

Nama : Denisa Septalian Alhamda

NPM : 21083010113

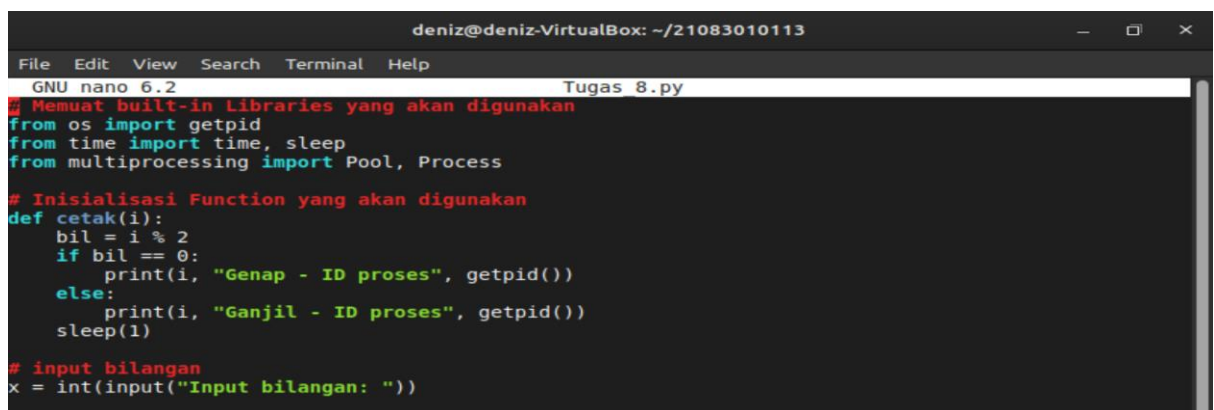
Kelas : Sistem Oprasi B

## Multiprocessing

Latihan soal

➤ Script

Pada script ini berisi program yang berfungsi untuk mencetak bilangan bulat positif beserta tipe (ganjil/genap) berdasarkan batasan bilangan yang ditentukan oleh user dengan pemrosesan sekuensial, multiprocessing dengan kelas Process, dan multiprocessing dengan kelas Pool.



```
deniz@deniz-VirtualBox: ~/21083010113
File Edit View Search Terminal Help
GNU nano 6.2 Tugas 8.py
# Memuat built-in Libraries yang akan digunakan
from os import getpid
from time import time, sleep
from multiprocessing import Pool, Process

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

# input bilangan
x = int(input("Input bilangan: "))
```

A. Memuat built-in libraries yang akan digunakan

1. getpid

Merupakan sebuah function yang mengembalikan (return) ID proses yang sedang berjalan.

2. time

Merupakan sebuah function yang berfungsi untuk mengambil waktu (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 computer.

B. Inisialisasi function yang akan digunakan

1.  $bil = i \% 2$

Dilakukan perhitungan modulus 2 (...%2) terhadap nilai variabel i untuk memeriksa bilangan merupakan bilangan genap atau ganjil. Hasil perhitungan disimpan oleh variabel bil.

2.  $if\ bil == 0$

Apabila hasil perhitungan modulus adalah 0, maka bilangan (i) merupakan bilangan genap. Dicitak nilai variabel i, kalimat "Genap – ID Proses", dan ID proses yang didapatkan melalui penggunaan function getpid().

3.  $if\ bil == 0$

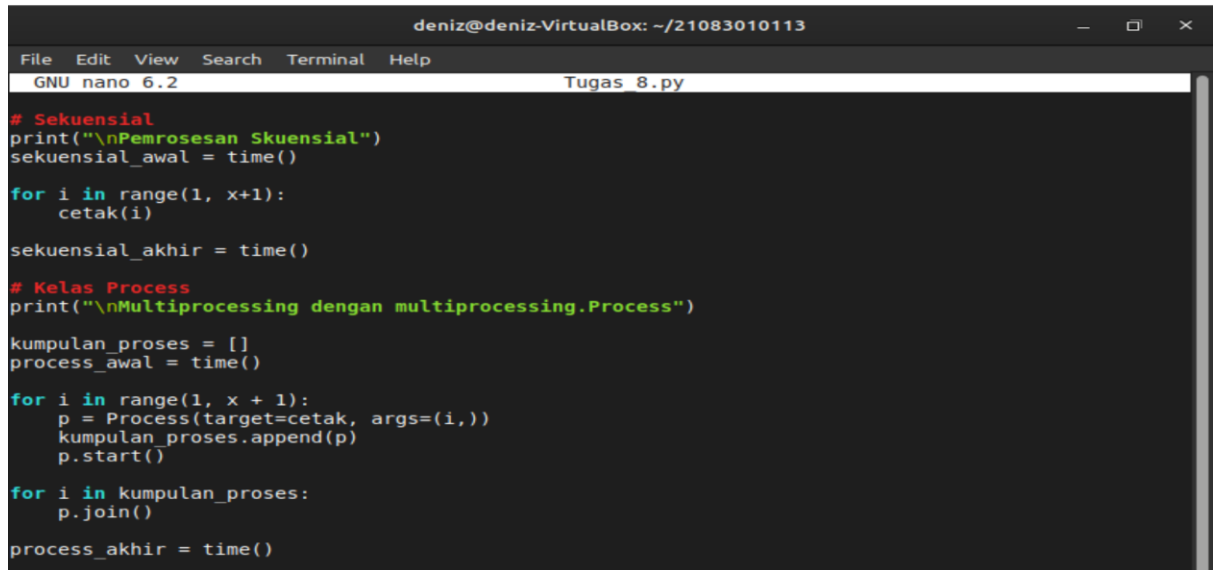
Apabila hasil perhitungan modulus adalah 0, maka bilangan (i) merupakan bilangan genap. Dicitak nilai variabel i, kalimat "Genap – ID Proses", dan ID proses yang didapatkan melalui penggunaan function getpid().

4.  $if\ bil == 0$

Apabila hasil perhitungan modulus adalah 0, maka bilangan (i) merupakan bilangan genap. Dicitak nilai variabel i, kalimat “Genap – ID Proses”, dan ID proses yang didapatkan melalui penggunaan function getpid().

### C. Input bilangan

Dibuat sebuah field input dengan batasan tipe data integer untuk user dengan teks “Input bilangan: ” yang disimpan dalam variabel x. Tujuan dari field input ini adalah untuk menginputkan bilangan yang digunakan sebagai batasan pada program.



```
deniz@deniz-VirtualBox: ~/21083010113
File Edit View Search Terminal Help
GNU nano 6.2 Tugas 8.py
# Sekuensial
print("\nPemrosesan Skuensial")
sekuensial_awal = time()

for i in range(1, x+1):
    cetak(i)

sekuensial_akhir = time()

# Kelas Process
print("\nMultiprocessing dengan multiprocessing.Process")

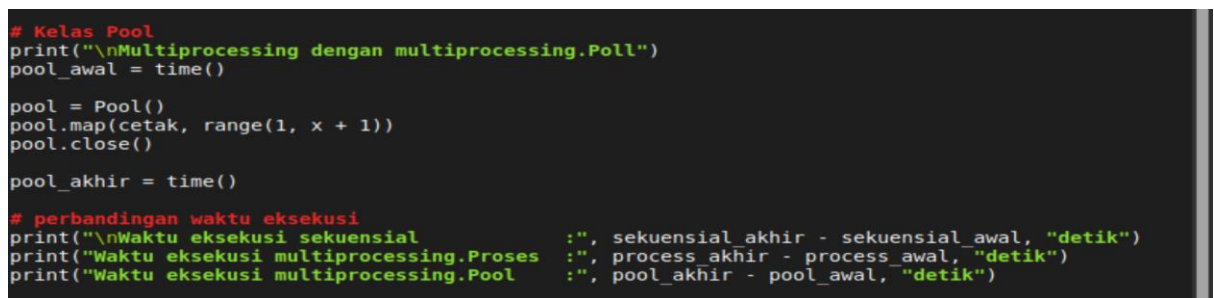
kumpulan_proses = []
process_awal = time()

for i in range(1, x + 1):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()

for i in kumpulan_proses:
    p.join()

process_akhir = time()
```

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 selanjut'nya ditangani oleh proses yang lain. Kumpulan proses harus ditampung dan digabung menjadi satu(p.join()) agar tidak merambah ke proses selanjutnya.



```
# Kelas Pool
print("\nMultiprocessing dengan multiprocessing.Pool")
pool_awal = time()

pool = Pool()
pool.map(cetak, range(1, x + 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")
```

Jumlah ID proses terbatas pada empat saja karena jumlah CPU pada komputer saya hanyalah 4. Jangan risaukan urutan angka yang dicetak jika tidak berurutan, kan emang ini pemrosesan paralel. Fungsi map() itu memetakan pemanggilan fungsi cetak ke dalam 4 CPU sebanyak 10 kali.

➤ Output

```
deniz@deniz-VirtualBox: ~/21083010113
File Edit View Search Terminal Help
deniz@deniz-VirtualBox:~/21083010113$ nano Tugas_8.py
deniz@deniz-VirtualBox:~/21083010113$ python3 Tugas_8.py
Input bilangan: 11

Pemrosesan Skuensial
1 Ganjil - ID proses 2337
2 Genap - ID proses 2337
3 Ganjil - ID proses 2337
4 Genap - ID proses 2337
5 Ganjil - ID proses 2337
6 Genap - ID proses 2337
7 Ganjil - ID proses 2337
8 Genap - ID proses 2337
9 Ganjil - ID proses 2337
10 Genap - ID proses 2337
11 Ganjil - ID proses 2337

Multiprocessing dengan multiprocessing.Process
2 Genap - ID proses 2339
7 Ganjil - ID proses 2344
3 Ganjil - ID proses 2340
1 Ganjil - ID proses 2338
4 Genap - ID proses 2341
5 Ganjil - ID proses 2342
6 Genap - ID proses 2343
9 Ganjil - ID proses 2346
8 Genap - ID proses 2345
11 Ganjil - ID proses 2348
10 Genap - ID proses 2347

Multiprocessing dengan multiprocessing.Pool
1 Ganjil - ID proses 2485
2 Genap - ID proses 2485
3 Ganjil - ID proses 2485
4 Genap - ID proses 2485
5 Ganjil - ID proses 2485
6 Genap - ID proses 2485
7 Ganjil - ID proses 2485
8 Genap - ID proses 2485
9 Ganjil - ID proses 2485
10 Genap - ID proses 2485
11 Ganjil - ID proses 2485

Waktu eksekusi sekuensial : 11.011492013931274 detik
Waktu eksekusi multiprocessing.Proces : 1.01924729347229 detik
Waktu eksekusi multiprocessing.Pool : 11.024683475494385 detik
```