

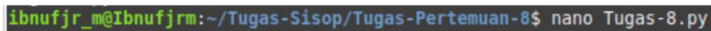
Nama : Mohamad Ibnu Fajar Maulana

Npm : 21083010106

Kelas : Sistem Operasi B

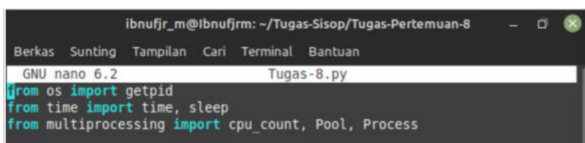
### Latihan Soal Multiprocessing

- a) Buatlah file dengan “nano Tugas-8.py”, seperti gambar dibawah ini.



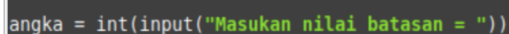
```
ibnufjr_m@ibnufjr:~/Tugas-Sisop/Tugas-Pertemuan-8$ nano Tugas-8.py
```

- b) Lakukan Coding seperti bawah ini.



```
ibnufjr_m@ibnufjr: ~/Tugas-Sisop/Tugas-Pertemuan-8
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 6.2      Tugas-8.py
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process
```

- Import modul seperti diatas yang mana memiliki kegunaan tersendiri
  - getpid digunakan untuk mendapatkan proses id.
  - time digunakan untuk mengambil waktu pada proses dijalankan atau diakhiri.
  - sleep digunakan untuk menunda waktu eksekusi.
  - cpu\_count digunakan untuk menghitung berapa banyak cpu yang tersedia.
  - Pool digunakan untuk melakukan pemrosesan parallel dengan menggunakan proses sebanyak jumlah CPU pada computer.
  - Process digunakan untuk melakukan pemrosesan parallel dengan menggunakan proses secara beruntun pada computer.



```
angka = int(input("Masukan nilai batasan = "))
```

- Membuat parameter angka agar user dapat menginputkan nilai batasan yang diinginkan oleh user sendiri.



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

- Terdapat fungsi def cetak(i): dengan bantuan percabangan yang mana.  
*if (i + 1)%2 == 0*: akan menghasilkan nilai genap kemudian dengan kejadian lain seperti *else*: akan menghasilkan nilai ganjil.

```

print("Sekunsial")
# UNTUK Mendapatkan Waktu Sebelum Eksekusi
sekuensial_awal = time()

# PROSES BERLANGSUNG
for i in range(angka):
    cetak(i)

# UNTUK Mendapatkan Waktu Setelah Eksekusi
sekuensial_akhir = time()

```

- Lakukan Sekunsial processing dengan looping for seperti gambar diatas.

```

print("multiprocessing.process")

#MULTIPROCESSING DENGAN KELAS PROCESS
process_awal=time()
for i in range(angka):
    p=Process(target=cetak, args=(i, ))
    p.start()
    p.join()
process_akhir=time()

```

- Lakukan Multiprocessing dengan kelas process yang mana.
  - process\_awal = time() untuk menyimpan waktu awal mulainya proses.
  - Kemudian lakukan looping for dengan range(angka) nilai batasan yang diinginkan oleh user.
  - p.start memiliki kegunaan dalam mengeksekusi fungsi parameter def cetak.
  - p.join() digunakan untuk meanmpung kumpulan proses menjadi satu agar proses tidak merambah ke proses selanjutnya.
  - process\_akhir=time() untuk mendapatkan waktu sebelum eksekusi.

```

print("multiprocessing.pool")

# UNTUK Mendapatkan Waktu Sebelum Eksekusi
pool_awal = time()

# PROSES BERLANGSUNG
pool = Pool()
pool.map(cetak, range(angka))
pool.close()

# UNTUK Mendapatkan Waktu Sebelum Eksekusi
pool_akhir = time()

```

- Kemudian lakukan multiprocessing dengan kelas pool. Perhatikan tanda # yang merupakan penjelasan logika dari script tersebut.

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

- Setelah itu, kita melakukan script diatas agar dapat membandingkan waktu yang dieksekusi.
- c) Jika masih belum terbayang silahkan liat eksekusi script file tersebut dibawah ini dengan “python3 Tugas-8.py”.

```
ibnufjr_m@Ibnufjrm:~/Tugas-Sisop/Tugas-Pertemuan-8$ python3 Tugas-8.py
Masukan nilai batasan = 3
Sekunsial
1 Ganjil -punya ID proses 2396
2 Genap -punya ID proses 2396
3 Ganjil -punya ID proses 2396
multiprocessing.process
1 Ganjil -punya ID proses 2397
2 Genap -punya ID proses 2398
3 Ganjil -punya ID proses 2400
multiprocessing.pool
1 Ganjil -punya ID proses 2401
2 Genap -punya ID proses 2401
3 Ganjil -punya ID proses 2401
Sekuensial : 3.0038530826568604 detik
Kelas Process : 3.020146608352661 detik
Kelas Pool : 3.0378592014312744 detik
ibnufjr_m@Ibnufjrm:~/Tugas-Sisop/Tugas-Pertemuan-8$
```

- Pada gambar diatas user menginputkan nilai batasan berjumlah 3.
- Pada Sekunsial, bisa dilihat bahwa memiliki ID proses yang sama.
- Kemudian pada multiprocessing kelas process memiliki nilai id proses yang berbeda namun beruntun. Karena proses tersebut dibantu dengan fungsi sleep(1) yang mana memberikan proses waktu satu detik.
- Kemudian Multiprocessing kelas pool memiliki nilai id proses yang sama.