Nama: Citra Amelia Intan Permadani

NPM: 21083010004

Kelas: SISTEM OPERASI-B

Penerapan dan Latihan Soal Multiprocessing

A. Penerapan

Langkah:

1. Buatlah folder dengan nama Tugas-Sisop_8 menggunakan perintah mkdir
2. Lalu masuk ke dalam folder Tugas-Sisop_8 menggunakan perintah cd

```
citra@citra-VirtualBox:~$ mkdir Tugas-Sisop_8
citra@citra-VirtualBox:~$ cd Tugas-Sisop_8
```

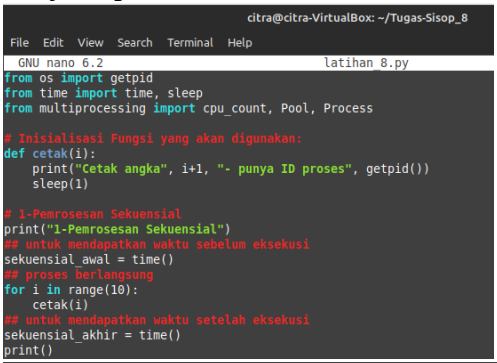
3. Buat file .py baru ‘disini saya memberi nama latihan_8.py’ dengan perintah nano

```
citra@citra-VirtualBox:~/Tugas-Sisop_8$ nano latihan_8.py
```

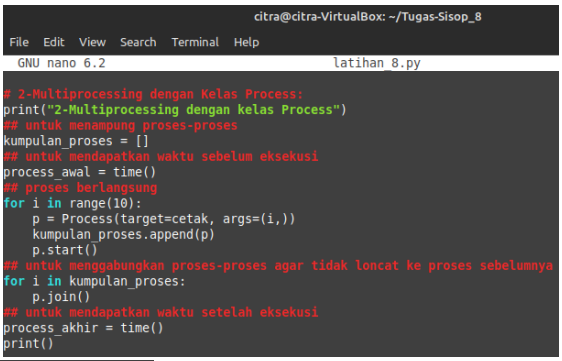
4. Karena, kita akan membuat file python, kita dapat mengecek apakah kita sudah menginstall python atau belum dengan mengecek versinya seperti di bawah ini. Jika sudah terinstall maka akan muncul output python version dalam linux kita

```
citra@citra-VirtualBox:~/Tugas-Sisop_8$ python3 --version
Python 3.10.4
```

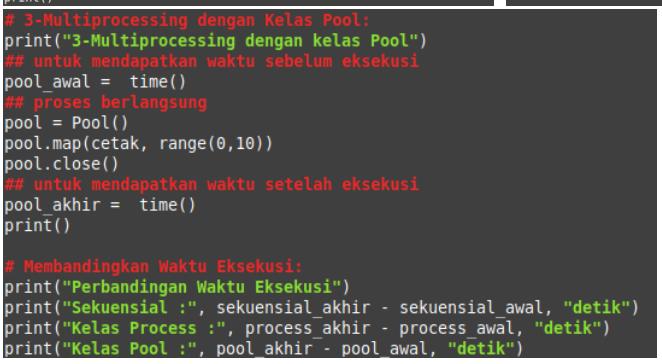
5. Lalu, buka kembali GNU Nano dengan perintah nano latihan_8.py dan ketikkan isinya seperti di bawah ini



```
citra@citra-VirtualBox: ~/Tugas-Sisop_8
File  Edit  View  Search  Terminal  Help
 GNU nano 6.2                    latihan_8.py
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

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

# 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()
```



```
citra@citra-VirtualBox: ~/Tugas-Sisop_8
File  Edit  View  Search  Terminal  Help
 GNU nano 6.2                    latihan_8.py
# 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()
```



```
# 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 setelah eksekusi
pool_akhir =  time()
print()

# Membandingkan Waktu Eksekusi:
print("Perbandingan 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")
```

6. Simpan file dengan CTRL+X pilih y dan tutup dengan enter
7. Lalu jalankan file latihan_8.py dengan perintah python3 latihan_8.py (disini saya menggunakan python3). Maka akan muncul output seperti di bawah ini.

```
citra@citra-VirtualBox:~/Tugas-Sisop_8$ python3 latihan_8.py
1-Pemrosesan Sekuensial
Cetak angka 1 - punya ID proses 1974
Cetak angka 2 - punya ID proses 1974
Cetak angka 3 - punya ID proses 1974
Cetak angka 4 - punya ID proses 1974
Cetak angka 5 - punya ID proses 1974
Cetak angka 6 - punya ID proses 1974
Cetak angka 7 - punya ID proses 1974
Cetak angka 8 - punya ID proses 1974
Cetak angka 9 - punya ID proses 1974
Cetak angka 10 - punya ID proses 1974

2-Multiprocessing dengan kelas Process
Cetak angka 1 - punya ID proses 1975
Cetak angka 2 - punya ID proses 1976
Cetak angka 3 - punya ID proses 1977
Cetak angka 4 - punya ID proses 1978
Cetak angka 5 - punya ID proses 1979
Cetak angka 6 - punya ID proses 1980
Cetak angka 7 - punya ID proses 1981
Cetak angka 8 - punya ID proses 1982
Cetak angka 9 - punya ID proses 1983
Cetak angka 10 - punya ID proses 1984
```

```
3-Multiprocessing dengan kelas Pool
Cetak angka 1 - punya ID proses 1985
Cetak angka 2 - punya ID proses 1986
Cetak angka 3 - punya ID proses 1987
Cetak angka 4 - punya ID proses 1988
Cetak angka 5 - punya ID proses 1987
Cetak angka 6 - punya ID proses 1985
Cetak angka 7 - punya ID proses 1986
Cetak angka 8 - punya ID proses 1988
Cetak angka 9 - punya ID proses 1987
Cetak angka 10 - punya ID proses 1986

Perbandingan Waktu Eksekusi
Sekuensial : 10.030864238739014 detik
Kelas Process : 1.0304484367370605 detik
Kelas Pool : 3.0468361377716064 detik
```

B. Latihan Soal

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.**

Dan output yang diminta:

Contoh *input* :

```
3
```

Contoh *Output* :

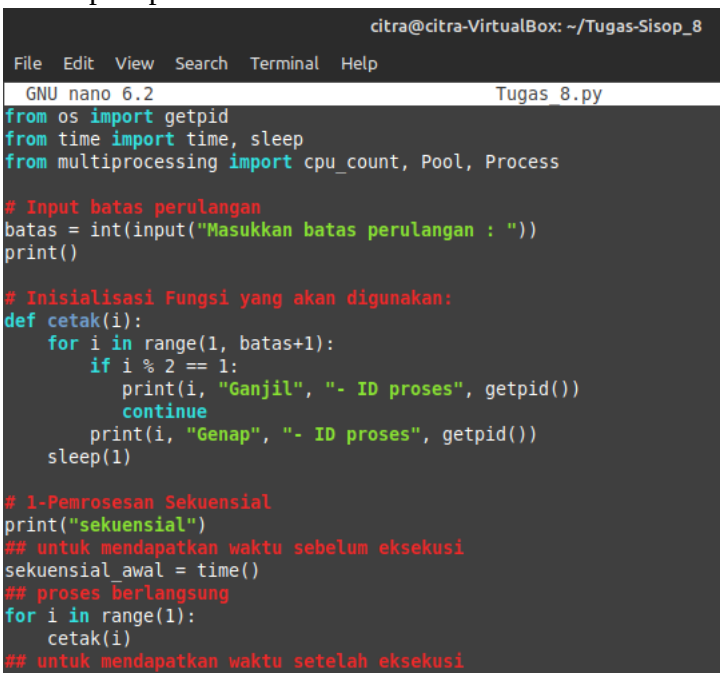
```
Sekuensial
1 Ganjil - ID proses ****
2 Genap - ID proses ****
3 Ganjil - ID proses ****

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
```

Jawaban:

1. Tetap di folder Tugas-Sisop_8 dan Buatlah file dengan format .py menggunakan perintah nano “nano Tugas_8.py”

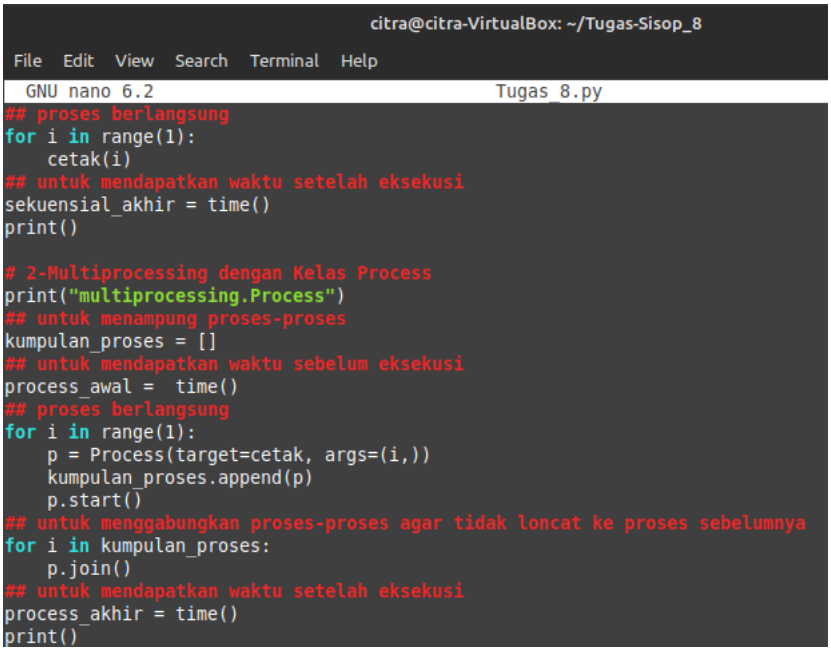
```
citra@citra-VirtualBox:~/Tugas-Sisop_8$ nano Tugas_8.py
```

2. Isi script seperti di bawah ini



   a. Kita import terlebih dahulu library-library dari modul di python
      - ⇨ 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
   b. Lalu kita buat inputan untuk batasan perulangan dengan memasukkan bilangan bulat atau integer dan kita beri nama ‘batas’
   c. Kita inisialisasikan fungsi yang digunakan menjadi output dimana kita menggunakan def cetak(i), i merupakan range antara 1 hingga batas+1. Lalu kita berikan perulangan dibawahnya dengan menggunakan if continue, jika hasil sisa pembagian i dengan 2 adalah 1 maka akan mengprint ganjil lalu dilanjutkan genap dan akan kembali lagi ke ganjil hingga batas perulangan yang ditetapkan. Disini kita memakai sleep (1) untuk memberi jeda waktu sebanyak parameter yang diberikan.
   d. Kita masuk pada Pemrosesan Sekuensial
      - ⇨ Kita memanggil time untuk mendapatkan waktu sebelum eksekusi di awal

- ⇨ Dan ketika proses berlangsung yang akan dijalankan adalah untuk i dalam range 1, (kenapa 1? Karena kita menggunakan batasan semisal 3 untuk 1 perulangan). Lalu i akan dicetak
- ⇨ Kemudian kita memanggil time lagi untuk mendapatkan waktu setelah eksekusi di akhir



e. Masuk ke multiprocessing kedua yakni Multiprocessing dengan Kelas Process
   - ⇨ Kita gunakan kurung [] sebagai tampungan proses-proses dalam list/index dengan nama kumpulan_proses
   - ⇨ Kita gunakan time untuk mendapatkan waktu sebelum eksekusi
   - ⇨ Prosesnya adalah untuk i dalam rentang 1, kita buat p = Process (target disamadengankan cetak, dan kita gunakan args agar fungsi memungkinan menerima semua argumen tanpa mengetahui seberapa banyak)
     Lalu tampungan proses di awal tadi kita panggil dan ditambahkan dengan append untuk menambahkan proses p dan kita gunakan start untuk memulai proses p
   - ⇨ Untuk menggabungkan proses-proses agar tidak loncat ke proses selanjutnya, kita gunakan for i dalam kumpulan_proses dan kita panggil fungsi join
   - ⇨ Proses sudah selesai dan kita memanggil time kembali untuk mendapatkan waktu setelah eksekusi di akhir.

```
citra@citra-VirtualBox: ~/Tugas-Sisop_8
File  Edit  View  Search  Terminal  Help
  GNU nano 6.2                                  Tugas_8.py
    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.Pool")
## untuk mendapatkan waktu sebelum eksekusi
pool_awal = time()
## proses berlangsung
pool = Pool()
pool.map(cetak, range(0,1))
pool.close()
## untuk mendapatkan waktu setelah eksekusi
pool_akhir = time()
print()

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

f. Yang ketiga adalah Multiprocessing dengan Kelas Pool
   - ⇨ Kita gunakan time untuk mendapatkan waktu sebelum eksekusi di awal
   - ⇨ Prosesnya:
     Memanggil fungsi Pool (), kemudian gunakan map (cetak, dan rangenya dari 0, 1) lalu ditutup dengan close ()
   - ⇨ Kita dapatkan waktu setelah eksekusi dengan time

g. Dan yang terakhir adalah membandingkan waktu eksekusi
   Dengan mengeprint satu-satu tiap langkah multiprocessing kemudian time akhir – time awal dengan keterangan detik

3. Lalu simpan dengan CTRL+X dan pilih y kemudian tekan enter untuk menutup
4. Eksekusi file py dengan perintah python3 Tugas_8.py dan akan menghasilkan hasil seperti di bawah ini. Maka hasil sudah sesuai dengan output yang diinginkan soal.

```
citra@citra-VirtualBox:~/Tugas-Sisop_8$ python3 Tugas_8.py
Masukkan batas perulangan : 3

sekuensial
1 Ganjil - ID proses 2142
2 Genap - ID proses 2142
3 Ganjil - ID proses 2142

multiprocessing.Process
1 Ganjil - ID proses 2143
2 Genap - ID proses 2143
3 Ganjil - ID proses 2143

multiprocessing.Pool
1 Ganjil - ID proses 2144
2 Genap - ID proses 2144
3 Ganjil - ID proses 2144

Waktu eksekusi sekuensial : 1.0135953426361084 detik
Waktu multiprocessing.Process : 1.023578405380249 detik
Waktu multiprocessing.Pool : 1.0208046436309814 detik
```