

Nama : Alvin Ryan Dana
NPM : 21083010035
Kelas : Sistem Operasi B

TUGAS 8

MULTIPROCESSING

Latihan Soal

Dengan menggunakan pemrosesan paralel buatlah program yang dapat menentukan sebuah bilangan itu ganjil atau genap!.

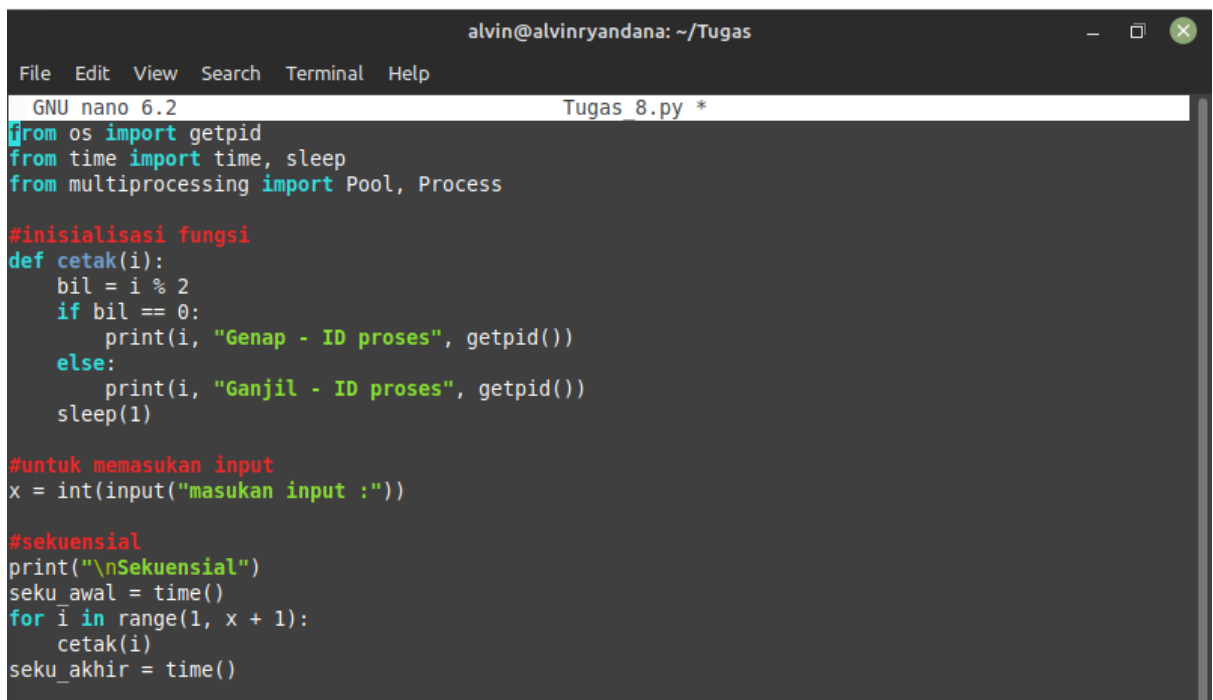
Batasan :

- 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. menggunakan nano untuk membuat file

```
alvin@alvinryandana:~/Tugas$ nano Tugas 8.py
```

2. Script



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 Tugas 8.py *
from os import getpid
from time import time, sleep
from multiprocessing import Pool, Process

#inisialisasi fungsi
def cetak(i):
    bil = i % 2
    if bil == 0:
        print(i, "Genap - ID proses", getpid())
    else:
        print(i, "Ganjil - ID proses", getpid())
    sleep(1)

#untuk memasukan input
x = int(input("masukan input :"))

#sekuensial
print("\nSekuensial")
seku_awal = time()
for i in range(1, x + 1):
    cetak(i)
seku_akhir = time()
```

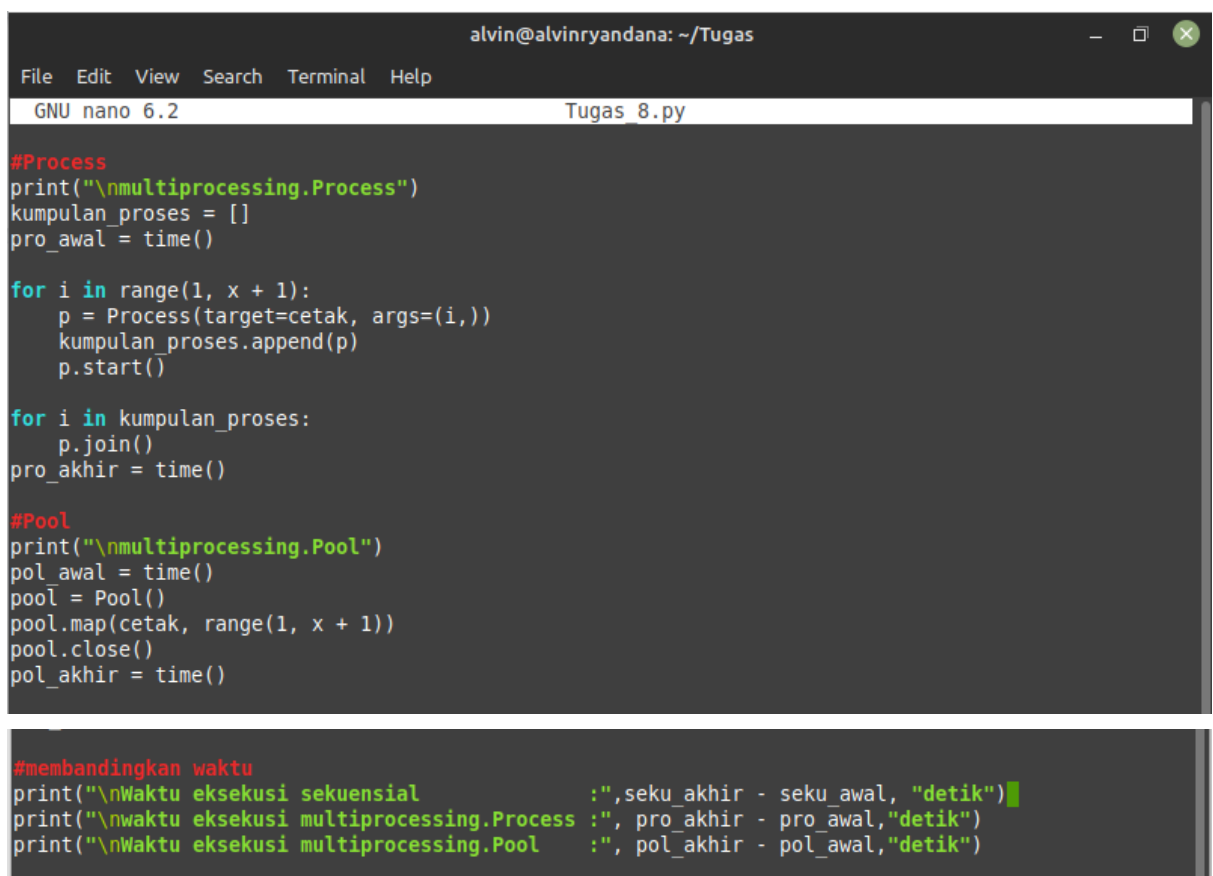
getpid adalah function yang mengembalikan ID proses yang berjalan, time merupakan sebuah function yang berguna untuk mengambil detik waktu, sleep berfungsi untuk menanggguhkan eksekusi perintah dalam jumlah waktu yang diberikan. Process digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun, dan Pool digunakan untuk melakukan pemrosesan paralel menggunakan jumlah CPU yang terdapat pada komputer

#inisialisasi fungsi

bil = 1 % 2 dilakukan perhitungan modulus untuk mengetahui apakah bilangan tersebut genap atau ganjil lalu apabila hasil perhitungan adalah 0 maka bilangan (i) adalah genap dan mengeluarkan output Genap - ID proses sebaliknya jika hasil perhitungan tidak 0 maka bilangan (i) adalah ganjil. getpid berguna untuk mendapatkan id proses, lalu sleep digunakan untuk memberikan jeda selama 1 detik

#Sekuensial

menggunakan function time() untuk mengambil waktu yang disimpan pada variabel seku_awal, lalu menggunakan looping for untuk melakukan perulangan i dalam range (1, x + 1) dan menggunakan function cetak. pada akhir sekuensia menggunakan time() untuk mengambil waktu yang disimpan dalam variabel seku_akhir



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
GNU nano 6.2 Tugas 8.py

#Process
print("\nmultiprocessing.Process")
kumpulan_proses = []
pro_awal = time()

for i in range(1, x + 1):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()

for i in kumpulan_proses:
    p.join()
pro_akhir = time()

#Pool
print("\nmultiprocessing.Pool")
pol_awal = time()
pool = Pool()
pool.map(cetak, range(1, x + 1))
pool.close()
pol_akhir = time()

#membandingkan waktu
print("\nWaktu eksekusi sekuensial          :", seku_akhir - seku_awal, "detik")
print("\nWaktu eksekusi multiprocessing.Process :", pro_akhir - pro_awal, "detik")
print("\nWaktu eksekusi multiprocessing.Pool      :", pol_akhir - pol_awal, "detik")
```

#Process

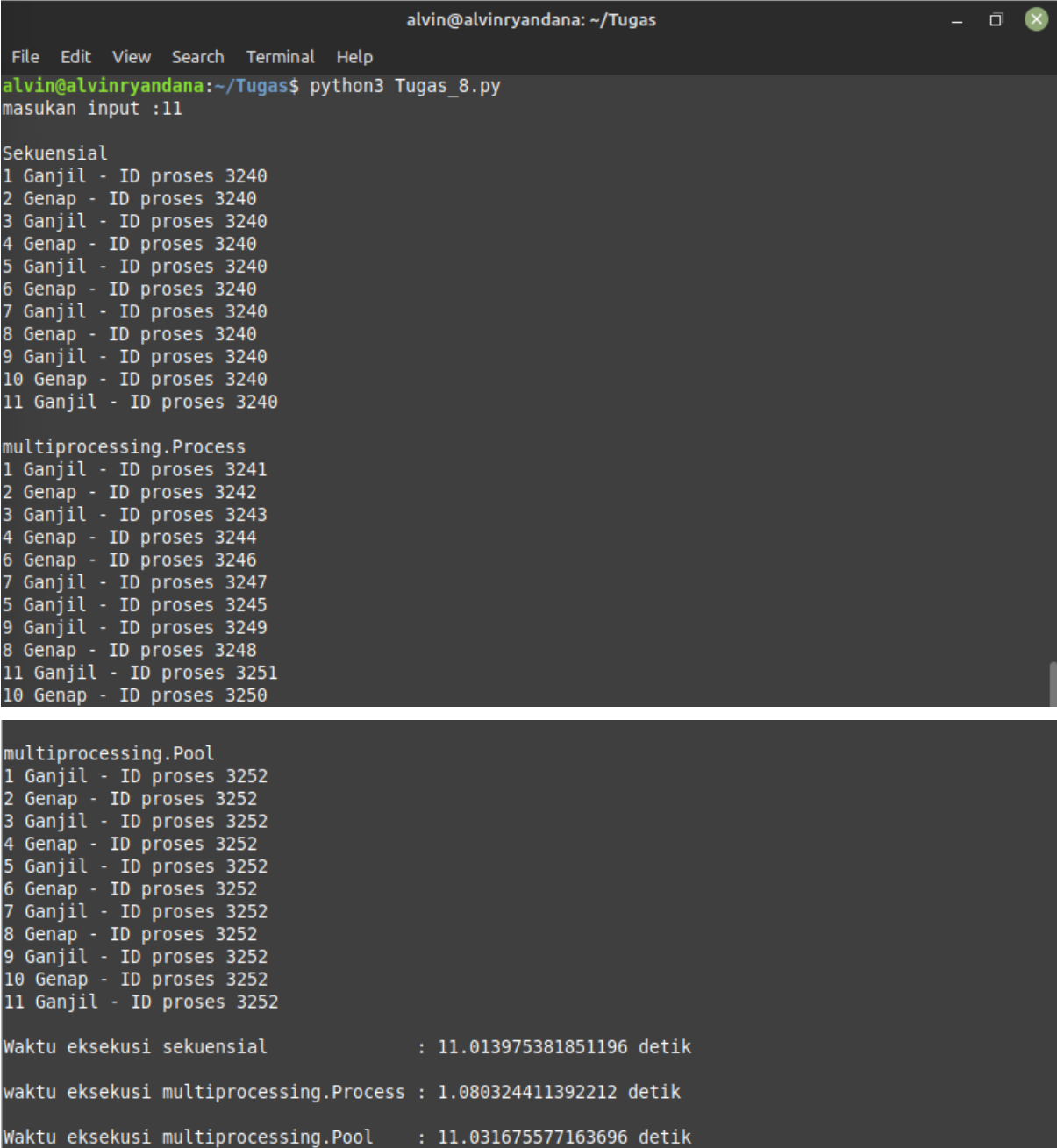
mendeklarasikan list kosong dengan nama variabel kumpulan_proses kemudian menggunakan looping for sama seperti sekuensial dan mendeklarasikan p yang berisikan Process(target=cetak, args=(i,)), lalu memasukan ke dalam list kumpulan_proses, looping yang kedua menggunakan p.join() untuk menggabungkan kumpulan proses yang telah ditampung

#Pool

mendeklarasikan `pool = Pool()` lalu untuk menjabarkan pemanggilan function cetak ke dalam CPU dapat dengan menggunakan `pool.map` dengan range yang sama seperti pemrosesan sebelumnya.

Untuk mendapatkan nilai waktu setiap pemrosesan dapat melalui perhitungan antara waktu akhir dikurangi dengan waktu awal dari setiap proses

3. Output



```
alvin@alvinryandana: ~/Tugas
File Edit View Search Terminal Help
alvin@alvinryandana:~/Tugas$ python3 Tugas_8.py
masukan input :11

Sekuensial
1 Ganjil - ID proses 3240
2 Genap - ID proses 3240
3 Ganjil - ID proses 3240
4 Genap - ID proses 3240
5 Ganjil - ID proses 3240
6 Genap - ID proses 3240
7 Ganjil - ID proses 3240
8 Genap - ID proses 3240
9 Ganjil - ID proses 3240
10 Genap - ID proses 3240
11 Ganjil - ID proses 3240

multiprocessing.Process
1 Ganjil - ID proses 3241
2 Genap - ID proses 3242
3 Ganjil - ID proses 3243
4 Genap - ID proses 3244
6 Genap - ID proses 3246
7 Ganjil - ID proses 3247
5 Ganjil - ID proses 3245
9 Ganjil - ID proses 3249
8 Genap - ID proses 3248
11 Ganjil - ID proses 3251
10 Genap - ID proses 3250

multiprocessing.Pool
1 Ganjil - ID proses 3252
2 Genap - ID proses 3252
3 Ganjil - ID proses 3252
4 Genap - ID proses 3252
5 Ganjil - ID proses 3252
6 Genap - ID proses 3252
7 Ganjil - ID proses 3252
8 Genap - ID proses 3252
9 Ganjil - ID proses 3252
10 Genap - ID proses 3252
11 Ganjil - ID proses 3252

Waktu eksekusi sekuensial : 11.013975381851196 detik
waktu eksekusi multiprocessing.Process : 1.080324411392212 detik
Waktu eksekusi multiprocessing.Pool : 11.031675577163696 detik
```

Pada hasil akhir `multiprocessing.process` adalah proses paling cepat dibandingkan dengan yang lainnya

