

Laporan Tugas 8

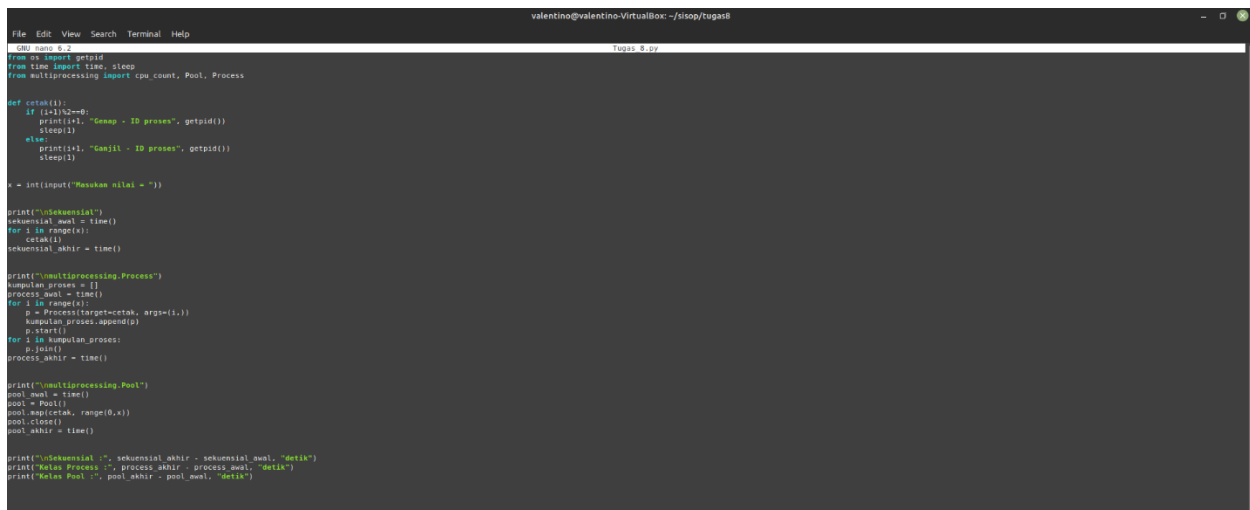
Soal latihan :

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

Batasan:

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

Script :



```
valentino@valentino-VirtualBox: ~/sisop/tugas8
File Edit View Search Terminal Help
Tugas 8.py
GNU nano 6.2
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

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

x = int(input("Masukan nilai = "))

print("\nsekuensial")
sekuensial_awal = time()
for i in range(x):
    cetak(i)
sekuensial_akhir = time()

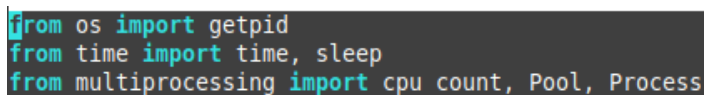
print("\nmultiprocessing.Process")
kumpulan_proses = []
proses_awal = time()
for i in range(x):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()
for i in kumpulan_proses:
    p.join()
proses_akhir = time()

print("\nmultiprocessing.Pool")
pool_awal = time()
pool = Pool()
pool.map(cetak, range(0,x))
pool.close()
pool_akhir = time()

print("\nsekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Kelas Process :", proses_akhir - proses_awal, "detik")
print("Kelas Pool :", pool_akhir - pool_awal, "detik")
```

Penjelasan script:

- Muat built-in libraries yang akan digunakan :



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

Memuat library yang dibutuhkan seperti:

- a. getpid digunakan untuk mengambil ID proses
- b. time digunakan untuk mengambil waktu(detik)
- c. sleep digunakan untuk memberi jeda waktu(detik)

- d. `cpu_count` digunakan untuk melihat jumlah CPU
 - e. Pool adalah sebuah class pada library `multiprocessing` yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada komputer
 - f. Process adalah sebuah class pada library `multiprocessing` yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada komputer
- Inisialisasi fungsi yang akan digunakan

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

Fungsi `def cetak(i)` digunakan untuk mencetak angka dari variabel `i` beserta ID proses sejumlah parameter yang diberikan. Kita panggil fungsi `sleep` untuk memberi jeda waktu selama 1 detik.

- membuat variabel untuk memunculkan banyaknya nilai

```
x = int(input("Masukan nilai = "))
```

- Pemrosesan sekuensial

```
print("\nSekuensial")
sekuensial_awal = time()
for i in range(x):
    cetak(i)
sekuensial_akhir = time()
```

Fungsi `time` pada `sekuensial_awal` dan `sekuensial_akhir` untuk mendapat waktu sesudah dan sebelum eksekusi. Penggunaan `for i in range` disesuaikan dengan input yang diberikan

- Multiprocessing dengan kelas Process

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

Kumpulan proses harus ditampung dan digabung menjadi satu (`p.join()`) agar tidak merambah ke proses selanjutnya. Silahkan eksekusi file berikut pada terminal anda, maka anda akan paham apa yang saya maksudkan.

- Multiprocessing dengan kelas Pool

```
print("\nmultiprocessing.Pool")
pool_awal = time()
pool = Pool()
pool.map(cetak, range(0,x))
pool.close()
pool_akhir = time()
```

- Bandingkan Waktu Eksekusi

```
print("\nSekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Kelas Process :", process_akhir - process_awal, "detik")
print("Kelas Pool :", pool_akhir - pool_awal, "detik")
```

Output :

```
File Edit View Search Terminal Help
valentino@valentino-VirtualBox:~/sisop/tugas8
valentino@valentino-VirtualBox:~/sisop/tugas8$ nano Tugas_8.py
valentino@valentino-VirtualBox:~/sisop/tugas8$ python3 Tugas_8.py
Masukan nilai = 3

Sekuensial
1 Ganjil - ID proses 1848
2 Genap - ID proses 1848
3 Ganjil - ID proses 1848

multiprocessing.Process
1 Ganjil - ID proses 1849
2 Genap - ID proses 1850
3 Ganjil - ID proses 1851

multiprocessing.Pool
1 Ganjil - ID proses 1852
2 Genap - ID proses 1853
3 Ganjil - ID proses 1855

Sekuensial : 3.0121808899595615 detik
Kelas Process : 1.0215392112731934 detik
Kelas Pool : 1.0386604551191602 detik
valentino@valentino-VirtualBox:~/sisop/tugas8$
```