - punya ID proses 3608
Cetak angka 3 - punya ID proses 3608
multiprocessing.Process
Cetak angka 1 - punya ID proses 3609
Cetak angka 2 - punya ID proses 3610
Cetak angka 3 - punya ID proses 3611
multiprocessing.Pool
Cetak angka 1 - punya ID proses 3612
Cetak angka 2 - punya ID proses 3612
Cetak angka 3 - punya ID proses 3612
Waktu eksekusi sekuensial : 3.003962278366089 detik
Waktu eksekusi multiprocessing.Process : 1.15541410446167 detik
Waktu eksekusi multiprocessing.Pool : 3.2430789470672607 detik
edina@edina-VirtualBox:~/Tugas8$ █

```

- Angka batasan yang kita masukkan disini adalah 3
- Dapat diperhatikan dengan seksama bahwa ID proses tiap memanggil fungsi cetak adalah berbeda-beda. Ini menandakan bahwa tiap pemanggilan fungsi cetak ditangani oleh satu proses saja. Kemudian untuk pemanggilan selanjutnya ditangani oleh proses yang lain.
- Jumlah ID proses yang tercetak nantinya terbatas pada empat saja karena jumlah CPU pada komputer saya hanyalah 4.
- Dan juga dapat dilihat bahwa proses sekuensial lebih lambat dibanding multiprocessing. Namun bukan berarti kita harus melakukan multiprocessing terus menerus, kita tetap harus gunakan metode sesuai kebutuhan.