TUGAS – 4A
IMPLEMENTASI PROGRAM SERVER DAN PROTOCOL
MENGGUNAKAN MULTI-PROCESS

Kelas : Pemrograman Jaringan D

Evelyn Tjitrodjojo   05111840000099

Dosen :

Royyana Muslim Ijtihadie, S.Kom.,M.Kom., Ph.D.

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas (FTEIC)

Institut Teknologi Sepuluh Nopember (ITS)

Surabaya

2021

SOAL

Implementasi program server sesuai contoh

Semua tugas berikut ini HARUS dijalankan di SIMULATOR

1. Jalankan program server seperti dalam pembahasan
2. Buatlah program client yang dapat melakukan 100 request get pada dalam satu saat untuk operasi get file "pokijan.jpg"
3. Capture dan submitlah poin 1 dan 2 dalam satu dokumen pdf. berikan DESKRIPSI dan PENJELASAN

- Gambar Arsitektur Jaringan (Dalam Simulator Gns3)
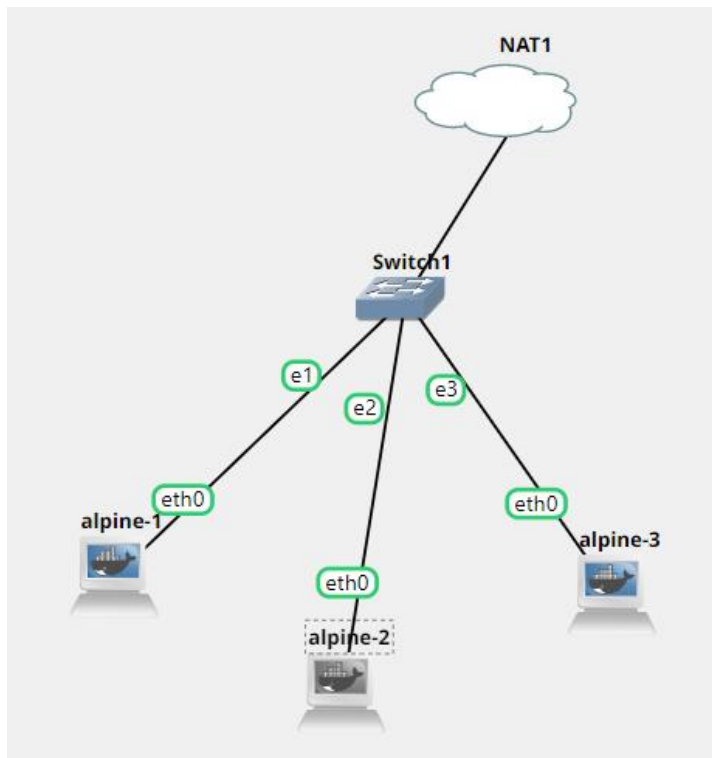
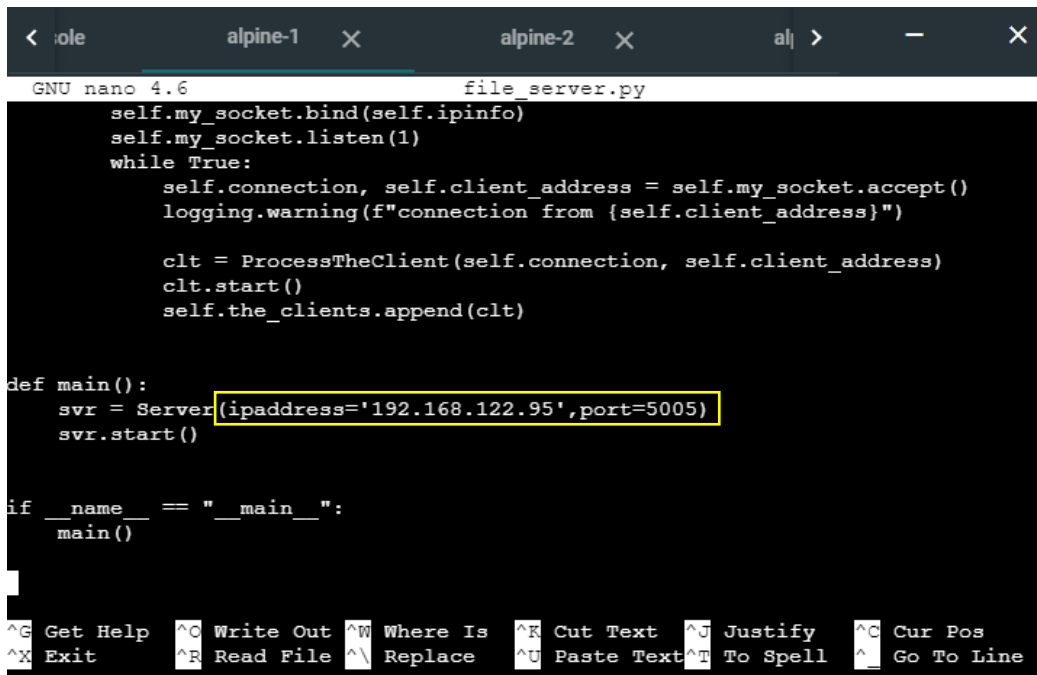Note: Alpine-1 sebagai server, Alpine 2 sebagai client, Alpine-3 tidak digunakan

- Pertama, cek terlebih dahulu IP address dari server (Alpine-1)

- Ubah IP pada file_server.py dengan IP address dari alpine-1 sebagai server.
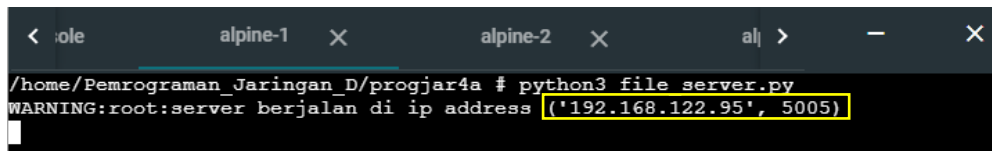
- Coba jalankan file_server.py, maka hasilnya seperti berikut.
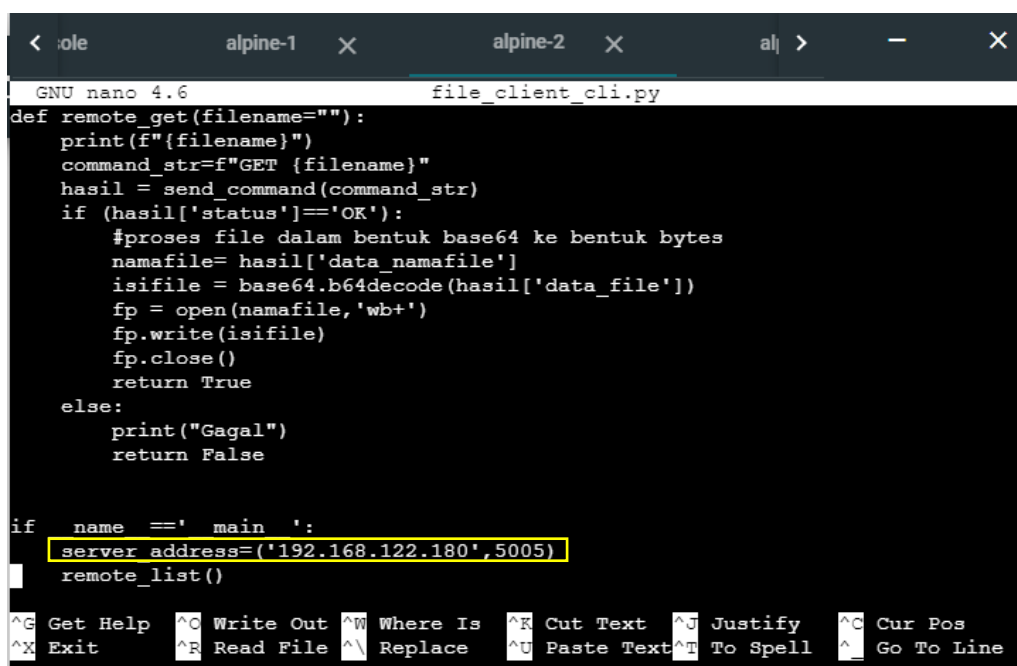
- Ubah server address pada file_client_cli.py menjadi alamat server berjalan.

- Buat file baru untuk multi process yang melakukan 100 request get file 'pokijan.jpg' dalam satu saat pada program client. Berikut gambar serta penjelasan.

```
GNU nano 4.6                    multi_process.py
from file_client_cli import remote_get
import time                                    Import fungsi remote_get pada file_client_cli.py
import datetime
from multiprocessing import Process

def download_100():
    texec = dict()                             Masukkan nama file
    catat_awal = datetime.datetime.now()
    filename = 'pokijan.jpg'
    for k in range(100):                       Perulangan sampai 100x
        print(f"download get {k}")
        waktu = time.time()
        # bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi down>
        texec[k] = Process(target=remote_get, args=(filename,))
        texec[k].start()
    # setelah menyelesaikan tugasnya, dikembalikan ke main process dengan join
    for k in range(100):                       Panggil fungsi remote_get dengan argumen filename
        texec[k].join()                        menggunakan multi proses, yang diulang sebanyak 100x.
    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
                      [ Read 26 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

```
GNU nano 4.6                    file_client_cli.py
def remote_get(filename=""):
    print(f"{filename}")
    command_str=f"GET {filename}"
    hasil = send_command(command_str)
    if (hasil['status']=='OK'):
        #proses file dalam bentuk base64 ke bentuk bytes
        namafile= hasil['data_namafile']
        isifile = base64.b64decode(hasil['data_file'])
        fp = open(namafile,'wb+')
        fp.write(isifile)
        fp.close()
        return True
    else:
        print("Gagal")
        return False
                                               Fungsi yang dipanggil dalam multi_process.py yaitu
                                               remote_get dan filename yang dikirimkan ke server.

if __name__=='__main__':
    server_address=('192.168.122.180',5005)
    remote_list()

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

HASIL RUN

Hasil run dari multi_process.py pada client (alpine-2), berjalan request get sebanyak 100x yaitu mulai 0-99.

Hasil yang diterima pada server (alpine-1) saat client di jalankan

Menerima request dan koneksi dari alamat client.