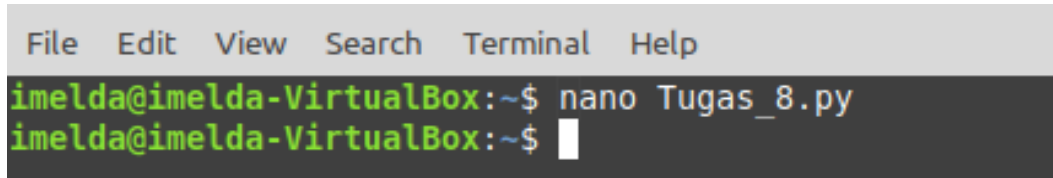


Nama : Imelda Widya Ningrum
NPM : 21083010052
Kelas : SISOP A

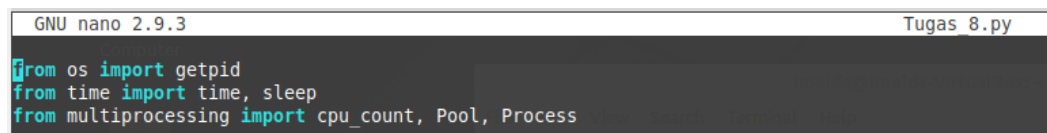
LAPORAN TUGAS 8

1. Membuat file dengan command `nano Tugas_8.py`



```
File Edit View Search Terminal Help
imelda@imelda-VirtualBox:~$ nano Tugas_8.py
imelda@imelda-VirtualBox:~$
```

2. Membuat script python
 - Import library python



```
GNU nano 2.9.3 Tugas_8.py
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process
```

Library python yang digunakan `sleep`, `cpu_count`, `Pool`, `Process`

- Inisialisasi fungsi



```
a = int(input("masukkan batasan: "))
def cetak(i):
    for i in range(a):
        if i % 2 == 0:
            print(f"bilangan ganjil-{i+1}", "- ID Proses ", getpid())
        else:
            print(f"bilangan genap-{i+1}", "- ID Proses ", getpid())
    sleep(1)
```

Fungsi di atas digunakan untuk mencetak angka dari variabel `i` beserta ID proses sejumlah parameter yang diberikan. Disini menggunakan metode percabangan `if else`. Lalu, kita panggil fungsi `sleep` untuk memberi jeda waktu(detik) sebanyak parameter yang diberikan.

- Sekuensial



```
#Sekuensial
print("Sekuensial")
sekuensial_awal = time()

for i in range(1):
    cetak(i)
sekuensial_akhir = time()
print(" ")
```

Program Sekuensial adalah program yang di dalam perintahnya disusun secara sekuensial (perintah disusun dari baris-perbaris dan tidak ada perintah percabangan, pengulangan, dan fungsi).

- Multiprocessing Process

```
#Multiprocessing Process
print("Multiprocessing.Process")
kumpulan_proses = []
proses_awal = time()

for i in range(1):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()
for i in kumpulan_proses:
    p.join()
proses_akhir=time()
print(" ")
```

Multiprocessing ini menawarkan konkurensi lokal dan jarak jauh Untuk menunjukkan ID proses individual yang terlibat

- Multiprocessing Pool

```
#Multiprocessing Pool
print("Multiprocessing.pool")
pool_awal = time()
pool = Pool()
pool.map(cetak, range(0,1))
pool.close()
pool_akhir=time()
print(" ")
```

Paralel pelaksanaan fungsi di beberapa nilai input, mendistribusikan data input di seluruh proses (paralelisme data).

- Perbandingan Waktu Eksekusi

```
#Perbandingan Waktu Eksekusi
print("Perbandingan Waktu eksekusi")
print("Sekuensial      : ",sekuensial_akhir - sekuensial_awal, "detik")
print("Multiprocessing.Process : ",proses_akhir - proses_awal, "detik")
print("Multiprocessing.Pool      : ",pool_akhir - pool_awal, "detik")
```

Perbandingan waktu eksekusi dari sekuensial, multiprocessing process, dan multiprocessing pool

3. Menjalankan script

```
imelda@imelda-VirtualBox: ~  
File Edit View Search Terminal Help  
imelda@imelda-VirtualBox:~$ python3 Tugas_8.py  
masukkan batasan: 4  
Sekuensial  
bilangan ganjil-1 - ID Proses 1487  
bilangan genap-2 - ID Proses 1487  
bilangan ganjil-3 - ID Proses 1487  
bilangan genap-4 - ID Proses 1487  
  
Multiprocessing.Process  
bilangan ganjil-1 - ID Proses 1488  
bilangan genap-2 - ID Proses 1488  
bilangan ganjil-3 - ID Proses 1488  
bilangan genap-4 - ID Proses 1488  
  
Multiprocessing.pool  
bilangan ganjil-1 - ID Proses 1489  
bilangan genap-2 - ID Proses 1489  
bilangan ganjil-3 - ID Proses 1489  
bilangan genap-4 - ID Proses 1489  
  
Perbandingan Waktu eksekusi  
Sekuensial : 1.0021960735321045 detik  
Multiprocessing.Process : 1.1301822662353516 detik  
Multiprocessing.Pool : 1.233417272567749 detik
```

Batas nilai $i = 4$ yang artinya proses data hanya sampai variabel 4.