

Multiprocessing

- **Memuat built-in libraries yang akan digunakan**

```
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 gunanya untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada computer
- **Process** adalah sebuah class pada library multiprocessing yang gunanya untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada computer

- **Inisialisasi fungsi yang akan digunakan**

```
# Inisialisasi fungsi yang akan digunakan:
def cetak(i):
    print("Cetak angka", i+1, "- punya ID proses", getpid())
    sleep(1)
```

Function di atas digunakan untuk mencetak angka (yang tersimpan dalam variabel i) beserta ID proses sejumlah parameter yang diberikan. Lalu sleep untuk memberi jeda waktu(detik) sebanyak parameter yang diberikan.

- **1. Pemrosesan sekuensial**

Pemrosesan sekuensial ini berlangsung secara berurutan/satu-satu (pemanggilan fungsi cetak ditangani oleh proses yang sama)

```
# 1. Pemrosesan Sekuensial
print("1. Pemrosesan Sekuensial")
# Untuk mendapatkan waktu sebelum eksekusi
sekuensial_awal = time()

# Proses berlangsung
for i in range(10):
    cetak(i)

# Untuk mendapatkan waktu setelah eksekusi
sekuensial_akhir = time()
print("-"*50)
```

Perulangan for untuk i dalam rentang jumlah parameter 10

Output :

```
1. Pemrosesan Sekuensial
Cetak angka 1 - punya ID proses 1715
Cetak angka 2 - punya ID proses 1715
Cetak angka 3 - punya ID proses 1715
Cetak angka 4 - punya ID proses 1715
Cetak angka 5 - punya ID proses 1715
Cetak angka 6 - punya ID proses 1715
Cetak angka 7 - punya ID proses 1715
Cetak angka 8 - punya ID proses 1715
Cetak angka 9 - punya ID proses 1715
Cetak angka 10 - punya ID proses 1715
-----
```

- **2. Multiprocessing dengan kelas Process**

Pemrosesan multiprocessing ini berlangsung secara paralel/bersamaan (pemanggilan fungsi cetak ditangani oleh satu proses saja, kemudian untuk pemanggilan selanjutnya ditangani oleh proses yang lain) dengan menggunakan proses secara beruntun pada komputer

```
# 2. Multiprocessing dengan kelas Process
print("2. Multiprocessing dengan kelas Process")
# Untuk menampung proses-proses
kumpulan_proses = []

# Untuk mendapatkan waktu sebelum eksekusi
process_awal = time()

# Proses berlangsung
for i in range(10):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()

# Untuk menggabungkan proses-proses agar tidak loncat ke proses sebelumnya
for i in kumpulan_proses:
    p.join()

# Untuk mendapatkan waktu setelah eksekusi
process_akhir = time()
print("-"*50)
```

Output

```
2. Multiprocessing dengan kelas Process
Cetak angka 1 - punya ID proses 1716
Cetak angka 3 - punya ID proses 1718
Cetak angka 2 - punya ID proses 1717
Cetak angka 4 - punya ID proses 1719
Cetak angka 6 - punya ID proses 1721
Cetak angka 5 - punya ID proses 1720
Cetak angka 8 - punya ID proses 1723
Cetak angka 7 - punya ID proses 1722
Cetak angka 9 - punya ID proses 1724
Cetak angka 10 - punya ID proses 1725
```

- **3. Multiprocessing dengan kelas Pool**

Pemrosesan multiprocessing ini berlangsung secara paralel dengan menggunakan proses sebanyak jumlah CPU pada computer.

```
# 3. Multiprocessing dengan kelas Pool:
print("3. Multiprocessing dengan kelas Pool")

# Untuk mendapatkan waktu sebelum eksekusi
pool_awal = time()

# Proses berlangsung
pool = Pool()
pool.map(cetak, range(0,10))
pool.close()

# Untuk mendapatkan waktu sebelum eksekusi
pool_akhir = time()
print("-"*50)
```

Output

```
3. Multiprocessing dengan kelas Pool
Cetak angka 1 - punya ID proses 1726
Cetak angka 2 - punya ID proses 1726
Cetak angka 3 - punya ID proses 1726
Cetak angka 4 - punya ID proses 1726
Cetak angka 5 - punya ID proses 1726
Cetak angka 6 - punya ID proses 1726
Cetak angka 7 - punya ID proses 1726
Cetak angka 8 - punya ID proses 1726
Cetak angka 9 - punya ID proses 1726
Cetak angka 10 - punya ID proses 1726
```

- **Membandingkan waktu eksekusi**

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

Output :

```
Bandingkan waktu eksekusi
Sekuensial : 10.013147115707397 detik
Kelas Process : 1.1147277355194092 detik
Kelas Pool: 10.14079236984253 detik
```

Proses sekuensial lebih lambat dibandingkan dengan multiprocessing (proses paralel). Untuk multiprocessing kelas Process lebih cepat dibandingkan kelas Pool. Penggunaan metode-metode processing ini disesuaikan dengan kebutuhan.

Penjelasan Latihan Soal

Contoh input :

3

Contoh Output :

```
Sekuensial
1 Ganjil - ID proses ****
2 Genap - ID proses ****
3 Ganjil - ID proses ****

multiprocessing.Process
1 Ganjil - ID proses ****
2 Genap - ID proses ****
3 Ganjil - ID proses ****

multiprocessing.Pool
1 Ganjil - ID proses ****
2 Genap - ID proses ****
3 Ganjil - ID proses ****

Waktu eksekusi sekuensial : ** detik
Waktu eksekusi multiprocessing.Process : ** detik
Waktu eksekusi multiprocessing.Pool : ** detik
```

Code

Pertama, memuat built-in libraries yang akan digunakan

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

Kedua, membuat variabel input agar user dapat memasukkan jumlah batasnya.

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

Ketiga, inisialisasi fungsi untuk mencetak a sebagai bilangan ganjil apabila $a \bmod 2 = 1$ dan mencetak bilangan genap untuk pengecualiannya serta jeda waktu yang diberikan adalah 1 detik.

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

Karena i dimulai dari 0 maka perlu ditambah 1 agar sesuai dengan output yang diharapkan

Keempat, pemrosesan sekuensial

```
# 1. Pemrosesan Sekuensial
print("- Pemrosesan Sekuensial -")

# Untuk mendapatkan waktu sebelum eksekusi
sekuensial_awal = time()

# Proses berlangsung
for i in range(batas):
    cetak(i)

# Untuk mendapatkan waktu setelah eksekusi
sekuensial_akhir = time()
print("")
```

Untuk rentang yang digunakan adalah nilai yang telah diinputkan oleh user dalam variable batas

Kelima, pemrosesan multiprocessing kelas Process

```
# 2. Multiprocessing dengan kelas Process
print("- Multiprocessing dengan kelas Process -")

# Untuk menampung proses-proses
kumpulan_proses = []

# Untuk mendapatkan waktu sebelum eksekusi
process_awal = time()

# Proses berlangsung
for i in range(batas):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()

# Untuk menggabungkan proses-proses agar tidak loncat ke proses sebelumnya
for i in kumpulan_proses:
    p.join()

# Untuk mendapatkan waktu setelah eksekusi
process_akhir = time()
print("")
```

Keenam, pemrosesan multiprocessing kelas Pool

```
# 3. Multiprocessing dengan kelas Pool:
print("- Multiprocessing dengan kelas Pool -")

# Untuk mendapatkan waktu sebelum eksekusi
pool_awal = time()

# Proses berlangsung
pool = Pool()
pool.map(cetak, range(batas))
pool.close()

# Untuk mendapatkan waktu sebelum eksekusi
pool_akhir = time()
print("")
print("-"*50)
```

Terakhir, membandingkan kecepatan waktu eksekusi untuk tiap processingnya

```
# Bandingkan waktu eksekusi
print("Perbandingan 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")
```

Output

```
Masukkan batas : 3
- Pemrosesan Sekuensial -
1 Ganjil - ID proses 1734
2 Genap - ID proses 1734
3 Ganjil - ID proses 1734

- Multiprocessing dengan kelas Process -
1 Ganjil - ID proses 1735
2 Genap - ID proses 1736
3 Ganjil - ID proses 1737

- Multiprocessing dengan kelas Pool -
1 Ganjil - ID proses 1738
2 Genap - ID proses 1738
3 Ganjil - ID proses 1738

-----
Perbandingan waktu eksekusi
Waktu eksekusi sekuensial : 3.0037899017333984 detik
Waktu eksekusi multiprocessing.Process : 1.046330213546753 detik
Waktu eksekusi multiprocessing.Pool: 3.070023775100708 detik
```

Full Script

```
GNU nano 6.2                                     Tugas_8.py
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

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

# Inisialisasi fungsi
def cetak(i):
    a = i + 1
    if a % 2 == 1:
        print(a, "Ganjil - ID proses", getpid())
    else:
        print(a, "Genap - ID proses", getpid())
        sleep(1)

# 1. Pemrosesan Sekuensial
print("- Pemrosesan Sekuensial -")

# Untuk mendapatkan waktu sebelum eksekusi
sekuensial_awal = time()

# Proses berlangsung
for i in range(batas):
    cetak(i)

# Untuk mendapatkan waktu setelah eksekusi
sekuensial_akhir = time()
print("")

# 2. Multiprocessing dengan kelas Process
print("- Multiprocessing dengan kelas Process -")

# Untuk menampung proses-proses
kumpulan_proses = []

# Untuk mendapatkan waktu sebelum eksekusi
process_awal = time()
```

```
# Proses berlangsung
for i in range(batas):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()

# Untuk menggabungkan proses-proses agar tidak loncat ke proses sebelumnya
for i in kumpulan_proses:
    p.join()

# Untuk mendapatkan waktu setelah eksekusi
process_akhir = time()
print("")

# 3. Multiprocessing dengan kelas Pool:
print("- Multiprocessing dengan kelas Pool -")

# Untuk mendapatkan waktu sebelum eksekusi
pool_awal = time()

# Proses berlangsung
pool = Pool()
pool.map(cetak, range(batas))
pool.close()

# Untuk mendapatkan waktu sebelum eksekusi
pool_akhir = time()
print("")
print("-"*50)

# Bandingkan waktu eksekusi
print("Perbandingan 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")
```