

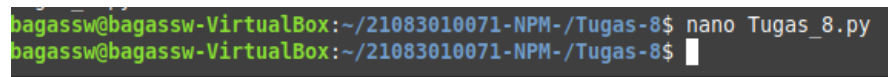
TUGAS 8

LAPORAN TUGAS

1. Latihan Soal

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

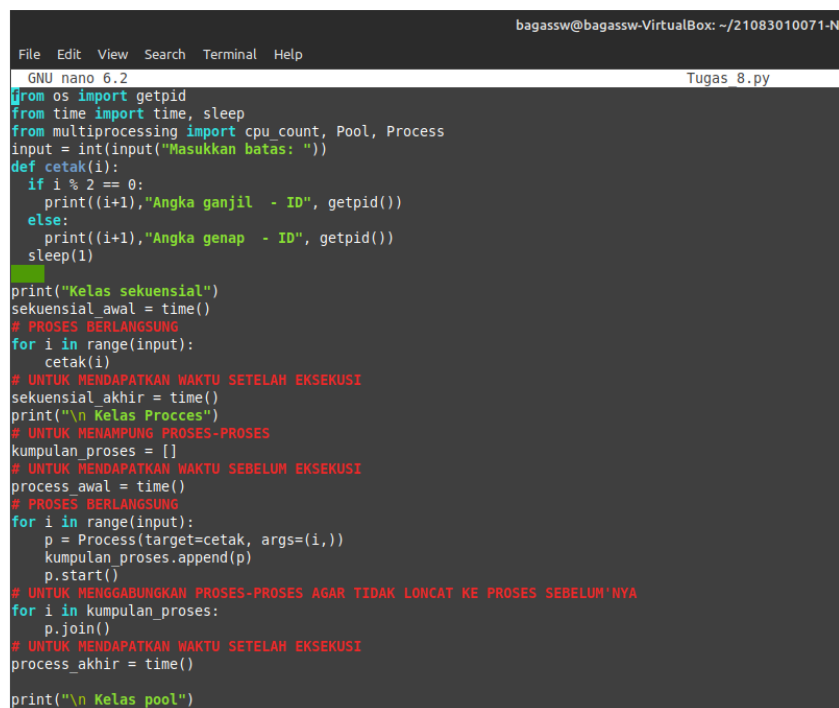
1. Pertama kita menggunakan perintah nano untuk membuat file



```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-8$ nano Tugas_8.py
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-8$
```

Gambar 1 Pembuatan file

2. lalu kita isi script seperti gambar dibawah



```
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
input = int(input("Masukkan batas: "))
def cetak(i):
    if i % 2 == 0:
        print((i+1), "Angka ganjil - ID", getpid())
    else:
        print((i+1), "Angka genap - ID", getpid())
        sleep(1)

print("Kelas sekuensial")
sekuensial_awal = time()
# PROSES BERLANGSUNG
for i in range(input):
    cetak(i)
# UNTUK MENDAPATKAN WAKTU SETELAH EKSEKUSI
sekuensial_akhir = time()
print("\n Kelas Procces")
# UNTUK MENAMPUNG PROSES-PROSES
kumpulan_proses = []
# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
process_awal = time()
# PROSES BERLANGSUNG
for i in range(input):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()
# UNTUK MENGGABUNGKAN PROSES-PROSES AGAR TIDAK LONCAT KE PROSES SEBELUM'NYA
for i in kumpulan_proses:
    p.join()
# UNTUK MENDAPATKAN WAKTU SETELAH EKSEKUSI
process_akhir = time()
print("\n Kelas pool")
```

Gambar 1.2 1 Isi script

```

print("\n Kelas pool")
# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
pool_awal = time()

# PROSES BERLANGSUNG
pool = Pool()
pool.map(cetak, range(0,input))
pool.close()
# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
pool_akhir = time()

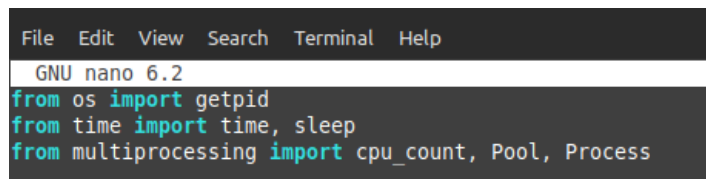
print("\n Hasilnya")
print("Sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Kelas Process :", process_akhir - process_awal, "detik")
print("Kelas Pool :", pool_akhir - pool_awal, "detik")

```

Gambar 1.2 2 Lanjutan isi script

2. Penjelasan Script

1. Import terlebih dahulu library library yang dibutuhkan di dalam script



```

File Edit View Search Terminal Help
GNU nano 6.2
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

```

Gambar 2 1 Import library

2. Inisialisasi variable untuk memasukkan inputan sebagai batas dari sebuah perulangan yaitu `input = int(input("Masukkan batas: "))`. Dan inisialisasi function cetak dengan parameter `i` dan kondisi jika `i` modulus `2 = 0` maka cetak `i + 1`, angka ganjil, ID proses. Jika kondisi tidak terpenuhi maka cetak `i + 1`, angka genap, ID proses

```

input = int(input("Masukkan batas: "))

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

```

Gambar 2 2 Deklarasi input dan function

3. Kelas sekuensial yaitu inisialisasi waktu awal sekuensial, melakukan perulangan iterasi `i` hingga batas inputan lalu cetak `i`, dan inisialisasi waktu akhir sekuensial.

```

print("Kelas sekuensial")
sekuensial_awal = time()
# PROSES BERLANGSUNG
for i in range(input):
    cetak(i)
# UNTUK MENDAPATKAN WAKTU SETELAH EKSEKUSI
sekuensial_akhir = time()

```

Gambar 2 3 Kelas sekuensial

4. Kelas multiprocessing pertama inialisasi variable untuk menampung kumpulan proses,waktu proses awal,dilanjutkan dengan perulangan iterasi i hingga batas input dengan p=target fungsi=cetak,argument=i dan semua proses p dikumpulkan pada kumpulan proses dan setelah dikumpulkan akan dijalankan , lalu untuk p.join untuk menyatukan semuanya agar proses tidak terus berulang ke proses sebelumnya.lalu simpan waktu akhir proses

```
print("\n Kelas Proses")
# UNTUK MENAMPUNG PROSES-PROSES
kumpulan_proses = []
# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
process_awal = time()
# PROSES BERLANGSUNG
for i in range(input):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()
# UNTUK MENGGABungkan PROSES-PROSES AGAR TIDAK LONCAT KE PROSES SEBELUM'NYA
for i in kumpulan_proses:
    i.join()
# UNTUK MENDAPATKAN WAKTU SETELAH EKSEKUSI
process_akhir = time()
```

Gambar 2 4 Kelas multi processing

5. Kelas process pool pertama inialisasi variable waktu awal proses lalu mendefinisikan pool merujuk ke fungsi library pool () lalu menjalankan perulangan dengan fungsi cetak lalu range 0 hingga input / batas lalu selesai dan simpan waktu akhir.

```
print("\n Kelas pool")
# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
pool_awal = time()
# PROSES BERLANGSUNG
pool = Pool()
pool.map(cetak, range(0,input))
pool.close()
# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
pool_akhir = time()
```

Gambar 2 5 Kelas processing pool

6. Menghitung semua waktu dari setiap kelas kelas yang telah dijalankan,yaitu dengan cara mengurangi waktu akhir dengan waktu awal setiap kelas proses yang dijalankan

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

Gambar 2 6 Hasil waktu setiap kelas

3. Output

Dengan menggunakan script dan input yang sama ternyata linux dan google colab memiliki hasil output yang berbeda

- Linux

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-8$ python3 Tugas_8.py
Masukkan batas: 3
Kelas sekuensial
1 Angka ganjil - ID 7405
2 Angka genap - ID 7405
3 Angka ganjil - ID 7405

Kelas Procces
2 Angka genap - ID 7407
3 Angka ganjil - ID 7408
1 Angka ganjil - ID 7406

Kelas pool
1 Angka ganjil - ID 7409
2 Angka genap - ID 7409
3 Angka ganjil - ID 7409

Hasilnya
Sekuensial : 3.001816749572754 detik
Kelas Process : 1.0062801837921143 detik
Kelas Pool : 3.0104167461395264 detik
```

Gambar 3 1 Hasil output linux

- Google Colab

```
Masukkan batas: 3
Kelas sekuensial
1 Angka ganjil - ID 5855
2 Angka genap - ID 5855
3 Angka ganjil - ID 5855

Kelas Procces
12 Angka genap - IDAngka ganjil - ID3 5905Angka ganjil - ID5902
5912

Kelas pool
12 Angka genap - IDAngka ganjil - ID 59415942
3 Angka ganjil - ID 5942

Hasilnya
Sekuensial : 3.004152297973633 detik
Kelas Process : 1.097538709640503 detik
Kelas Pool : 2.092423915863037 detik
```

Gambar 3 2 Hasil output google colab