

Nama : Awal Lidya Musaffak

NPM : 21083010088

Sistem Operasi A

Soal latihan :

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

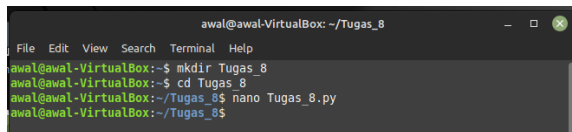
Batasan :

Nilai yang dijadikan argumen pada fungsi sleep() adalah satu detik.

Masukkan jumlah'nya satu dan berupa bilangan bulat.

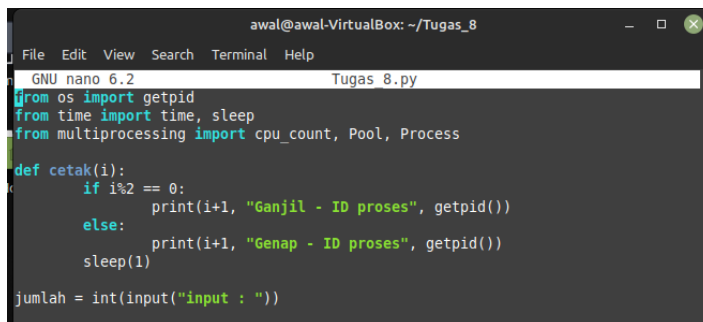
Masukkan adalah batas dari perulangan tersebut.

Setelah perulangan selesai program menampilkan waktu eksekusi pemrosesan sekuensial dan paralel.



```
awal@awal-VirtualBox: ~/Tugas_8
File Edit View Search Terminal Help
awal@awal-VirtualBox:~$ mkdir Tugas_8
awal@awal-VirtualBox:~$ cd Tugas_8
awal@awal-VirtualBox:~/Tugas_8$ nano Tugas_8.py
awal@awal-VirtualBox:~/Tugas_8$
```

Hal pertama yaitu membuat direktori baru untuk tugas 8, kemudian membuat file nano Tugas_8.py



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

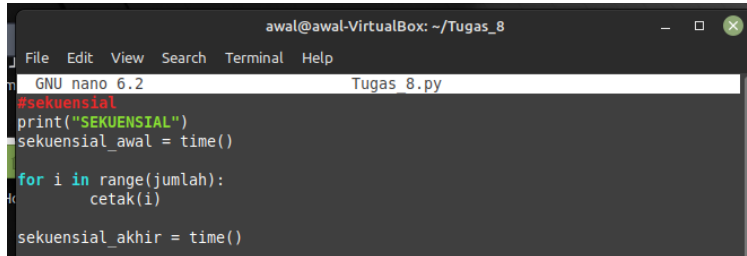
jumlah = int(input("input : "))
```

Import library yang akan digunakan untuk mengeksekusi program, disini saya import getpid untuk mengambil id proses, time untuk mengambil waktu dalam satuan detik, sleep untuk memberi jeda waktu, cpu_count untuk melihat jumlah CPU. Pool disini merupakan sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada computer sedangkan process adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada computer.

Selanjutnya membuat fungsi def cetak dengan parameter i. kemudian membuat fungsi if else apabila index i dibagi dengan 2 sama dengan 0 maka akan dilakukan perulangan index i

ditambahkan 1 begitu juga apabila index i dibagi dengan 2 tidak dengan 0 maka akan dilakukan perulangan index i ditambahkan 1. Dengan adanya `sleep(1)` nantinya saat perulangan akan diberikan jeda masing-masing 1 detik.

Masukkan inputan ke dalam variabel jumlah.

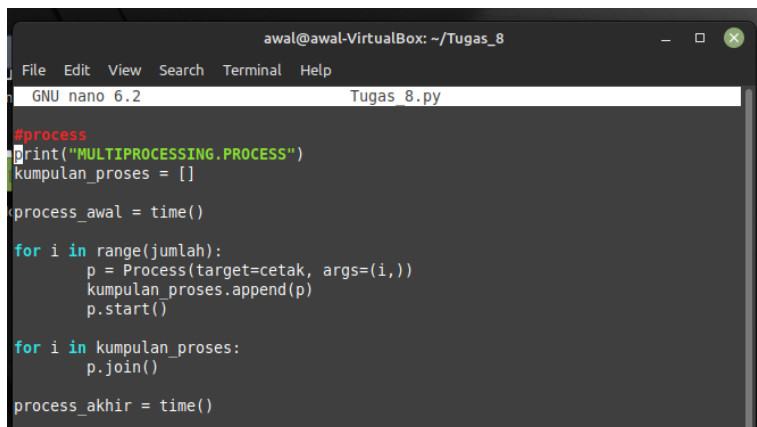


```
awal@awal-VirtualBox: ~/Tugas_8
GNU nano 6.2 Tugas_8.py
#sekuensial
print("SEKUENSIAL")
sekuensial_awal = time()

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

sekuensial_akhir = time()
```

Print “sekuensial” untuk menandakan masuk ke dalam tahap sekuensial. Variabel `sekuensial_awal` dilakukan untuk mendapatkan waktu sebelum di eksekusi, ketika proses berlangsung (menggunakan loop) dengan index i sebanyak jumlah yang telah diinputkan. Kemudian hasil dari looping tersebut akan tersimpan pada variabel `sekuensial_akhir`.



```
awal@awal-VirtualBox: ~/Tugas_8
GNU nano 6.2 Tugas_8.py
#process
print("MULTIPROCESSING.PROCESS")
kumpulan_proses = []

process_awal = time()

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

for i in kumpulan_proses:
    p.join()

process_akhir = time()
```

Print “multiprocessing.proces” untuk menandakan kita masuk ke dalam tahap multiprocessing dengan kelas `process`. Deklarasikan variabel `kumpulan_proses` untuk menampung semua proses-proses. Kemudian untuk mendapatkan proses awal waktu yang belum dieksekusi dapat disimpan menggunakan variabel `process_awal`. Pada saat proses berlangsung dilakukan looping dengan setiap index i pada variabel jumlah yang telah diinputkan akan dilakukan `p` sebagai proses(`target=cetak, args=i`) nantinya akan ditambahkan ke dalam kumpulan process dan untuk kembali ke awal digunakan `p.start`. Untuk menggabungkan proses agar tidak loncat ke proses sebelumnya menggunakan looping dari setiap i yang tersimpan di dalam kumpulan `proses` nantinya akan dijadikan satu dengan fungsi `p.join`. Dan setelah seluruh proses eksekusi selesai akan dimasukkan ke dalam variabel `process_akhir`.

```

#pool
print("MULTIPROCESSING.POOL")
pool_awal = time()

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

pool_akhir = time()

#membandingkanwaktu
print("MEMBANDINGKAN WAKTU")
print("Sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Kelas Process :", process_akhir - process_awal, "detik")
print("Kelas Pool:", pool_akhir - pool_awal, "detik")

```

Print “multiprocessing.pool” untuk menandakan kita masuk ke dalam tahap multiprocessing dengan kelas pool. Untuk mendapatkan waktu sebelum di eksekusi kita dapat menyimpannya ke dalam variabel pool_awal. Pada saat proses berlangsung pool.map(cetak, range(0, jumlah)). Dan untuk mendapatkan waktu setelah di eksekusi dapat disimpan ke dalam variabel pool_akhir.

Print semua tahapan processing untuk membandingkan waktu.

```

awal@awal-VirtualBox:~/Tugas_8$ nano Tugas_8.py
awal@awal-VirtualBox:~/Tugas_8$ python3 Tugas_8.py
input : 3
SEKUENSIAL
1 Ganjil - ID proses 3340
2 Genap - ID proses 3340
3 Ganjil - ID proses 3340
MULTIPROCESSING.PROCESS
1 Ganjil - ID proses 3341
2 Genap - ID proses 3342
3 Ganjil - ID proses 3343
MULTIPROCESSING.POOL
1 Ganjil - ID proses 3344
2 Genap - ID proses 3345
3 Ganjil - ID proses 3344
MEMBANDINGKAN WAKTU
Sekuensial : 3.0068135261535645 detik
Kelas Process : 1.0543522834777832 detik
Kelas Pool: 1668016236.2409594 detik

```

Berikut merupakan output yang akan muncul setelah kita mengeksekusi file nano.