

Nama : Dendy Arizki Kuswardana  
NPM : 21083010006  
Kelas : Sistem Operasi (A)

## LAPORAN TUGAS 8

### 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.

### 1. Membuat File .py

Sebagai awalan saya membuat file yang berbentuk .py dengan menggunakan nano seperti gambar di bawah ini.

```
bash: cd: Tugas8: no such file or directory
dendy@dendy:~/Sisop$ cd Tugas8
dendy@dendy:~/Sisop/Tugas8$ nano Tugas8.py
dendy@dendy:~/Sisop/Tugas8$ bash Tugas8.py
```

Script awal yang perlu diimport yaitu built-in libraries yang diperlukan untuk menjalankan keseluruhan script nantinya. Antara lain;

- 1) getpid digunakan untuk mengambil ID proses.
- 2) time digunakan untuk mengambil waktu (detik).
- 3) sleep digunakan untuk memberi jeda waktu (detik).
- 4) cpu\_count digunakan untuk melihat jumlah CPU.
- 5) Pool digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada komputer.
- 6) Process digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada komputer.

```
GNU nano 6.2
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

def cetak(i):
```

## 2. Merubah Nama Fungsi

yang akan digunakan dengan penyesuaian nama masing-masing fungsi. Pada gambar di bawah ini terdapat fungsi conditional statements if else untuk menentukan apakah sebuah bilangan ganjil atau genap. Dengan diberi tambahan fungsi sleep yang digunakan untuk memberi jeda waktu selama 1 detik.

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

## 3. Penyebutan “n”

n disini dijadikan sebagai penyebutan nama yang memiliki fungsi untuk inputan range angka dengan tipe integer.

```
n = int(input("Masukkan Range Angka : "))
```

## 4. Melakukan Proses Sekuensial

Penggunaan fungsi *time*, agar dalam proses ini terdapat sebelum dan sesudah eksekusi. Berikutnya terdapat *loop for i in* digunakan untuk memproses sekuensial sebanyak range n.

```
print("Sekuensial")  
  
# sebagai waktu sebelum eksekusi  
sekuensial_awal = time()  
  
# mengproses  
for i in range(n):  
    cetak(i)  
  
# sebagai waktu setelah eksekusi  
sekuensial_akhir = time()
```

## 5. Multiprocessing Dengan Process

Terdapat proses *multiprocessing* dengan kelas *Process*. Disini saya menggunakan fungsi *time* untuk mendapatkan waktu sebelum dan sesudah eksekusi. Inisialisasikan array kosong untuk menampung kumpulan proses. Lalu, *loop for i in* digunakan untuk memproses multiprocessing dengan kelas *Process* sebanyak range n. Pada loop ini masukkan fungsi cetak yang telah diinisialisasikan sebelumnya, tetapi dengan ID proses yang berbeda-beda. Hal ini berarti tiap pemanggilan fungsi cetak ditangani oleh satu proses saja. Kemudian untuk pemanggilan selanjutnya ditangani oleh proses yang lain. Kumpulan proses ditampung dan digabung menjadi satu pada *p.join()* agar tidak merambah ke proses selanjutnya.

```

print("multiprocessing.Process")
# sebagai wadah seluruh proses
kumpulan_proses = []

# sebagai waktu sebelum eksekusi
process_awal = time()

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

# menyatukan keseluruhan proses
for i in kumpulan_proses:
    p.join()

# sebagai waktu setelah eksekusi
process_akhir = time()

```

## 6. *Multiprocessing Dengan Kelas Pool.*

Pada proses ini juga menggunakan fungsi time yang bertujuan mendapatkan waktu sebelum dan sesudah eksekusi. Fungsi map() digunakan untuk memetakan pemanggilan fungsi cetak ke dalam CPU yang dimiliki komputer sebanyak n kali. Jumlah ID proses terbatas tergantung jumlah CPU pada komputer.

```

print("multiprocessing.Pool")
# sebagai waktu sebelum eksekusi
pool_awal = time()

# mengproses
pool = Pool()
pool.map(cetak, range(0,n))
pool.close

# sebagai waktu setelah eksekusi
pool_akhir = time()

```

## 7. Menampilkan Waktu

Agar dapat menampilkan waktu eksekusi pada pemrosesan sekuensial dan parallel, maka diperlukannya mengurangi waktu sesudah dengan waktu sebelum eksekusi seperti gambar di bawah ini.

```

print("Waktu eksekusi sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Waktu eksekusi multiprocessing.Process :", process_akhir - process_awal, "detik")

```

## 8. Hasil Output

Dari script yang telah dibuat, akan dapat menampilkan keseluruhan perintah yang telah diinput pada file Tugas8.py. untuk menguji, saya memasukkan angka 8. Dari angka yang telah saya masukkan tersebut, maka batas dari perulangan pemrosesan sekuensial dan paralel adalah sebanyak 8. Dibagian terakhir akan memunculkan waktu yang digunakan untuk tiap-tiap proses eksekusinya seperti gambar di bawah ini.

```
dendy@dendy:~/Sisop/Tugas8$ nano Tugas8.py
dendy@dendy:~/Sisop/Tugas8$ python3 Tugas8.py
Masukkan Range Angka : 8
Sekuensial
1 Ganjil - ID proses 1972
2 Genap - ID proses 1972
3 Ganjil - ID proses 1972
4 Genap - ID proses 1972
5 Ganjil - ID proses 1972
6 Genap - ID proses 1972
7 Ganjil - ID proses 1972
8 Genap - ID proses 1972
multiprocessing.Process
3 Ganjil - ID proses 1975
4 Genap - ID proses 1976
8 Genap - ID proses 1980
5 Ganjil - ID proses 1977
6 Genap - ID proses 1978
2 Genap - ID proses 1974
7 Ganjil - ID proses 1979
1 Ganjil - ID proses 1973
multiprocessing.Pool
1 Ganjil - ID proses 1981
2 Genap - ID proses 1981
3 Ganjil - ID proses 1981
4 Genap - ID proses 1981
5 Ganjil - ID proses 1981
6 Genap - ID proses 1981
7 Ganjil - ID proses 1981
8 Genap - ID proses 1981
Waktu eksekusi sekuensial : 8.010557413101196 detik
Waktu eksekusi multiprocessing.Process : 1.0222156047821045 detik
Waktu eksekusi multiprocessing.Pool : 8.027279615402222 detik
```