# Tugas Implementasi Concurrency

OLEH :

RYUKAZU ANDARA SAVIESTYAN

05111840000129

Pemrograman Jaringan - D

DOSEN PENGAMPU :

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

S1 TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS

INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

2021

## Tugas

1. Buatlah program yang mengimplementasikan

- Multi process
- Multi thread
- Multi process asynchronous
- Multi thread asynchronous

Dengan menggunakan protokol transport UDP, kasus dapat didefinisikan sendiri dan buatlah arsitektur jaringan anda sendiri di simulator gns3

2. Buatlah laporan dalam bentuk PDF yang berisikan screenshot dari

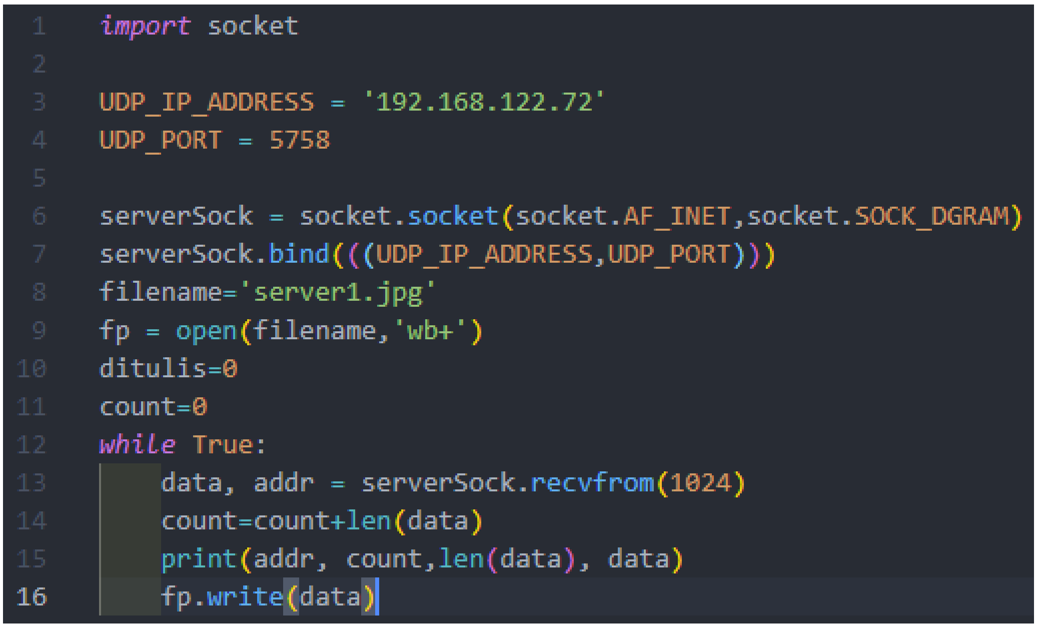
- deskripsi kasus yang dibuat
- gambar arsitektur jaringan (dalam simulator GNS3)
- program yang dibuat (1-4)
- hasil outputnya

## Jawab

1. Deskripsi Kasus : Mendownload sebuah file bertipe image menggunakan project GNS3 yang memiliki 3 alpine dimana Alpine-1 dan Alpine-2 sebagai server dan Alpine-3 akan digunakan sebagai client. File pdf kemudian akan dikirimkan kepada server menggunakan protokol transport UDP. Pengiriman dari client menuju server menggunakan 4 program yang berisikan tentang multi process, multi thread, multi process, asynchronous, multi thread asynchronous.

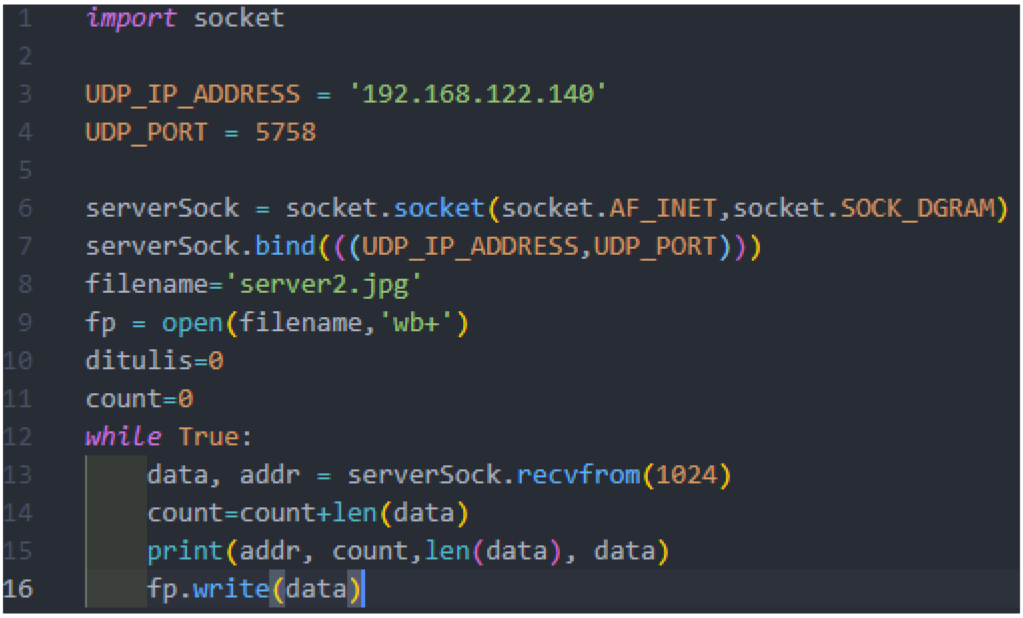
2. Menambah file server1.py dan file server2.py pada folder progjar3 dan mengarahkan IP Address masing-masing alpine1 untuk server 1 dan alpine2 untuk server2

- server1.py



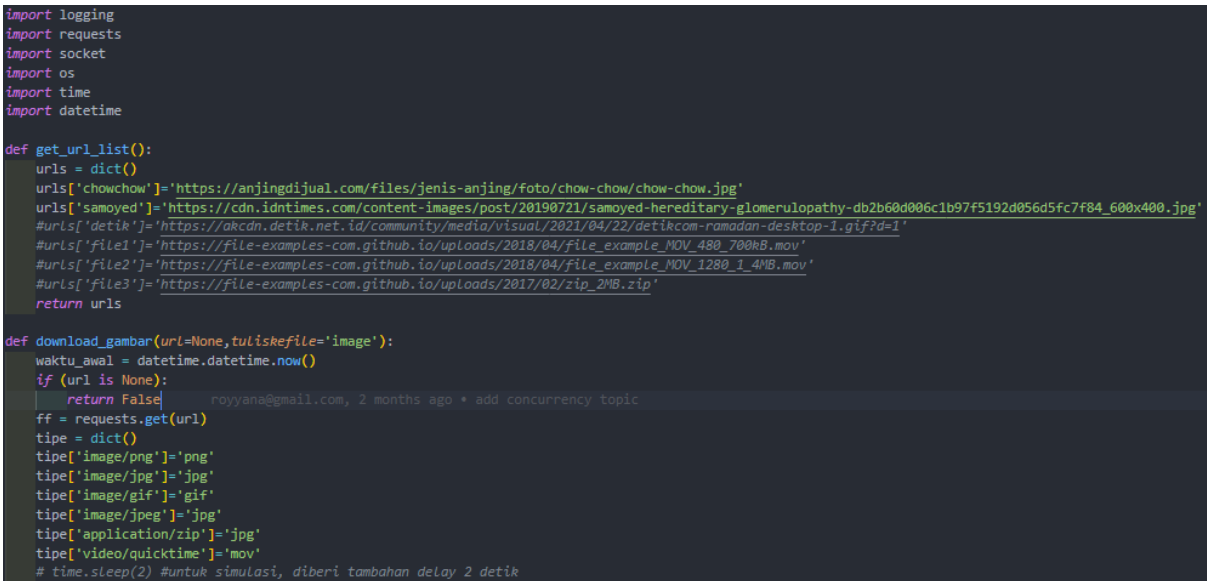
```
import socket

UDP_IP_ADDRESS = '192.168.122.72'
UDP_PORT = 5758

serverSock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
serverSock.bind(((UDP_IP_ADDRESS,UDP_PORT)))
filename='server1.jpg'
fp = open(filename,'wb+')
ditulis=0
count=0
while True:
    data, addr = serverSock.recvfrom(1024)
    count=count+len(data)
    print(addr, count,len(data), data)
    fp.write(data)
```

- server2.py

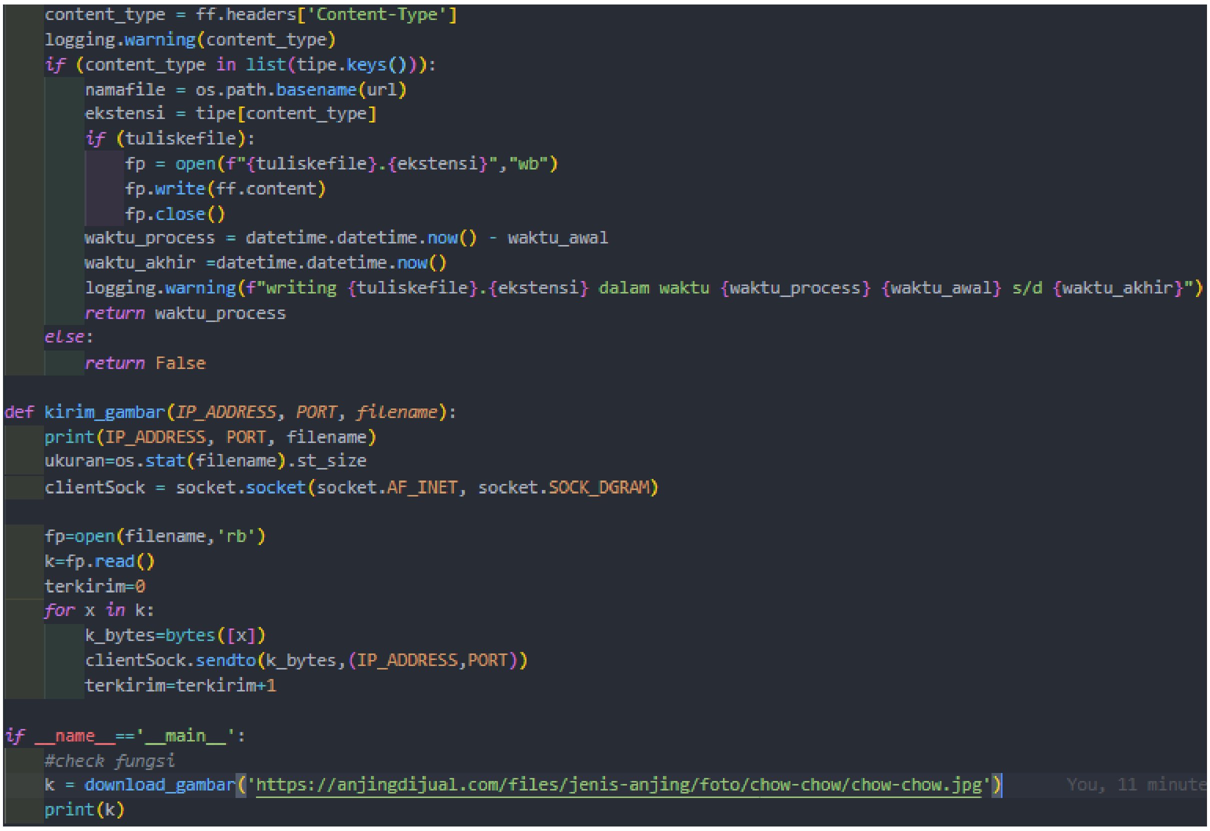


```python
import socket

UDP_IP_ADDRESS = '192.168.122.140'
UDP_PORT = 5758

serverSock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
serverSock.bind(((UDP_IP_ADDRESS,UDP_PORT)))
filename='server2.jpg'
fp = open(filename,'wb+')
ditulis=0
count=0
while True:
    data, addr = serverSock.recvfrom(1024)
    count=count+len(data)
    print(addr, count,len(data), data)
    fp.write(data)
```

3. Mengedit file library.py pada folder progjar 3 menjadi seperti berikut.



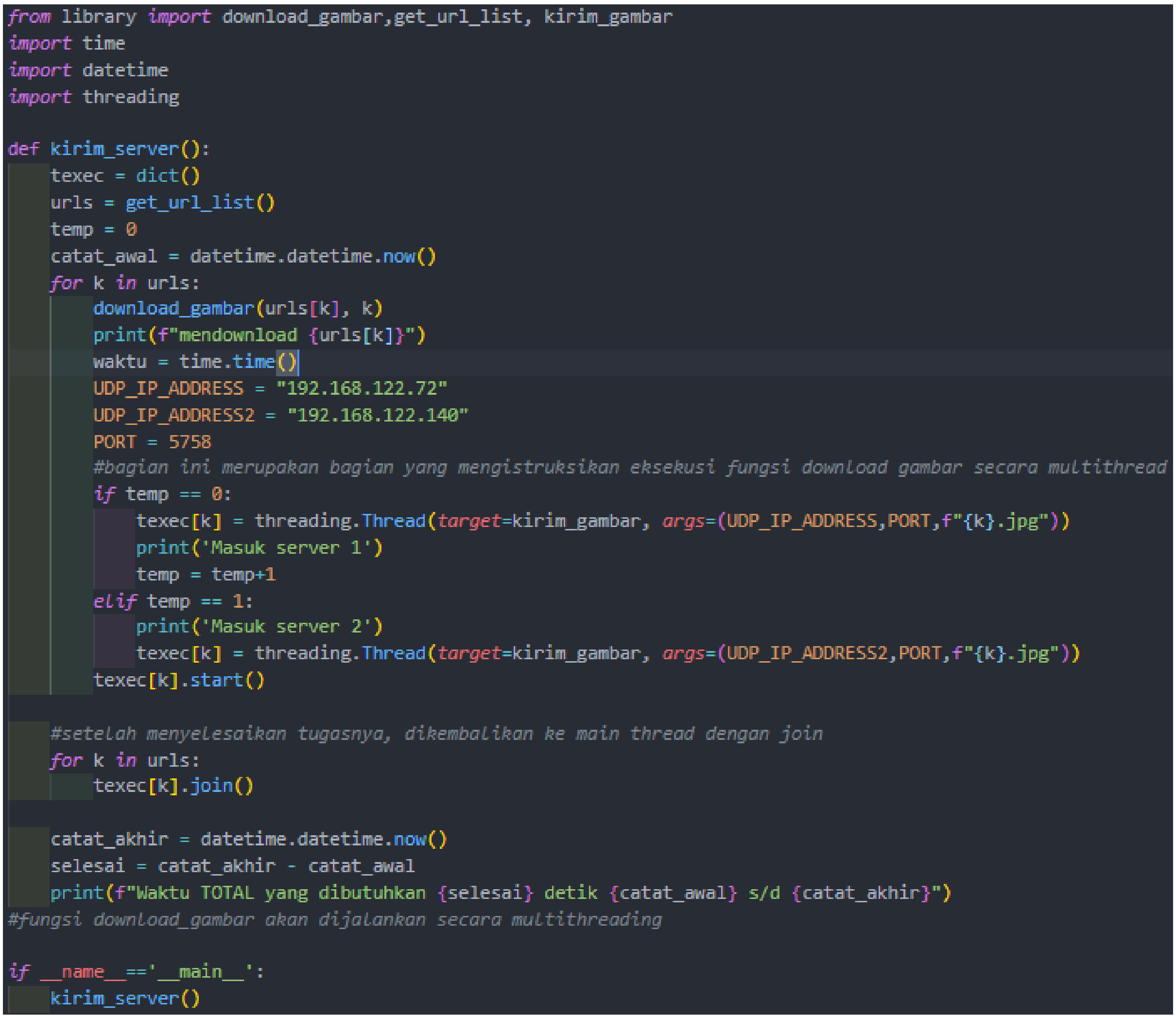
```python
import logging
import requests
import socket
import os
import time
import datetime

def get_url_list():
    urls = dict()
    urls['chowchow']='https://anjingdijual.com/files/jenis-anjing/foto/chow-chow/chow-chow.jpg'
    urls['samoyed']='https://cdn.idntimes.com/content-images/post/20190721/samoyed-hereditary-glomerulopathy-db2b60d006c1b97f5192d056d5fc7f84_600x400.jpg'
    #urls['detik']='https://akcdn.detik.net.id/community/media/visual/2021/04/22/detikcom-ramadan-desktop-1.gif?d=1'
    #urls['file1']='https://file-examples-com.github.io/uploads/2018/04/file_example_MOV_480_700kB.mov'
    #urls['file2']='https://file-examples-com.github.io/uploads/2018/04/file_example_MOV_1280_1_4MB.mov'
    #urls['file3']='https://file-examples-com.github.io/uploads/2017/02/zip_2MB.zip'
    return urls

def download_gambar(url=None,tuliskefile='image'):
    waktu_awal = datetime.datetime.now()
    if (url is None):
        return False      royyana@gmail.com, 2 months ago • add concurrency topic
    ff = requests.get(url)
    tipe = dict()
    tipe['image/png']='png'
    tipe['image/jpg']='jpg'
    tipe['image/gif']='gif'
    tipe['image/jpeg']='jpg'
    tipe['application/zip']='jpg'
    tipe['video/quicktime']='mov'
    # time.sleep(2) #untuk simulasi, diberi tambahan delay 2 detik
```



```python
    content_type = ff.headers['Content-Type']
    logging.warning(content_type)
    if (content_type in list(tipe.keys())):
        namafile = os.path.basename(url)
        ekstensi = tipe[content_type]
        if (tuliskefile):
            fp = open(f"{tuliskefile}.{ekstensi}","wb")
            fp.write(ff.content)
            fp.close()
        waktu_process = datetime.datetime.now() - waktu_awal
        waktu_akhir =datetime.datetime.now()
        logging.warning(f"writing {tuliskefile}.{ekstensi} dalam waktu {waktu_process} {waktu_awal} s/d {waktu_akhir}")
        return waktu_process
    else:
        return False

def kirim_gambar(IP_ADDRESS, PORT, filename):
    print(IP_ADDRESS, PORT, filename)
    ukuran=os.stat(filename).st_size
    clientSock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    fp=open(filename,'rb')
    k=fp.read()
    terkirim=0
    for x in k:
        k_bytes=bytes([x])
        clientSock.sendto(k_bytes,(IP_ADDRESS,PORT))
        terkirim=terkirim+1

if __name__=='__main__':
    #check fungsi
    k = download_gambar('https://anjingdijual.com/files/jenis-anjing/foto/chow-chow/chow-chow.jpg')        You, 11 minute
    print(k)
```

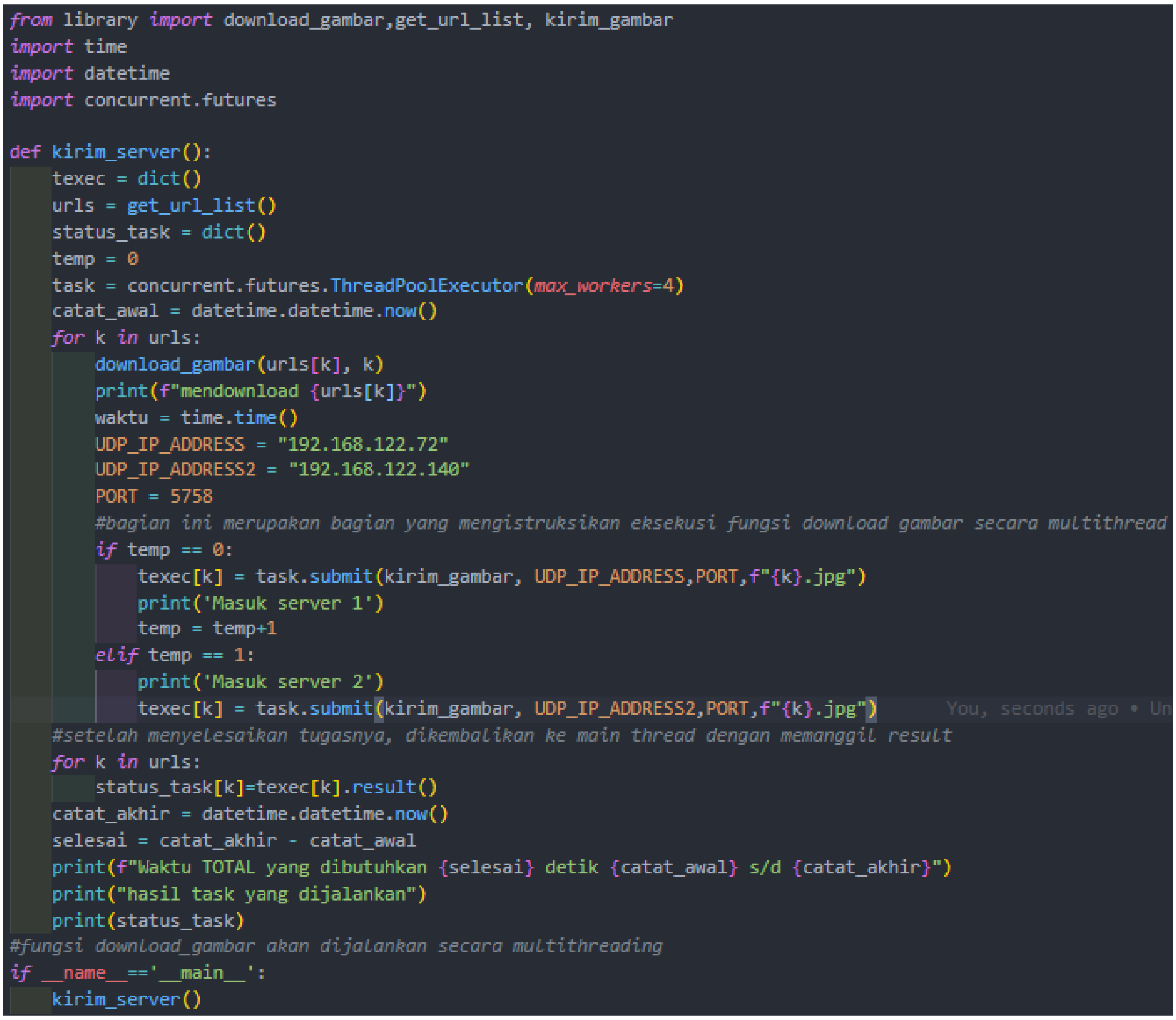
4. Mengedit file multi_thread.py dan multi_thread_async.py pada folder progjar3 menjadi sebagai berikut.

- multi_thread.py



```python
from library import download_gambar,get_url_list, kirim_gambar
import time
import datetime
import threading

def kirim_server():
    texec = dict()
    urls = get_url_list()
    temp = 0
    catat_awal = datetime.datetime.now()
    for k in urls:
        download_gambar(urls[k], k)
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        UDP_IP_ADDRESS = "192.168.122.72"
        UDP_IP_ADDRESS2 = "192.168.122.140"
        PORT = 5758
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi download gambar secara multithread
        if temp == 0:
            texec[k] = threading.Thread(target=kirim_gambar, args=(UDP_IP_ADDRESS,PORT,f"{k}.jpg"))
            print('Masuk server 1')
            temp = temp+1
        elif temp == 1:
            print('Masuk server 2')
            texec[k] = threading.Thread(target=kirim_gambar, args=(UDP_IP_ADDRESS2,PORT,f"{k}.jpg"))
        texec[k].start()

    #setelah menyelesaikan tugasnya, dikembalikan ke main thread dengan join
    for k in urls:
        texec[k].join()

    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
    print(f"Waktu TOTAL yang dibutuhkan {selesai} detik {catat_awal} s/d {catat_akhir}")
#fungsi download_gambar akan dijalankan secara multithreading

if __name__=='__main__':
    kirim_server()
```

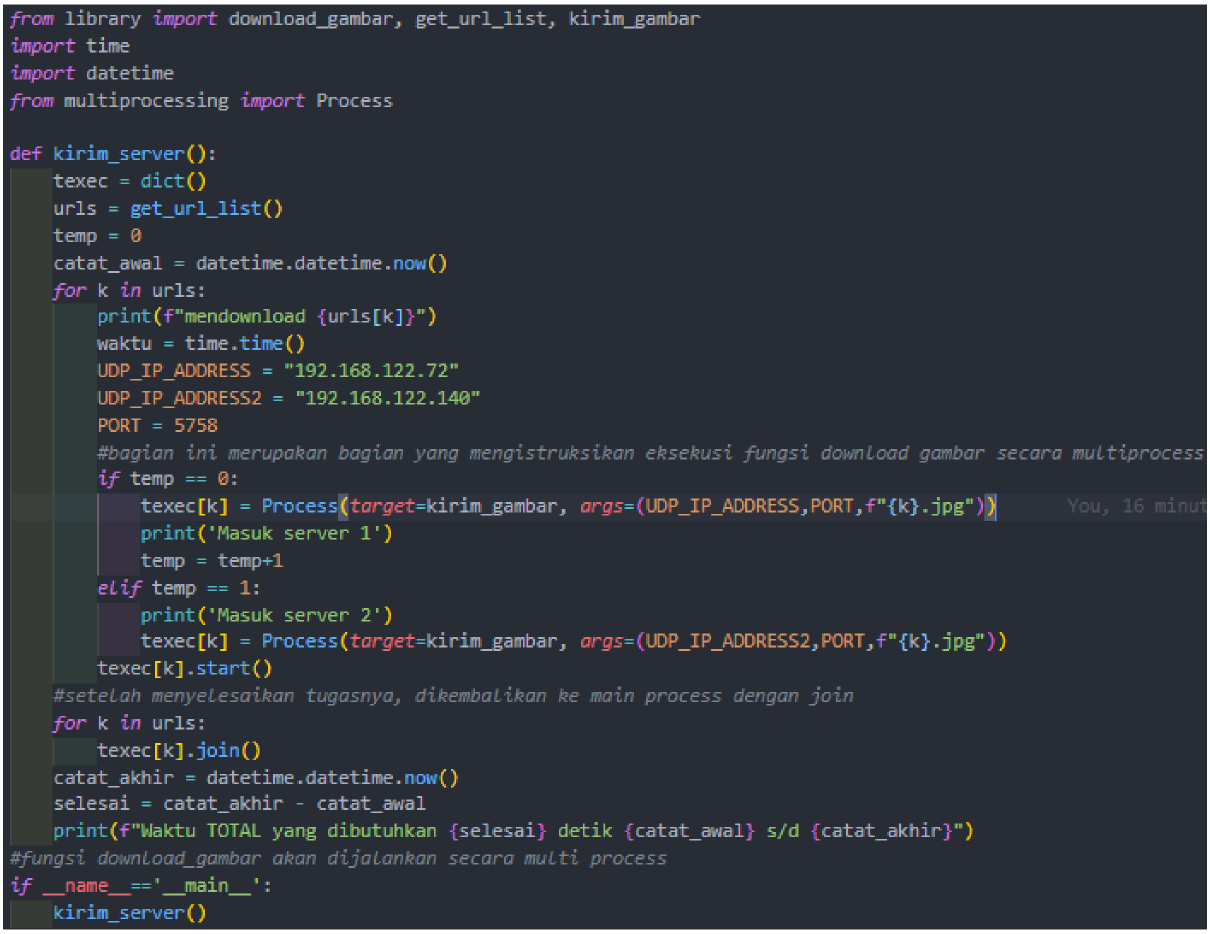
- multi_thread_async.py



```
from library import download_gambar,get_url_list, kirim_gambar
import time
import datetime
import concurrent.futures

def kirim_server():
    texec = dict()
    urls = get_url_list()
    status_task = dict()
    temp = 0
    task = concurrent.futures.ThreadPoolExecutor(max_workers=4)
    catat_awal = datetime.datetime.now()
    for k in urls:
        download_gambar(urls[k], k)
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        UDP_IP_ADDRESS = "192.168.122.72"
        UDP_IP_ADDRESS2 = "192.168.122.140"
        PORT = 5758
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi download gambar secara multithread
        if temp == 0:
            texec[k] = task.submit(kirim_gambar, UDP_IP_ADDRESS,PORT,f"{k}.jpg")
            print('Masuk server 1')
            temp = temp+1
        elif temp == 1:
            print('Masuk server 2')
            texec[k] = task.submit(kirim_gambar, UDP_IP_ADDRESS2,PORT,f"{k}.jpg")      You, seconds ago • Un
    #setelah menyelesaikan tugasnya, dikembalikan ke main thread dengan memanggil result
    for k in urls:
        status_task[k]=texec[k].result()
    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
    print(f"Waktu TOTAL yang dibutuhkan {selesai} detik {catat_awal} s/d {catat_akhir}")
    print("hasil task yang dijalankan")
    print(status_task)
#fungsi download_gambar akan dijalankan secara multithreading
if __name__=='__main__':
    kirim_server()
```

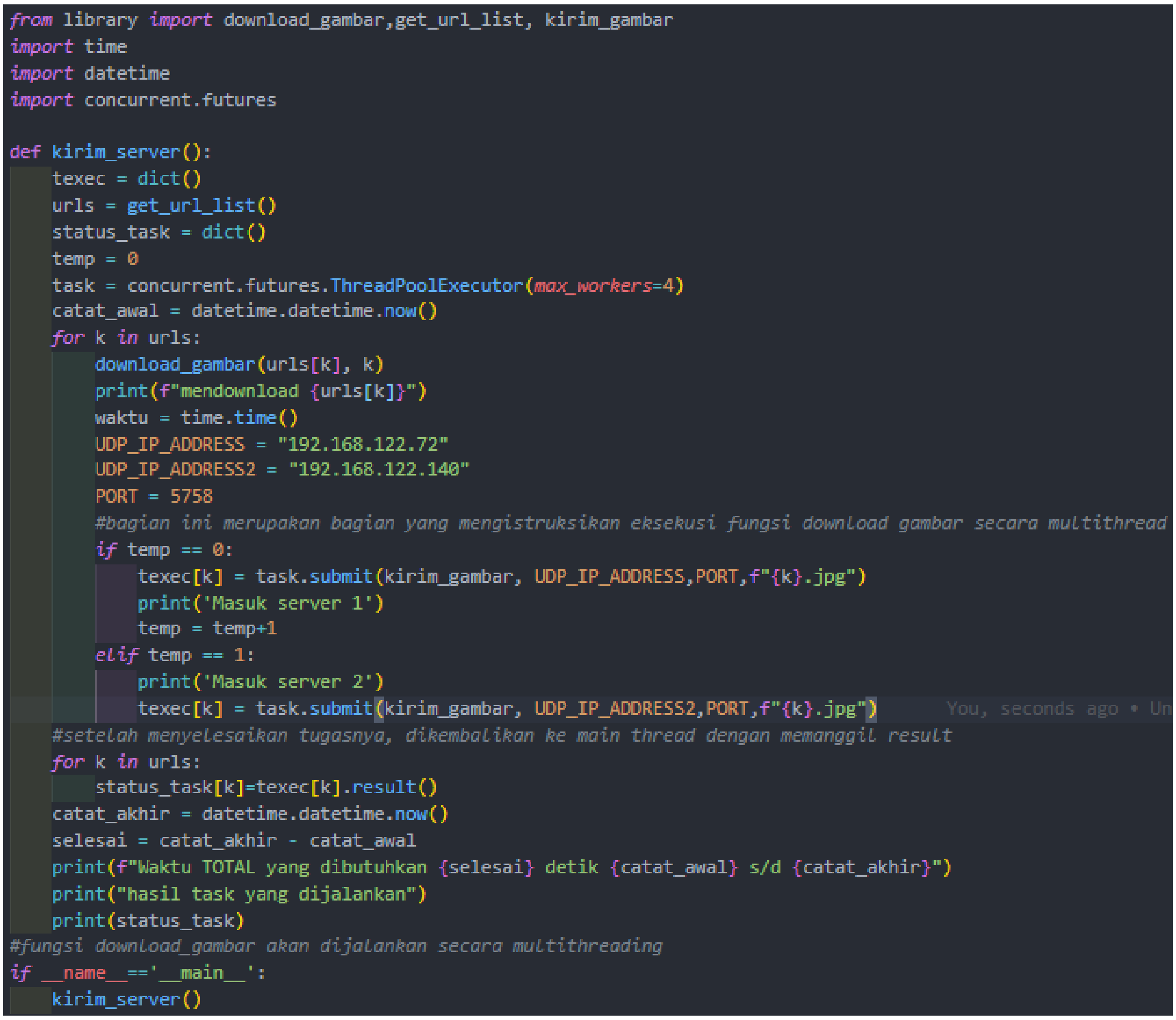
5. Mengedit file multi_process.py dan multi_process_async.py pada folder progjar3 menjadi sebagai berikut.

- multi_process.py



```
from library import download_gambar, get_url_list, kirim_gambar
import time
import datetime
from multiprocessing import Process

def kirim_server():
    texec = dict()
    urls = get_url_list()
    temp = 0
    catat_awal = datetime.datetime.now()
    for k in urls:
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        UDP_IP_ADDRESS = "192.168.122.72"
        UDP_IP_ADDRESS2 = "192.168.122.140"
        PORT = 5758
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi download gambar secara multiprocess
        if temp == 0:
            texec[k] = Process(target=kirim_gambar, args=(UDP_IP_ADDRESS,PORT,f"{k}.jpg"))       You, 16 minut
            print('Masuk server 1')
            temp = temp+1
        elif temp == 1:
            print('Masuk server 2')
            texec[k] = Process(target=kirim_gambar, args=(UDP_IP_ADDRESS2,PORT,f"{k}.jpg"))
        texec[k].start()
    #setelah menyelesaikan tugasnya, dikembalikan ke main process dengan join
    for k in urls:
        texec[k].join()
    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
    print(f"Waktu TOTAL yang dibutuhkan {selesai} detik {catat_awal} s/d {catat_akhir}")
#fungsi download_gambar akan dijalankan secara multi process
if __name__=='__main__':
    kirim_server()
```

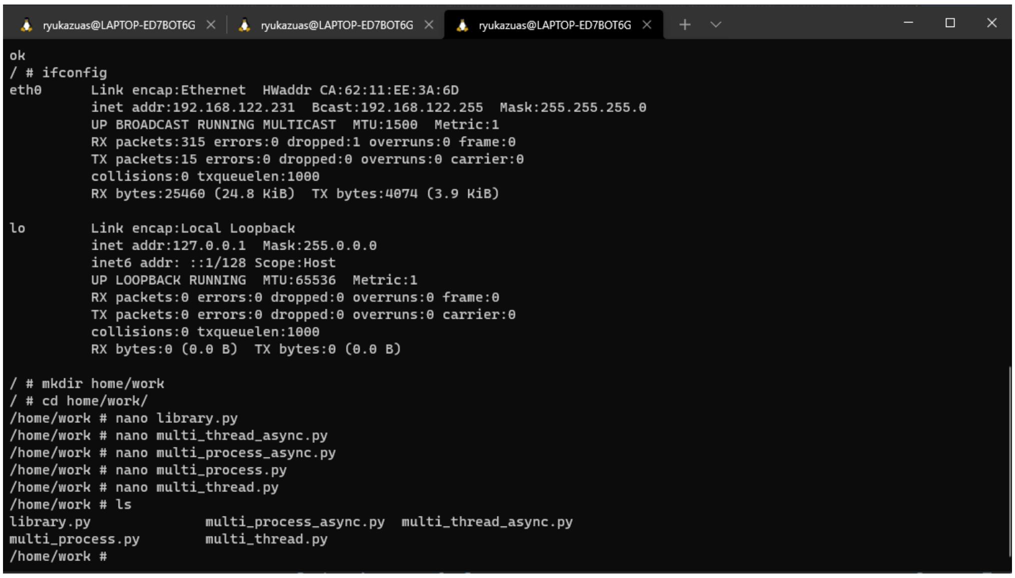
- multi_process_async.py



```python
from library import download_gambar,get_url_list, kirim_gambar
import time
import datetime
import concurrent.futures

def kirim_server():
    texec = dict()
    urls = get_url_list()
    status_task = dict()
    temp = 0
    task = concurrent.futures.ThreadPoolExecutor(max_workers=4)
    catat_awal = datetime.datetime.now()
    for k in urls:
        download_gambar(urls[k], k)
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        UDP_IP_ADDRESS = "192.168.122.72"
        UDP_IP_ADDRESS2 = "192.168.122.140"
        PORT = 5758
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi download gambar secara multithread
        if temp == 0:
            texec[k] = task.submit(kirim_gambar, UDP_IP_ADDRESS,PORT,f"{k}.jpg")
            print('Masuk server 1')
            temp = temp+1
        elif temp == 1:
            print('Masuk server 2')
            texec[k] = task.submit(kirim_gambar, UDP_IP_ADDRESS2,PORT,f"{k}.jpg")      You, seconds ago • Un
    #setelah menyelesaikan tugasnya, dikembalikan ke main thread dengan memanggil result
    for k in urls:
        status_task[k]=texec[k].result()
    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
    print(f"Waktu TOTAL yang dibutuhkan {selesai} detik {catat_awal} s/d {catat_akhir}")
    print("hasil task yang dijalankan")
    print(status_task)
#fungsi download_gambar akan dijalankan secara multithreading
if __name__=='__main__':
    kirim_server()
```

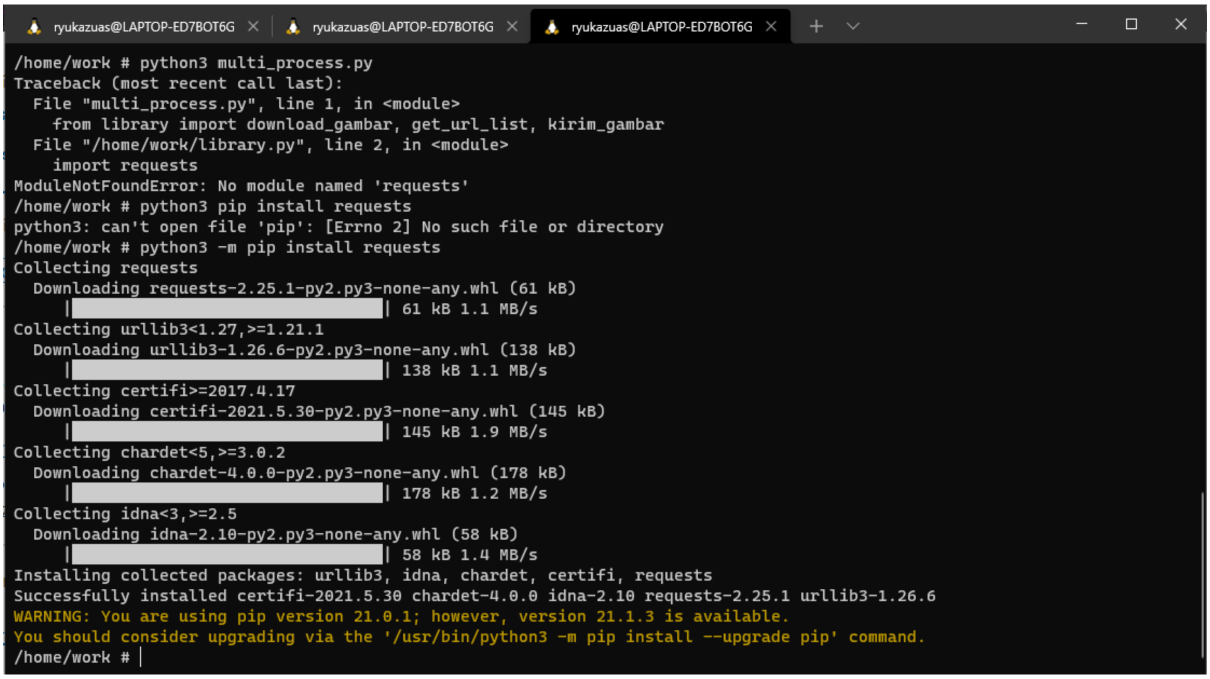
6. Masukkan file server1.py ke alpine 1 dan file server2.py ke alpine 2

7. Masukkan file library.py, multi_process.py, multi_process_async.py, multi_thread.py, dan multi_thread_async.py pada client (alpine 3)


ryukazuas@LAPTOP-ED7BOT6G
ryukazuas@LAPTOP-ED7BOT6G
ryukazuas@LAPTOP-ED7BOT6G

```
ok
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr CA:62:11:EE:3A:6D
          inet addr:192.168.122.231  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:315 errors:0 dropped:1 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:25460 (24.8 KiB)  TX bytes:4074 (3.9 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/ # mkdir home/work
/ # cd home/work/
/home/work # nano library.py
/home/work # nano multi_thread_async.py
/home/work # nano multi_process_async.py
/home/work # nano multi_process.py
/home/work # nano multi_thread.py
/home/work # ls
library.py             multi_process_async.py  multi_thread_async.py
multi_process.py       multi_thread.py
/home/work #
```

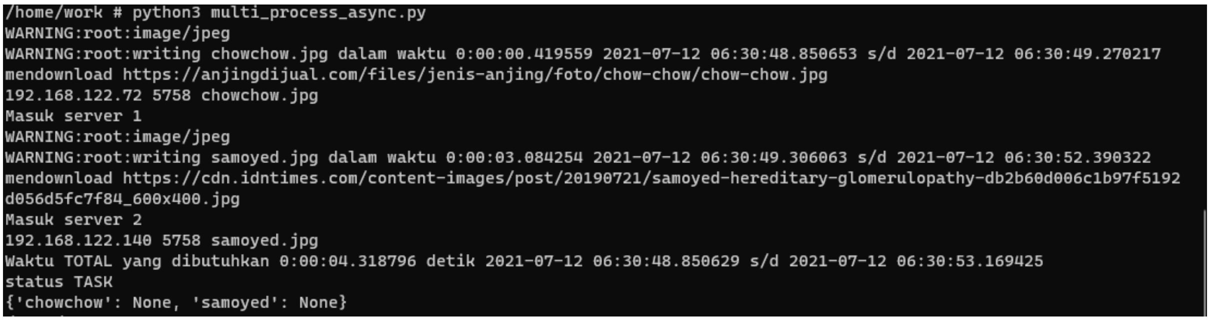
8. Jalankan file server1.py dan server2.py lalu install module requests pada client


ryukazuas@LAPTOP-ED7BOT6G
ryukazuas@LAPTOP-ED7BOT6G
ryukazuas@LAPTOP-ED7BOT6G

```
/home/work # python3 multi_process.py
Traceback (most recent call last):
  File "multi_process.py", line 1, in <module>
    from library import download_gambar, get_url_list, kirim_gambar
  File "/home/work/library.py", line 2, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
/home/work # python3 pip install requests
python3: can't open file 'pip': [Errno 2] No such file or directory
/home/work # python3 -m pip install requests
Collecting requests
  Downloading requests-2.25.1-py2.py3-none-any.whl (61 kB)
     |████████████████████████████████| 61 kB 1.1 MB/s
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.6-py2.py3-none-any.whl (138 kB)
     |████████████████████████████████| 138 kB 1.1 MB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2021.5.30-py2.py3-none-any.whl (145 kB)
     |████████████████████████████████| 145 kB 1.9 MB/s
Collecting chardet<5,>=3.0.2
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
     |████████████████████████████████| 178 kB 1.2 MB/s
Collecting idna<3,>=2.5
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
     |████████████████████████████████| 58 kB 1.4 MB/s
Installing collected packages: urllib3, idna, chardet, certifi, requests
Successfully installed certifi-2021.5.30 chardet-4.0.0 idna-2.10 requests-2.25.1 urllib3-1.26.6
WARNING: You are using pip version 21.0.1; however, version 21.1.3 is available.
You should consider upgrading via the '/usr/bin/python3 -m pip install --upgrade pip' command.
/home/work # |
```

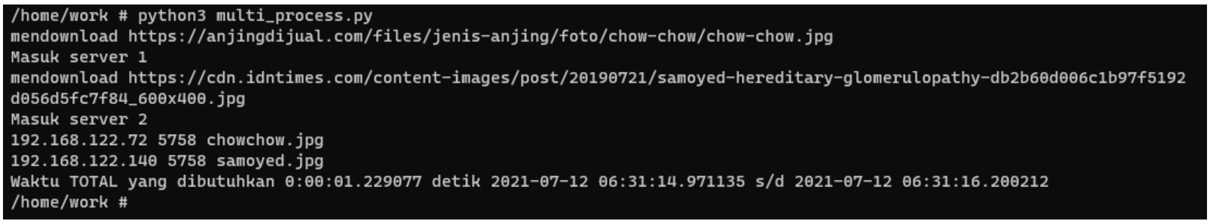
9. Jalankan file multi_process_async.py terlebih dahulu kemudian menjalankan file multi_process.py

- multi_process_async.py



```
/home/work # python3 multi_process_async.py
WARNING:root:image/jpeg
WARNING:root:writing chowchow.jpg dalam waktu 0:00:00.419559 2021-07-12 06:30:48.850653 s/d 2021-07-12 06:30:49.270217
mendownload https://anjingdijual.com/files/jenis-anjing/foto/chow-chow/chow-chow.jpg
192.168.122.72 5758 chowchow.jpg
Masuk server 1
WARNING:root:image/jpeg
WARNING:root:writing samoyed.jpg dalam waktu 0:00:03.084254 2021-07-12 06:30:49.306063 s/d 2021-07-12 06:30:52.390322
mendownload https://cdn.idntimes.com/content-images/post/20190721/samoyed-hereditary-glomerulopathy-db2b60d006c1b97f5192
d056d5fc7f84_600x400.jpg
Masuk server 2
192.168.122.140 5758 samoyed.jpg
Waktu TOTAL yang dibutuhkan 0:00:04.318796 detik 2021-07-12 06:30:48.850629 s/d 2021-07-12 06:30:53.169425
status TASK
{'chowchow': None, 'samoyed': None}
```

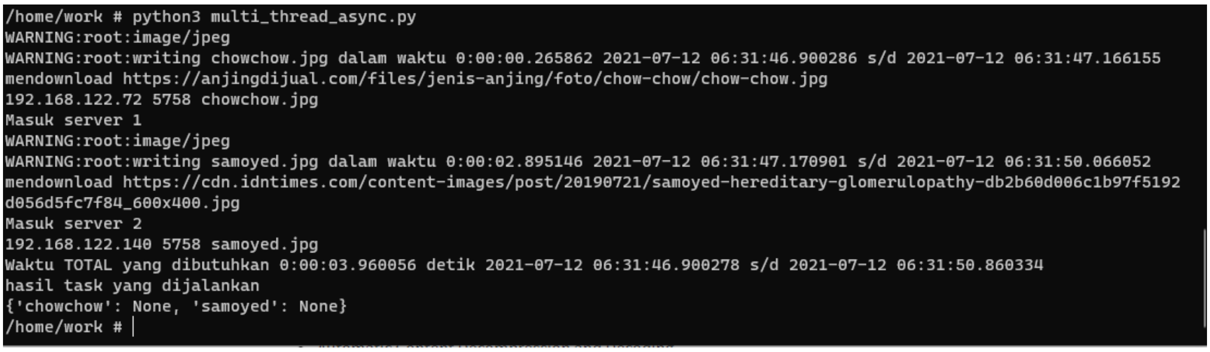
- multi_process.py



```
/home/work # python3 multi_process.py
mendownload https://anjingdijual.com/files/jenis-anjing/foto/chow-chow/chow-chow.jpg
Masuk server 1
mendownload https://cdn.idntimes.com/content-images/post/20190721/samoyed-hereditary-glomerulopathy-db2b60d006c1b97f5192
d056d5fc7f84_600x400.jpg
Masuk server 2
192.168.122.72 5758 chowchow.jpg
192.168.122.140 5758 samoyed.jpg
Waktu TOTAL yang dibutuhkan 0:00:01.229077 detik 2021-07-12 06:31:14.971135 s/d 2021-07-12 06:31:16.200212
/home/work #
```

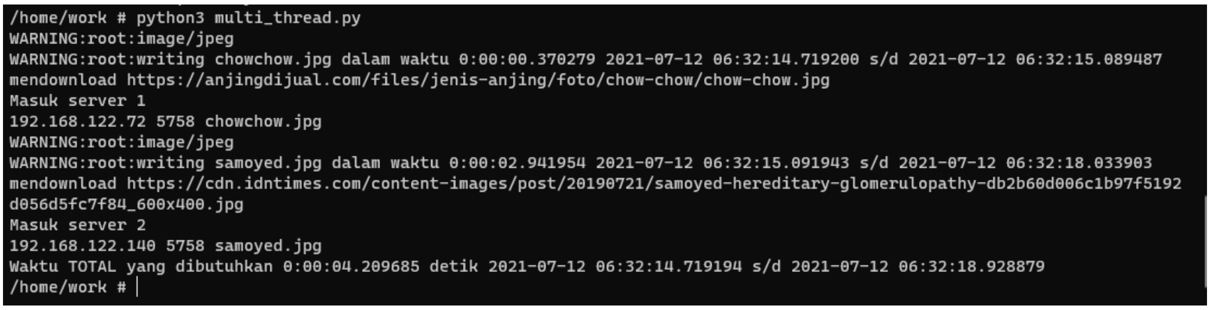
10. Jalankan file multi_thread_async.py terlebih dahulu kemudian menjalankan file multi_thread.py

- multi_thread_async.py



```
/home/work # python3 multi_thread_async.py
WARNING:root:image/jpeg
WARNING:root:writing chowchow.jpg dalam waktu 0:00:00.265862 2021-07-12 06:31:46.900286 s/d 2021-07-12 06:31:47.166155
mendownload https://anjingdijual.com/files/jenis-anjing/foto/chow-chow/chow-chow.jpg
192.168.122.72 5758 chowchow.jpg
Masuk server 1
WARNING:root:image/jpeg
WARNING:root:writing samoyed.jpg dalam waktu 0:00:02.895146 2021-07-12 06:31:47.170901 s/d 2021-07-12 06:31:50.066052
mendownload https://cdn.idntimes.com/content-images/post/20190721/samoyed-hereditary-glomerulopathy-db2b60d006c1b97f5192
d056d5fc7f84_600x400.jpg
Masuk server 2
192.168.122.140 5758 samoyed.jpg
Waktu TOTAL yang dibutuhkan 0:00:03.960056 detik 2021-07-12 06:31:46.900278 s/d 2021-07-12 06:31:50.860334
hasil task yang dijalankan
{'chowchow': None, 'samoyed': None}
/home/work # |
```

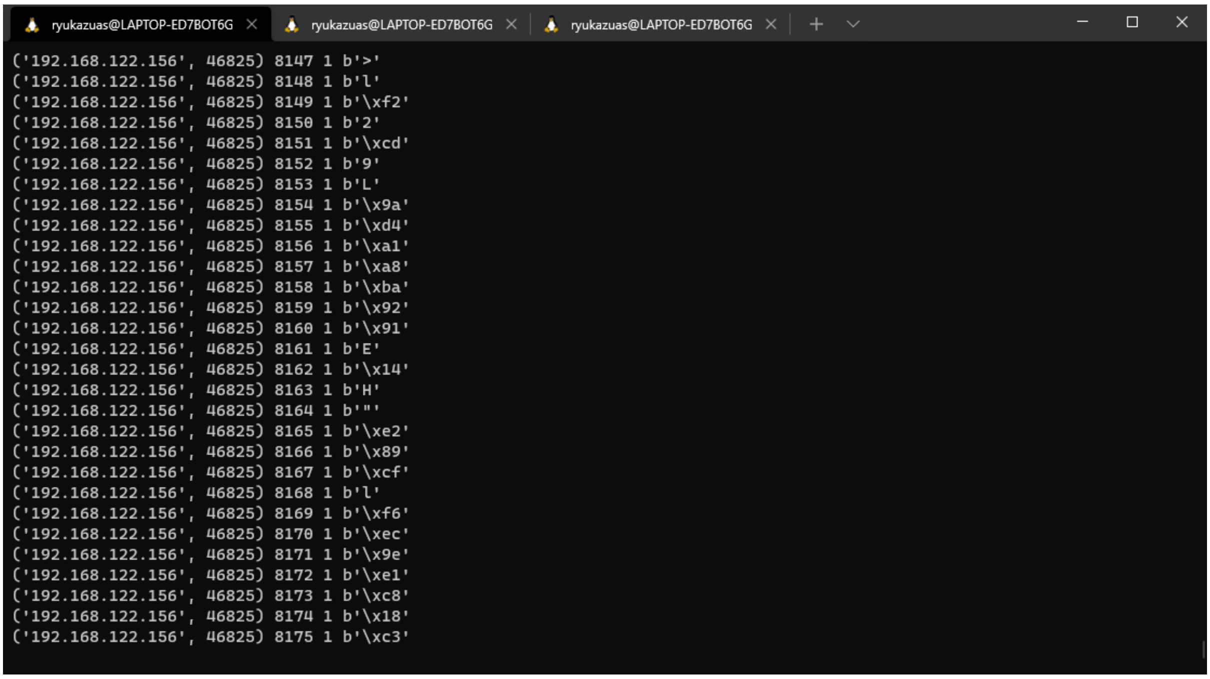
- multi_thread.py



```
/home/work # python3 multi_thread.py
WARNING:root:image/jpeg
WARNING:root:writing chowchow.jpg dalam waktu 0:00:00.370279 2021-07-12 06:32:14.719200 s/d 2021-07-12 06:32:15.089487
mendownload https://anjingdijual.com/files/jenis-anjing/foto/chow-chow/chow-chow.jpg
Masuk server 1
192.168.122.72 5758 chowchow.jpg
WARNING:root:image/jpeg
WARNING:root:writing samoyed.jpg dalam waktu 0:00:02.941954 2021-07-12 06:32:15.091943 s/d 2021-07-12 06:32:18.033903
mendownload https://cdn.idntimes.com/content-images/post/20190721/samoyed-hereditary-glomerulopathy-db2b60d006c1b97f5192
d056d5fc7f84_600x400.jpg
Masuk server 2
192.168.122.140 5758 samoyed.jpg
Waktu TOTAL yang dibutuhkan 0:00:04.209685 detik 2021-07-12 06:32:14.719194 s/d 2021-07-12 06:32:18.928879
/home/work # |
```

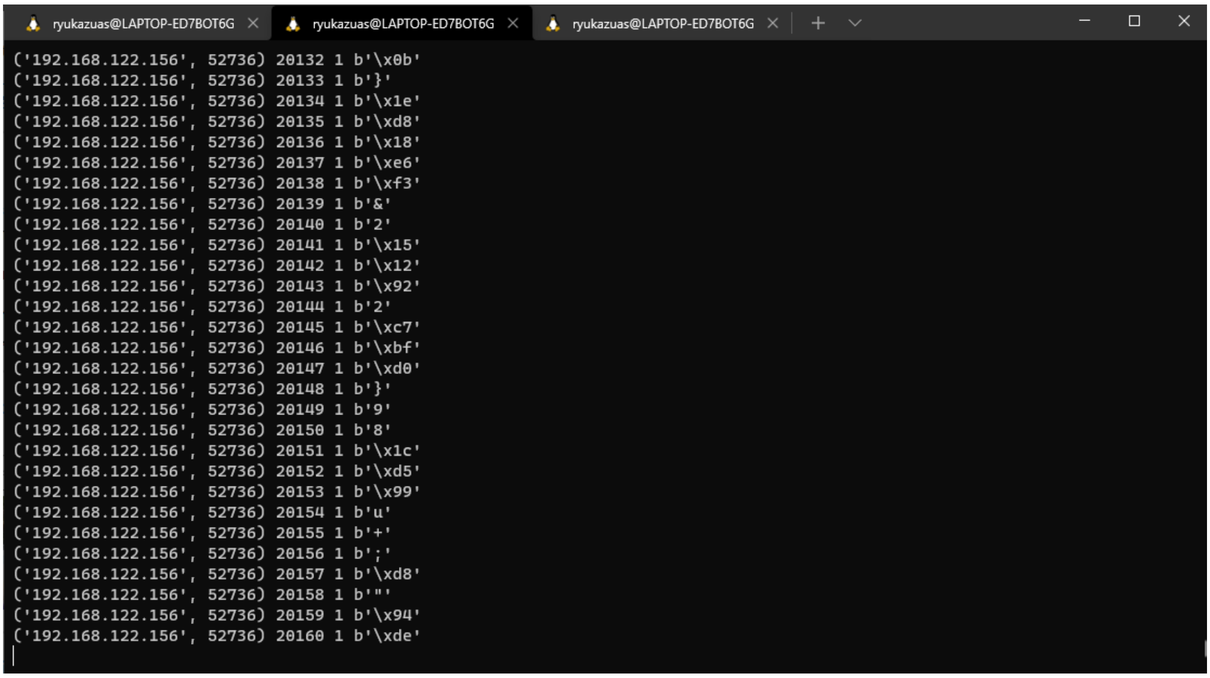
11. Pada server 1 dan server 2 akan tampak hasil seperti berikut:

- server 1


ryukazuas@LAPTOP-ED7BOT6G
ryukazuas@LAPTOP-ED7BOT6G
ryukazuas@LAPTOP-ED7BOT6G

```
('192.168.122.156', 46825) 8147 1 b'>'
('192.168.122.156', 46825) 8148 1 b'l'
('192.168.122.156', 46825) 8149 1 b'\xf2'
('192.168.122.156', 46825) 8150 1 b'2'
('192.168.122.156', 46825) 8151 1 b'\xcd'
('192.168.122.156', 46825) 8152 1 b'9'
('192.168.122.156', 46825) 8153 1 b'L'
('192.168.122.156', 46825) 8154 1 b'\x9a'
('192.168.122.156', 46825) 8155 1 b'\xd4'
('192.168.122.156', 46825) 8156 1 b'\xa1'
('192.168.122.156', 46825) 8157 1 b'\xa8'
('192.168.122.156', 46825) 8158 1 b'\xba'
('192.168.122.156', 46825) 8159 1 b'\x92'
('192.168.122.156', 46825) 8160 1 b'\x91'
('192.168.122.156', 46825) 8161 1 b'E'
('192.168.122.156', 46825) 8162 1 b'\x14'
('192.168.122.156', 46825) 8163 1 b'H'
('192.168.122.156', 46825) 8164 1 b'"'
('192.168.122.156', 46825) 8165 1 b'\xe2'
('192.168.122.156', 46825) 8166 1 b'\x89'
('192.168.122.156', 46825) 8167 1 b'\xcf'
('192.168.122.156', 46825) 8168 1 b'l'
('192.168.122.156', 46825) 8169 1 b'\xf6'
('192.168.122.156', 46825) 8170 1 b'\xec'
('192.168.122.156', 46825) 8171 1 b'\x9e'
('192.168.122.156', 46825) 8172 1 b'\xe1'
('192.168.122.156', 46825) 8173 1 b'\xc8'
('192.168.122.156', 46825) 8174 1 b'\x18'
('192.168.122.156', 46825) 8175 1 b'\xc3'
```

- server 2



ryukazuas@LAPTOP-ED7BOT6G
ryukazuas@LAPTOP-ED7BOT6G
ryukazuas@LAPTOP-ED7BOT6G

```
('192.168.122.156', 52736) 20132 1 b'\x0b'
('192.168.122.156', 52736) 20133 1 b'}'
('192.168.122.156', 52736) 20134 1 b'\x1e'
('192.168.122.156', 52736) 20135 1 b'\xd8'
('192.168.122.156', 52736) 20136 1 b'\x18'
('192.168.122.156', 52736) 20137 1 b'\xe6'
('192.168.122.156', 52736) 20138 1 b'\xf3'
('192.168.122.156', 52736) 20139 1 b'&'
('192.168.122.156', 52736) 20140 1 b'2'
('192.168.122.156', 52736) 20141 1 b'\x15'
('192.168.122.156', 52736) 20142 1 b'\x12'
('192.168.122.156', 52736) 20143 1 b'\x92'
('192.168.122.156', 52736) 20144 1 b'2'
('192.168.122.156', 52736) 20145 1 b'\xc7'
('192.168.122.156', 52736) 20146 1 b'\xbf'
('192.168.122.156', 52736) 20147 1 b'\xd0'
('192.168.122.156', 52736) 20148 1 b'}'
('192.168.122.156', 52736) 20149 1 b'9'
('192.168.122.156', 52736) 20150 1 b'8'
('192.168.122.156', 52736) 20151 1 b'\x1c'
('192.168.122.156', 52736) 20152 1 b'\xd5'
('192.168.122.156', 52736) 20153 1 b'\x99'
('192.168.122.156', 52736) 20154 1 b'u'
('192.168.122.156', 52736) 20155 1 b'+'
('192.168.122.156', 52736) 20156 1 b';'
('192.168.122.156', 52736) 20157 1 b'\xd8'
('192.168.122.156', 52736) 20158 1 b'"'
('192.168.122.156', 52736) 20159 1 b'\x94'
('192.168.122.156', 52736) 20160 1 b'\xde'
```