Nama : Baginda

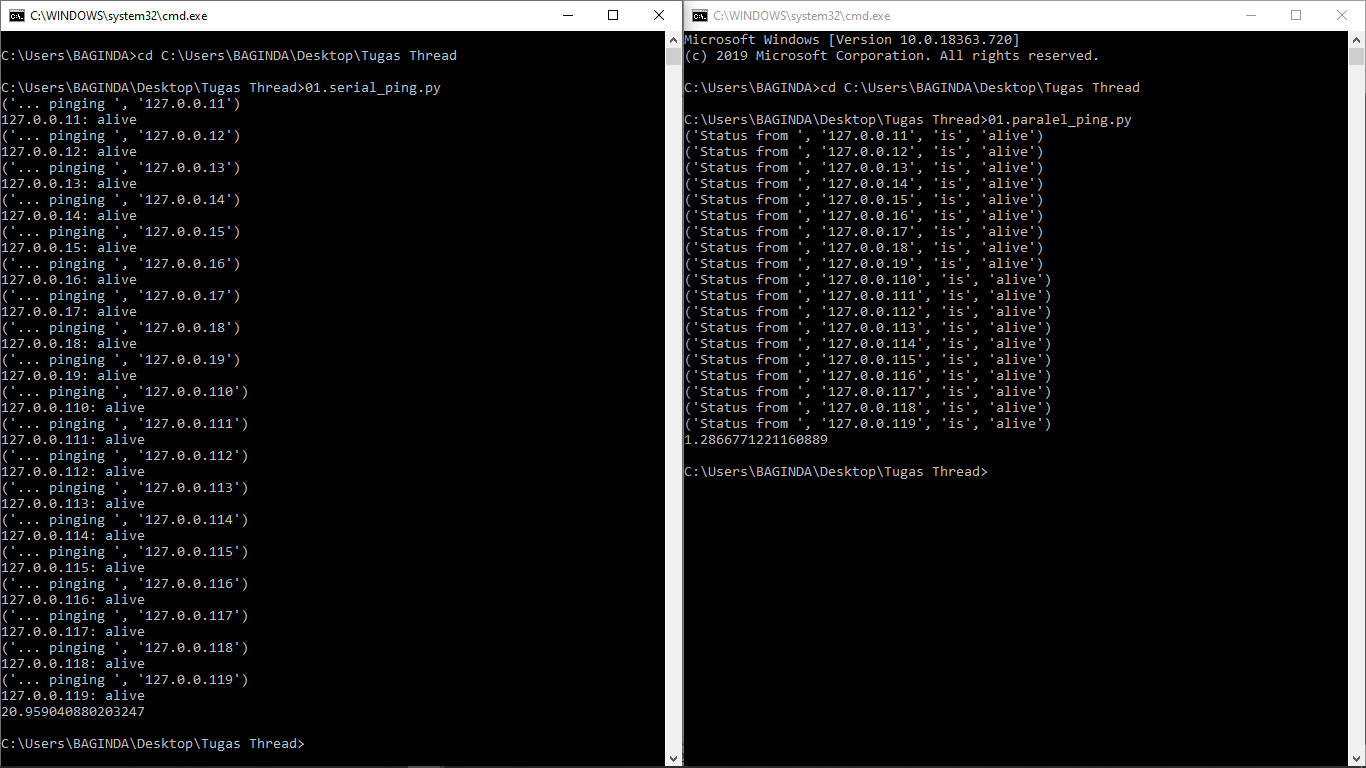
Kelas : IF-41

NIM : 130-

**CSH3J3 – SISTEM PARALEL DAN TERDISTRIBUSI**

**TUGAS THREAD**

* **Tugas 1: Parallel Ping**

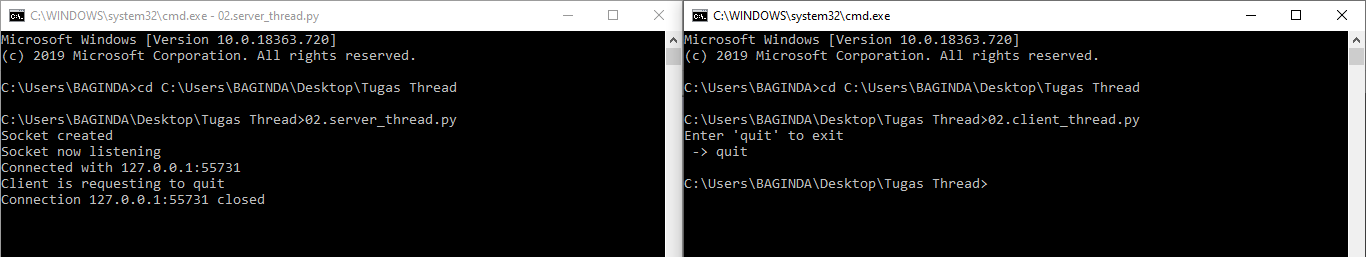
****

Program yang pertama kali dijalankan (sebelah kiri) adalah serial\_ping.py. Program ini akan melakukan PING secara serial pada *host-host* yang telah ditentukan. Setelah itu, dijalankan program kedua (sebelah kanan) yaitu paralel\_ping.py, yang mana program ini adalah program sebelumnya yang telah diubah menjadi bentuk paralel dengan menggunakan *thread*.

Serial\_ping.py melakukan PING kepada 20 *host* yang kemudian akan menghasilkan status “alive” jika *received* = 2, “alive but losses” jika *received* = 1, dan “no response” jika *received* = 0 (*received* = respon ping yang diperoleh), dan di akhir *looping* PING 20 *host*, akan ditampilkan selisih waktu akhir dengan awal untuk mengetahui lama waktu eksekusi.

Paralel\_ping.py membuat kelas ip\_check yang berperan sebagai kelas turunan dari (threading.Thread) dan *meng-override* fungsi/metode yang ada. Selanjutnya, dilakukan PING terhadap 20 *host* yang kemudian akan menghasilkan status “alive” jika *received* = 2, “alive, but 50% package loss” jika *received* = 1, “no response” jika *received* = 0, dan “shouldn’t occur” jika *received* = ELSE. Dilakukan pengambilan IP dengan *thread*, lalu IP dimasukkan ke dalam *list* penampungan. Saat *thread* dijalankan, *thread* mulai melakukan PING sampai *thread* tersebut memanggil metode join() dari *thread* lain, lalu *thread* tersebut melepaskan GIL-nya untuk menunggu *thread* yang *join* tadi untuk dimatikan sebelum diproses.

* **Tugas 2: Multithreaded Server**

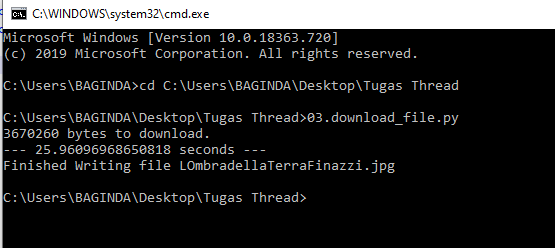
****

Pertama, server\_thread.py (sebelah kiri) membuat *server* dengan IP *localhost* “127.0.0.1” dengan *port server* 8888, lalu dibuat *socket* bertipe TCP. Kemudian, dilakukan *binding* antara IP *server* dan *port server*. Lalu, *server* menunggu untuk menerima permintaan koneksi dari *client*.

Kedua, client\_thread.py (sebelah kanan) membuat *client* dengan IP *localhost* dan dengan *port* 8888. *Client* mulai meminta koneksi ke *server*. Jika terhubung, server\_thread.py akan menampilkan bahwa *server* sudah terkoneksi dengan *client* yang meminta koneksi. Lalu, server\_thread.py akan menunggu pesan yang masuk dari *client* menggunakan *thread*.

Ketiga, client\_thread.py akan terus melakukan *looping* yang hanya akan berhenti jika *client* menuliskan/*meng-input* kata “quit”. Jika ditulis, maka server\_thread akan memutuskan koneksinya dengan *client*.

* **Tugas 3: download\_file.py**

****

Program download\_file.py membuat kelas turunan dari threading.Thread. Program ini akan *men-download* *file* secara paralel menggunakan *thread*. Program akan mencari ukuran *file* yang ada pada *link* dengan *header content-length*. Setelah itu, program akan memecah ukuran *file* menjadi tiga bagian, agar tiga buah *thread* dapat *men-download file* tersebut berdasarkan awal dan akhir masing-masing ukuran *file*. Selanjutnya, program akan mengeluarkan lamanya waktu eksekusi saat *men-download* *file* tersebut, hingga akhirnya program akan menampilkan nama *file* beserta tipenya, dan program dinyatakan selesai.