

Nama : Fadlila Agustina  
NPM : 21083010050  
Kelas : Sistem Operasi (B)

## Multiprocessing

Pemrograman paralel adalah sebuah teknik eksekusi perintah yang mana dilakukan secara bersamaan pada CPU. Seluruh bahasa pemrograman yang populer dapat melakukan pemrograman paralel dengan modul bawaan atau memang pengaturan defaultnya seperti itu. Manfaat Multiprocessing :

- Menggunakan CPU untuk komputasi.
- Tidak berbagi sumber daya memori.
- Memerlukan sumber daya memori dan waktu yang tidak sedikit.
- Tidak memerlukan sinkronisasi memori.

### Latihan Soal

Langkah yang pertama adalah membuat file bernama Tugas\_8.py menggunakan perintah nano.

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

**getpid** digunakan untuk mengambil ID proses.

**time** digunakan untuk mengambil waktu (detik).

**sleep** digunakan untuk memberi jeda waktu (detik).

**cpu\_count** digunakan untuk melihat jumlah CPU.

**Pool** adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada komputer.

**Process** adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada komputer.

```
bil = int(input("Masukkan batasan: "))
def cetak(i):
    for i in range(bil):
        if i % 2 == 0:
            print(f"{i+1} Ganjil", " - ID Proses ", getpid() )
        else:
            print(f"{i+1} Genap", " - ID Proses" , getpid())
        sleep(1)
```

Mendeklarasikan bilangan dengan menginput batasan yang diinginkan di soal yaitu 3. Fungsi di atas digunakan untuk mencetak angka dari variabel i beserta proses sejumlah parameter yang diberikan. Lalu menggunakan looping **for** yang dibatasi oleh batasan tadi. Adapun **if else** yang menunjukkan looping dari batasan yg dimasukkan itu merupakan bilangan ganjil atau genap. **Sleep(1)** agar terjeda selama 1 detik.

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

for i in range(1):
    cetak(i)

sekuensial_akhir = time()
print(" ")
```

Yang pertama pada pemrosesan sekuensial mendapatkan waktu sebelum eksekusi, kemudian menggunakan **looping for** dengan **range 1** yang berarti setiap perulangan hanya terjadi sekali. **Sekuensial\_akhir** digunakan untuk mendapatkan waktu setelah eksekusi.

```
#Multiprocessing dengan kelas Process
print("Multiprocessing.Process")
kumpulan_proses = []
process_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()

process_akhir = time()
print(" ")
```

Kedua ada multiprocessing dengan Process. Langkah pertama adalah menampung seluruh proses lalu mendapatkan waktu sebelum eksekusi. Sedikit berbeda dengan proses sekuensial, proses ini menggunakan **looping for** dan seluruh proses ditampung dan digabung menjadi satu menggunakan **p.join()** agar tidak merambah proses selanjutnya. Setelah itu mendapatkan waktu sesudah eksekusi.

```
#Multiprocessing dengan kelas Pool
print("Multiprocessing.Pool")
pool_awal = time()

pool = Pool()
pool.map(cetak, range(0,1))
pool.close()

pool_akhir = time()
print(" ")
```

Proses ketiga yaitu dengan kelas pool. Fungsi **map()** digunakan untuk memetakan pemanggilan fungsi cetak ke dalam CPU sebanyak sekali.

```
#Bandingkan waktu eksekusi
print("Waktu eksekusi sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Waktu eksekusi multiprocessing.Process :", process_akhir - process_awal, "detik")
print("Waktu eksekusi multiprocessing.Pool :", pool_akhir - pool_awal, "detik")
```

Pada perbandingan waktu eksekusi, waktu eksekusi sekuensial didapatkan dari pengurangan waktu sekuensial akhir dan sekuensial awal dalam satuan detik. Waktu eksekusi multiprocessing.Process didapatkan dari pengurangan waktu process akhir dan process awal dalam satuan detik. Waktu eksekusi multiprocessing.Pool didapatkan dari pengurangan waktu pool akhir dan pool awal dalam satuan detik.

```
fadlila@fadlila-VirtualBox:~$ python3 Tugas_8.py
Masukkan batasan: 3
Sekensial
1 Ganjil - ID Proses 3060
2 Genap - ID Proses 3060
3 Ganjil - ID Proses 3060

Multiprocessing.Process
1 Ganjil - ID Proses 3061
2 Genap - ID Proses 3061
3 Ganjil - ID Proses 3061

Multiprocess.Pool
1 Ganjil - ID Proses 3062
2 Genap - ID Proses 3062
3 Ganjil - ID Proses 3062

Waktu eksekusi sekuensial : 1.0035388469696045 detik
Waktu eksekusi multiprocessing.Process : 1.0112030506134033 detik
Waktu eksekusi multiprocessing.Pool : 1.0200669765472412 detik
fadlila@fadlila-VirtualBox:~$
```

Untuk memunculkan output dapat menggunakan perintah python3 nama file.py