

**LAPORAN
MATA KULIAH SISTEM OPERASI
KELAS B**



“Multiprocessing”

DISUSUN OLEH :

1. RAJA VALENTINO KRISTANANDA (21083010068)

DOSEN PENGAMPU:

MOHAMMAD IDHOM, SP., S.Kom., M.Kom

**PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL
“VETERAN” JAWA TIMUR
2021**

Laporan Praktikum Multiprocessing

1. Soal Latihan Multiprocessing

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

Batasan:

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

Memulai dengan membuat file dengan menggunakan nama "nano Tugas_8.py"

```
valentino@Valentino: ~/TugasSisop7
File Edit View Search Terminal Help
valentino@Valentino:~/TugasSisop7$ nano Tugas_8.py
valentino@Valentino:~/TugasSisop7$
```

Menuliskan script sesuai pada Code Material pada google colab

```
valentino@Valentino: ~/TugasSisop7
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py
from os import getpid
from time import time,sleep
from multiprocessing import cpu_count, Pool, Process

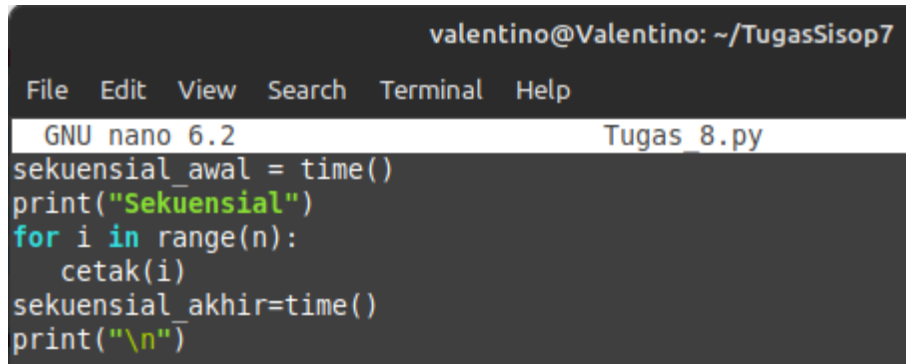
def cetak(i):
    if (i+1)%2==0:
        print(i+1, "genap - ID Process", getpid())
    else:
        print(i+1, "ganjil - ID Process", getpid())
        sleep(1)

n=int(input("Nilai angka batasan? "))
print("\n")
```

Lakukan import module yang diperlukan dalam melakukan multiprocessing sesuai yang diminta pada soal.

- i. getpid digunakan untuk mengambil ID proses
- ii. time digunakan untuk mengambil waktu(dektik)
- iii. cpu_count digunakan untuk melihat jumlah CPU
- iv. Pool adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada komputer
- v. Process adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada komputer

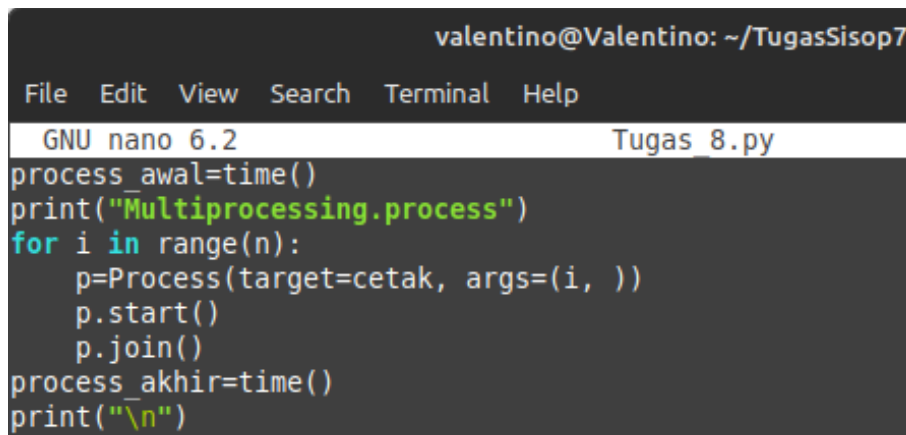
Dilanjutkan dengan menuliskan fungsi cetak(i): digunakan untuk mencetak angka variable I beserta ID proses sejumlah parameter angka yang dimasukan. Jika hasil parameter ditambahkan 1 dan dimodulo 2 dan hasilnya samadengan 0, maka masuk ke kondisi 1, jika tidak maka masuk ke kondisi 2. Selanjutnya memanggil fungsi sleep untuk memberikan jeda tiap detik setiap parameter yang diberikan. Dan melakukan inisiasi penginputan variable n untuk menuliskan nilai batasan pada script.



```
valentino@Valentino: ~/TugasSisop7
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py
sekuensial_awal = time()
print("Sekuensial")
for i in range(n):
    cetak(i)
sekuensial_akhir=time()
print("\n")
```

Proses selanjutnya adalah melakukan proses sekuensial

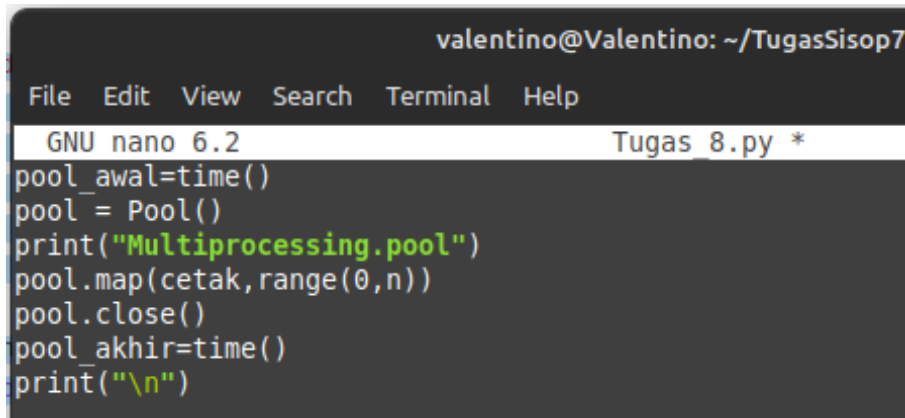
- i. variable sekuensial_awal digunakan untuk mendapatkan waktu durasi sebelum proses sekuensial berlangsung
- ii. proses dijalankan menggunakan looping for sebanyak angka yang dimasukan, dan memanggil fungsi cetak yang telah diisi untuk mencetak setiap angka ganjil atau genap dengan id prosesnya masing-masing
- iii. variable sekuensial_akhir digunakan untuk mendapatkan waktu durasi setelah proses sekuensial berlangsung



```
valentino@Valentino: ~/TugasSisop7
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py
process_awal=time()
print("Multiprocessing.process")
for i in range(n):
    p=Process(target=cetak, args=(i, ))
    p.start()
    p.join()
process_akhir=time()
print("\n")
```

Setelah proses sekuensial, dilanjutkan dengan proses multiprocessing pada tahap kelas process

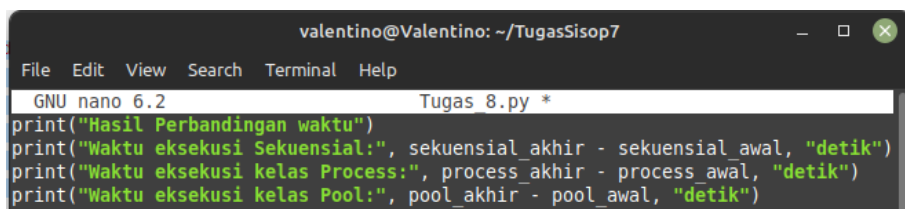
- i. variable process_awal digunakan untuk mendapatkan waktu awal mulainya proses dijalankan
- ii. proses dijalankan menggunakan looping for sebanyak angka yang dimasukkan dan menggunakan fungsi cetak yang sudah kita isi di awal untuk mencetak setiap angka ganjil atau genap dengan id proses masing – masing
- iii. p.start() digunakan untuk mengeksekusi fungsi cetak di kelas process
- iv. p.join() digunakan agar proses ditunggu hingga proses sebelumnya selesai. Sehingga akan menghasilkan id proses yang berbeda – beda tiap prosesnya.
- v. variable process_akhir digunakan untuk mendapatkan waktu berakhirnya proses dijalankan



```
valentino@Valentino: ~/TugasSisop7
File Edit View Search Terminal Help
GNU nano 6.2 Tugas 8.py *
pool_awal=time()
pool = Pool()
print("Multiprocessing.pool")
pool.map(cetak,range(0,n))
pool.close()
pool_akhir=time()
print("\n")
```

Setelah tahap multiprocessing kelas process dilanjutkan dengan multiprocessing tahap kelas pool

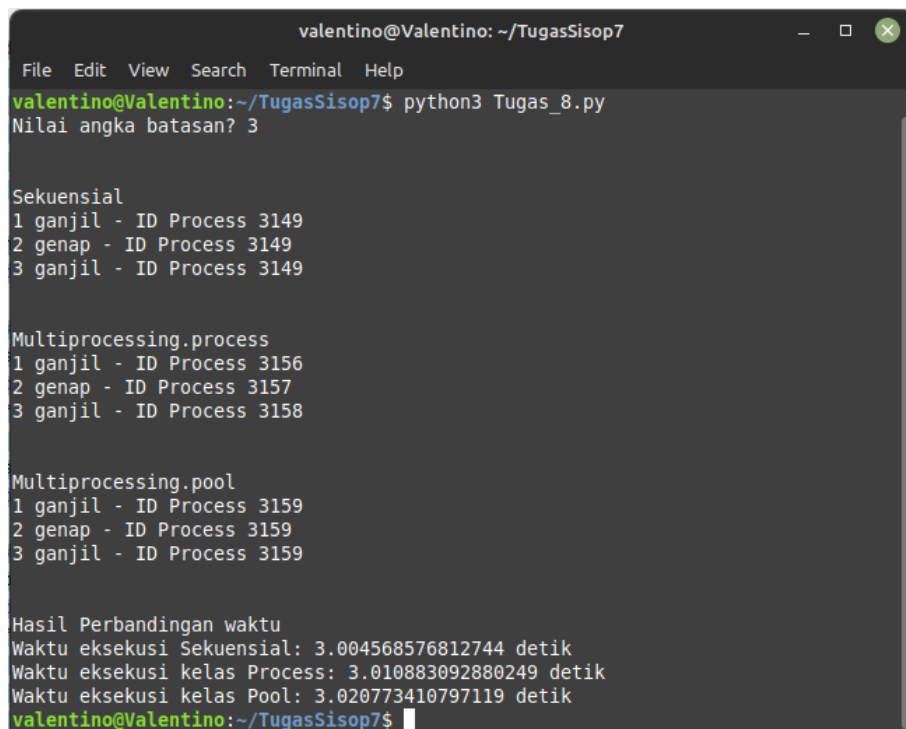
- i. variable pool_awal digunakan untuk mendapatkan waktu awal mulainya proses dijalankan
- ii. variable pool digunakan untuk menjalankan fungsi Pool
- iii. meninisiasi pool.map dengan fungsi map() digunakan untuk memetakan pemanggilan fungsi cetak ke dalam setiap CPU yang tersedia sebanyak 0-n kali yang mana 'n' adalah inputan batasan dari user.
- iv. variable pool_akhir digunakan untuk mendapatkan waktu berakhirnya proses dijalankan



```
valentino@Valentino: ~/TugasSisop7
File Edit View Search Terminal Help
GNU nano 6.2 Tugas 8.py *
print("Hasil Perbandingan waktu")
print("Waktu eksekusi Sekuensial:", sekuensial_akhir - sekuensial_awal, "detik")
print("Waktu eksekusi kelas Process:", process_akhir - process_awal, "detik")
print("Waktu eksekusi kelas Pool:", pool_akhir - pool_awal, "detik")
```

Proses terakhir yaitu dengan melakukan pemanggilan seluruh proses yang dilakukan dengan menggunakan fungsi print. Dengan ketentuan waktu eksekusi sekuensial dilakukan dengan memanggil variable sekuensial_akhir dikurangi sekuensial_awal, waktu eksekusi kelas process dilakukan dengan memanggil variable process_akhir dikurangi process_awal, dan waktu eksekusi kelas pool dilakukan dengan memanggil variable pool_akhir dikurangi pool_awal.

Hasil Output yang dikeluarkan :



```
valentino@Valentino: ~/TugasSisop7
File Edit View Search Terminal Help
valentino@Valentino:~/TugasSisop7$ python3 Tugas_8.py
Nilai angka batasan? 3

Sekuensial
1 ganjil - ID Process 3149
2 genap - ID Process 3149
3 ganjil - ID Process 3149

Multiprocessing.process
1 ganjil - ID Process 3156
2 genap - ID Process 3157
3 ganjil - ID Process 3158

Multiprocessing.pool
1 ganjil - ID Process 3159
2 genap - ID Process 3159
3 ganjil - ID Process 3159

Hasil Perbandingan waktu
Waktu eksekusi Sekuensial: 3.004568576812744 detik
Waktu eksekusi kelas Process: 3.010883092880249 detik
Waktu eksekusi kelas Pool: 3.020773410797119 detik
valentino@Valentino:~/TugasSisop7$
```

Memanggil berkas script yang telah dibuat dengan menggunakan python3 Tugas_8.py dan menginputkan angka 3 sebagai batasan yang diminta pada soal latihan.

- i. Pada tahap sekuensial memprintout id proses pada masing2 bilangan yang di inputkan
- ii. Pada tahap multiprocessing kelas process memprintout id process, masing2 bilangan dengan id yang berbeda dikarenakan tiap pemanggilan fungsi cetak dilakukan oleh satu proses saja
- iii. Pada tahap multiprocessing kelas pool memprintout id process pada masing2 bilangan

Memprintout seluruh waktu yang dilakukan pada tahap multiprocessing, dapat kita lihat pada proses eksekusi kelas process adalah proses yang memakan waktu cukup lama dibandingkan dengan yang lain karena pada saat melakukan proses, kelas process melakukan pemanggilan tiap fungsi cetak yang dilakukan oleh satu proses saja.