

Nama : Gema Khusnul Ma'rifah  
NPM : 21083010034

## TUGAS 8

### MULTIPROCESSING

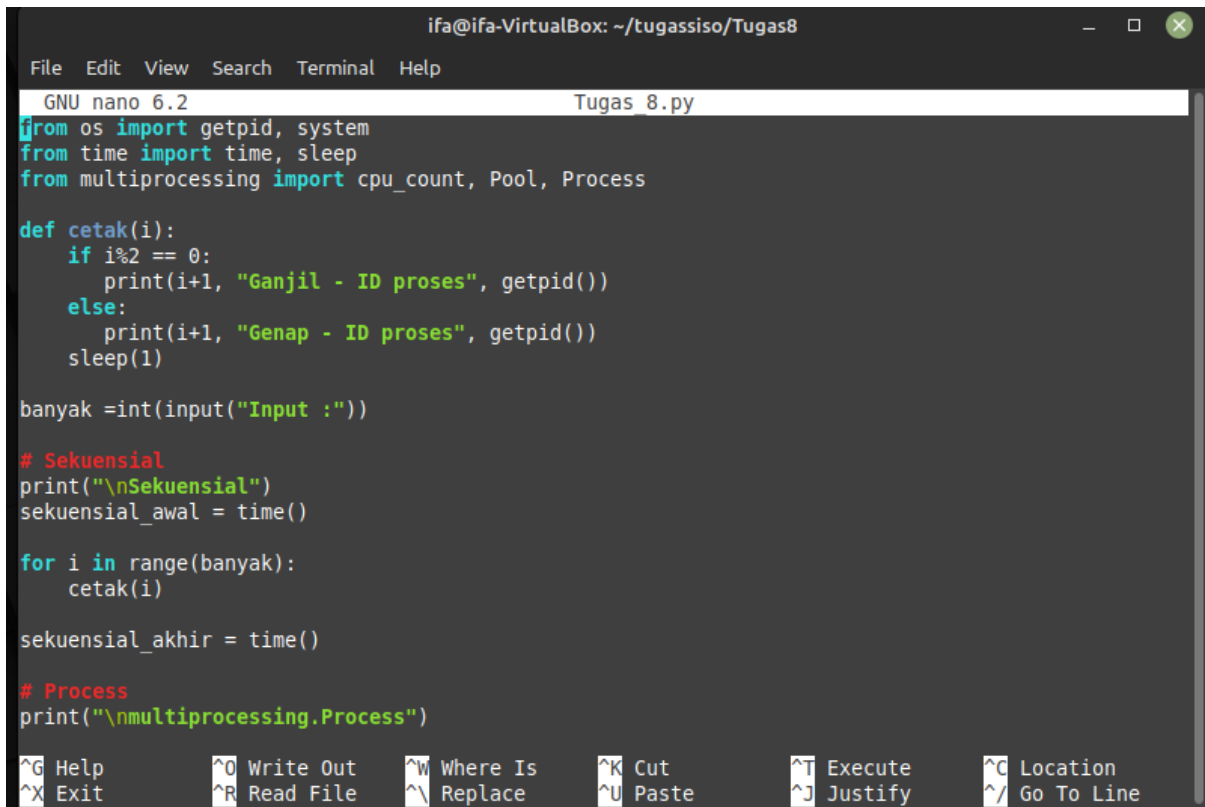
#### Latihan Soal:

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

Batasan:

- Nilai yang dijadikan argumen pada fungsi sleep() adalah satu detik.
- Masukkan jumlahnya satu dan berupa bilangan bulat.
- Masukkan adalah batas dari perulangan tersebut.
- Setelah perulangan selesai program menampilkan waktu eksekusi pemrosesan sekuensial dan paralel.

#### Penyelesaian:



```
ifafa@ifa-VirtualBox: ~/tugassiso/Tugas8
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py
from os import getpid, system
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

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

banyak =int(input("Input :"))

# Sekuensial
print("\nSekuensial")
sekuensial_awal = time()

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

sekuensial_akhir = time()

# Process
print("\nmultiprocessing.Process")

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

#### Script:

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

def cetak(i):
```

Nama : Gema Khusnul Ma'rifah

NPM : 21083010034

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

banyak = int(input("Input :"))

# Sekuensial
print("\nSekuensial")
sekuensial_awal = time()

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

sekuensial_akhir = time()

# Process
print("\nmultiprocessing.Process")
kumpulan_proses = []

process_awal = time()

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

for i in kumpulan_proses:
    p.join()

process_akhir = time()

# Pool
print("\nmultiprocessing.Pool")

pool_awal = time()

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

pool_akhir = time()
```

Nama : Gema Khusnul Ma'rifah

NPM : 21083010034

```
# Banding 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 Script:

- Memuat Built-in Libraries

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

Pertama-tama yang perlu dilakukan adalah meng-*import built-in libraries* yang akan digunakan yaitu “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” adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan parallel dengan menggunakan proses sebanyak jumlah CPU pada komputer, dan “Process” adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan parallel dengan menggunakan proses secara beruntun pada komputer.

- Inisialisasi Fungsi yang akan Digunakan

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

Selanjutnya yaitu melakukan inisialisasi sebuah fungsi yang akan digunakan yaitu fungsi “cetak” yang akan digunakan untuk mencetak angka dari variabel i beserta ID proses sejumlah parameter yang diberikan. Pada fungsi “cetak” digunakan percabangan if-else untuk menjadikan variabel i yang terpanggil terbagi menjadi ganjil serta genap dengan ketentuan “i%2 == 0” untuk merujuk pada variabel ganjil dan yang lainnya merujuk pada variabel genap. Selain itu untuk fungsi sleep digunakan untuk memberi jeda waktu(detik) sebanyak parameter yang diberikan.

- Membuat Variabel untuk Banyaknya Angka yang akan Muncul

```
banyak =int(input("Input :"))
```

Nama : Gema Khusnul Ma'rifah

NPM : 21083010034

#### - Pemrosesan Sekuensial

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

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

sekuensial_akhir = time()
```

Menggunakan perulangan for dengan range disesuaikan pada inputan yang telah dilakukan kemudian menjalankan fungsi “cetak” yang telah diinisialisasikan sebelumnya.

#### - Multiprocessing dengan Kelas Process

```
# Process
print("\nmultiprocessing.Process")
kumpulan_proses = []

process_awal = time()

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

for i in kumpulan_proses:
    p.join()

process_akhir = time()
```

Menggunakan perulangan for dengan range disesuaikan pada inputan yang telah dilakukan kemudian menjalankan fungsi “Process” yang telah di-import sebelumnya dengan berdasarkan kepada fungsi “cetak”, dengan menambahkannya kedalam list “kumpulan\_proses” yang telah diinisialisasikan diawal. Terdapat perulangan for untuk kumpulan\_proses agar proses yang berjalan tidak bergabung dengan proses sebelumnya.

#### - Multiprocessing dengan Kelas Pool

```
# Pool
print("\nmultiprocessing.Pool")

pool_awal = time()

pool = Pool()
pool.map(cetak, range(0, banyak))
```

Nama : Gema Khusnul Ma'rifah

NPM : 21083010034

```
pool.close()

pool_akhir = time()
```

Menggunakan fungsi map untuk memetakan pemanggilan fungsi cetak ke dalam CPU sebanyak angka yang telah diinputkan sebelumnya.

- Membandingkan Waktu Eksekusi

```
# Banding 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")
```

Terakhir yaitu dilakukan perbandingan terhadap lamanya waktu eksekusi pada masing-masing proses dalam detik. Masing-masing eksekusi dilakukan perhitungan dengan waktu akhir dikurangi dengan waktu awal sehingga didapatkan output sesuai dengan eksekusinya masing-masing.

- print("\nWaktu eksekusi Sekuensial :", sekuensial\_akhir - sekuensial\_awal, "detik") untuk melihat selang waktu pada eksekusi sekuensial.
- print("Waktu eksekusi multiprocessing.Process :", process\_akhir - process\_awal, "detik") untuk melihat selang waktu pada eksekusi Multiprocessing Process.
- print("Waktu eksekusi multiprocessing.Pool :", pool\_akhir - pool\_awal, "detik") untuk melihat selang waktu pada eksekusi Multiprocessing Pool.

Nama : Gema Khusnul Ma'rifah

NPM : 21083010034

### Output:

```
ifa@ifa-VirtualBox: ~/tugassiso/Tugas8
File Edit View Search Terminal Help
ifa@ifa-VirtualBox:~$ cd tugassiso/Tugas8
ifa@ifa-VirtualBox:~/tugassiso/Tugas8$ nano Tugas_8.py
ifa@ifa-VirtualBox:~/tugassiso/Tugas8$ python3 Tugas_8.py
Input :3

Sekuensial
1 Ganjil - ID proses 1850
2 Genap - ID proses 1850
3 Ganjil - ID proses 1850

multiprocessing.Process
3 Ganjil - ID proses 1853
2 Genap - ID proses 1852
1 Ganjil - ID proses 1851

multiprocessing.Pool
1 Ganjil - ID proses 1854
2 Genap - ID proses 1854
3 Ganjil - ID proses 1854

Waktu eksekusi Sekuensial : 3.003361940383911 detik
Waktu eksekusi multiprocessing.Process : 1.0073673725128174 detik
Waktu eksekusi multiprocessing.Pool : 3.0151915550231934 detik
```