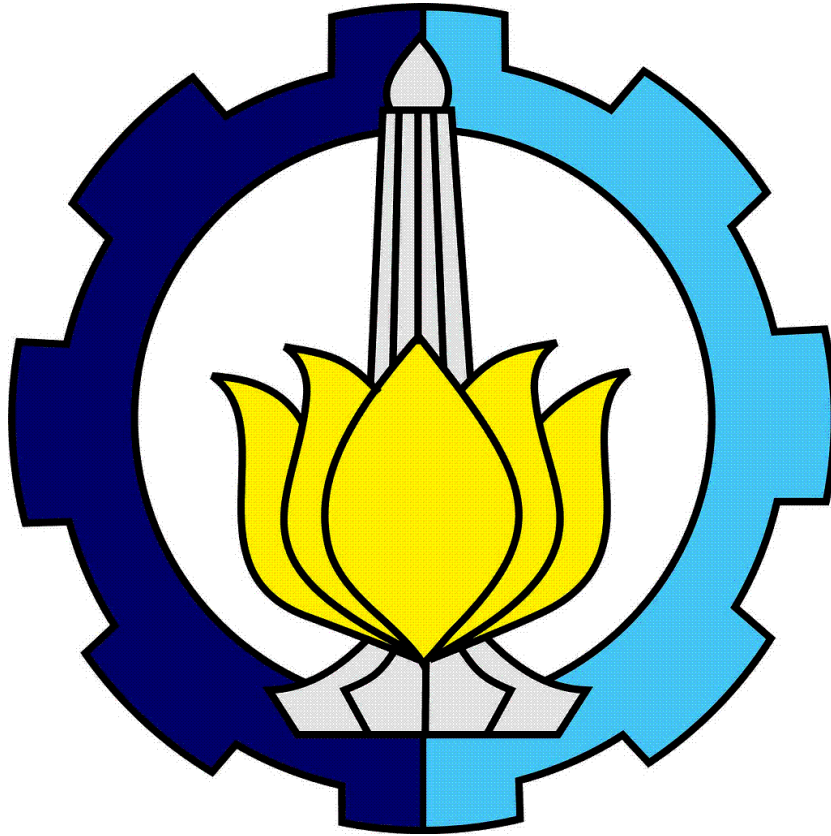


Dokumentasi
Tugas Implementasi Concurrency



Pemrograman Jaringan
Kelas E

Afia Hana Yusriya

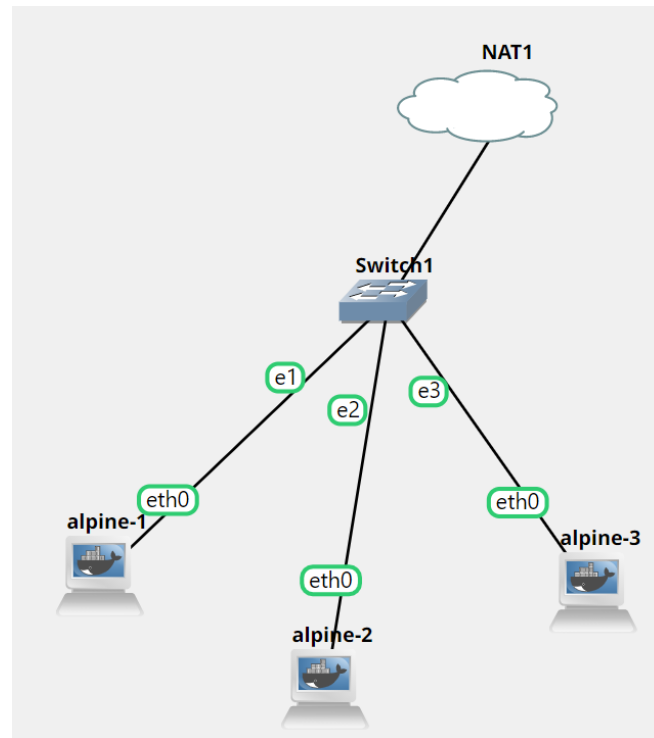
05111840000111

Institut Teknologi Sepuluh Nopember
Surabaya
2021

A. Deskripsi Kasus

Kasus yang diangkat adalah mendownload sebuah file dengan tipe image menggunakan GNS3 Project dengan 3 alpine. 2 alpine akan bertindak sebagai server dan 1 alpine sebagai client.

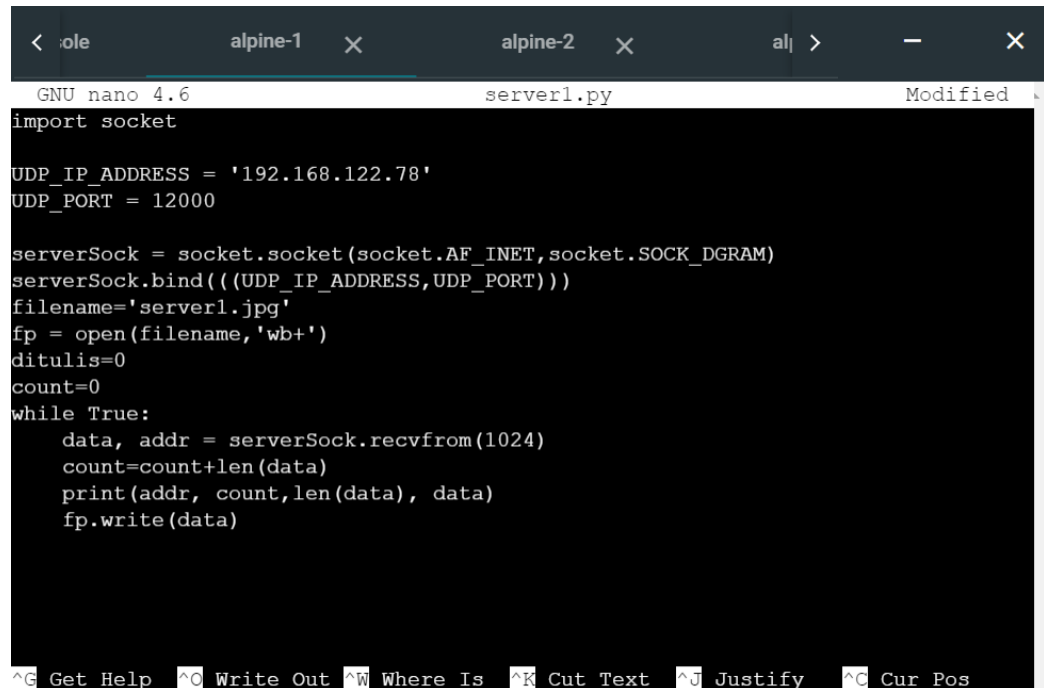
B. Gambar Arsitektur Jaringan



C. Program yang Dibuat

1. Pada alpine-1, alpine-2, dan alpine-3 clone repository yang sudah di fork dengan command **git clone** <https://github.com/afiahana/progjar>
2. Pada alpine-1, alpine-2, dan alpine-3 jalankan command **ifconfig** untuk mendapatkan IP address masing-masing alpine. Didapatkan IP sebagai berikut
Alpine-1 : 192.168.122.78 (server 1)
Alpine-2 : 192.168.122.54 (server 2)
Alpine-3 : 192.168.122.26 (client)
3. Pada masing-masing alpine, masuk ke dalam **folder progjar3**

4. Pada alpine-1, buatlah file baru dengan nama **server1.py** lalu isi dengan kode sebagai berikut (IP disesuaikan dengan IP alpine-1)

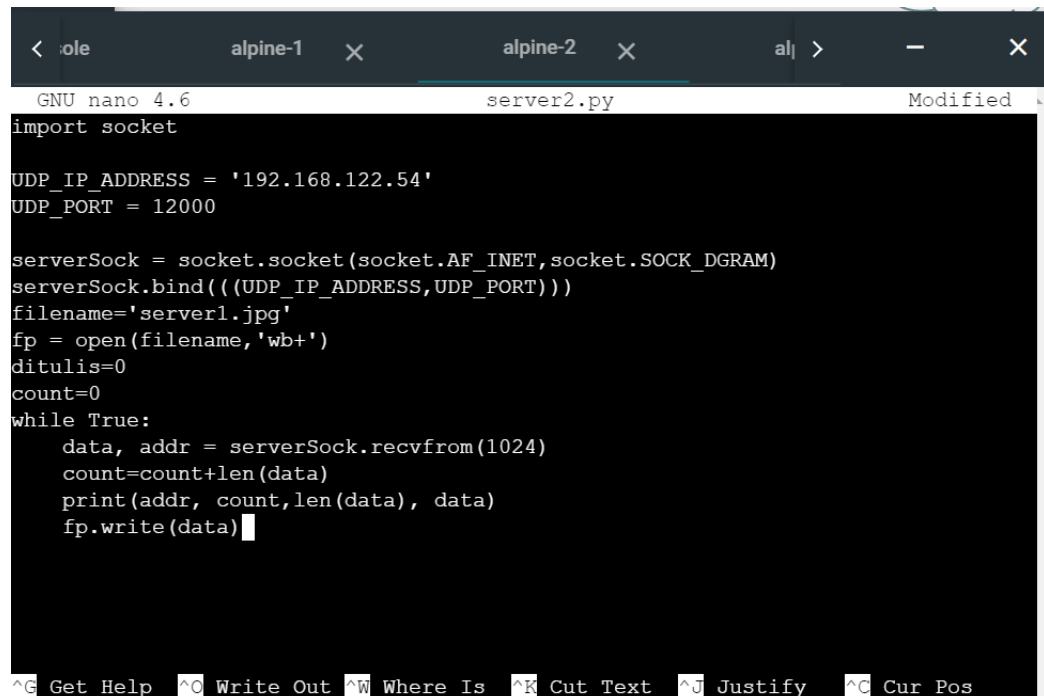


```
GNU nano 4.6 server1.py Modified
import socket

UDP_IP_ADDRESS = '192.168.122.78'
UDP_PORT = 12000

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)
```

5. Pada alpine-2, buatlah file baru dengan nama **server2.py** lalu isi dengan kode sebagai berikut (IP disesuaikan dengan IP alpine-2)



```
GNU nano 4.6 server2.py Modified
import socket

UDP_IP_ADDRESS = '192.168.122.54'
UDP_PORT = 12000

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)
```

6. Pada alpine-3, modifikasi file **library.py** sebagai berikut

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

def get_url_list():
    urls = dict()
    urls['persia'] = 'https://s2.bukalapak.com/img/7611942961/large/IMG_20170929_WA0026_scaled.jpg'
    urls['anggara'] = 'https://embaran.id/wp-content/uploads/2020/12/Kucing-Persia-Abu-Abu.jpg'
    return urls

def download_gambar(url=None, tuliskefile='image'):
    waktu_awal = datetime.datetime.now()
    if (url is None):
        return False
    ff = requests.get(url)
    tipe = dict()
    tipe['image/png'] = 'png'
    tipe['image/jpeg'] = 'jpg'
    tipe['image/gif'] = 'gif'
    tipe['image/jpeg'] = 'jpg'
    tipe['application/zip'] = 'zip'
    tipe['video/quicktime'] = 'mov'

    # time.sleep(2) #untuk simulasi, diberi tambahan delay 2 detik

    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://s2.bukalapak.com/img/7611942961/large/IMG_20170929_WA0026_scaled.jpg')
    print(k)
```

7. Pada alpine-3, modifikasi file **multi_thread.py** sebagai berikut

```
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 = "192.168.122.78"
        UDP_IP_ADDRESS2 = "192.168.122.54"
        PORT = 12000
        #bagian ini merupakan bagian yang menginstruksikan 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()
```

8. Pada alpine-3, modifikasi file **multi_process.py** sebagai berikut

```
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 = "192.168.122.78"
        UDP_IP_ADDRESS2 = "192.168.122.54"
        PORT = 12000
        #bagian ini merupakan bagian yang menginstruksikan eksekusi fungsi download gambar secara multiprocessing
        if temp == 0:
            texec[k] = Process(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] = 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()
```

9. Pada alpine-3, modifikasi file **multi_thread_async.py** sebagai berikut

```
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 = "192.168.122.78"
        UDP_IP_ADDRESS2 = "192.168.122.54"
        PORT = 12000
        #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")
            #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()
```

10. Pada alpine-3, modifikasi file **multi_process_async.py** sebagai berikut

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

def kirim_server():
    texec = dict()
    urls = get_url_list()
    status_task = dict()
    temp = 0
    task_pool = Pool(processes=20) #2 task yang dapat dikerjakan secara simultan, dapat diset sesuai jumlah core
    catat_awal = datetime.datetime.now()
    for k in urls:
        download_gambar(urls[k],k)
        print(f"mendownload {urls[k]}")
        #bagian ini merupakan bagian yang menginstruksikan eksekusi fungsi download gambar secara multiprocessing
        UDP_IP_ADDRESS = "192.168.122.78"
        UDP_IP_ADDRESS2 = "192.168.122.54"
        PORT = 12000
        if temp == 0:
            texec[k] = task_pool.apply_async(func=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] = task_pool.apply_async(func=kirim_gambar, args=(UDP_IP_ADDRESS2,PORT,f"{k}.jpg"))

    #setelah menyelesaikan tugasnya, dikembalikan ke main process dengan mengambil hasilnya dengan get
    for k in urls:
        status_task[k]=texec[k].get(timeout=10)

    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("status TASK")
    print(status_task)

#fungsi download_gambar akan dijalankan secara multi process

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

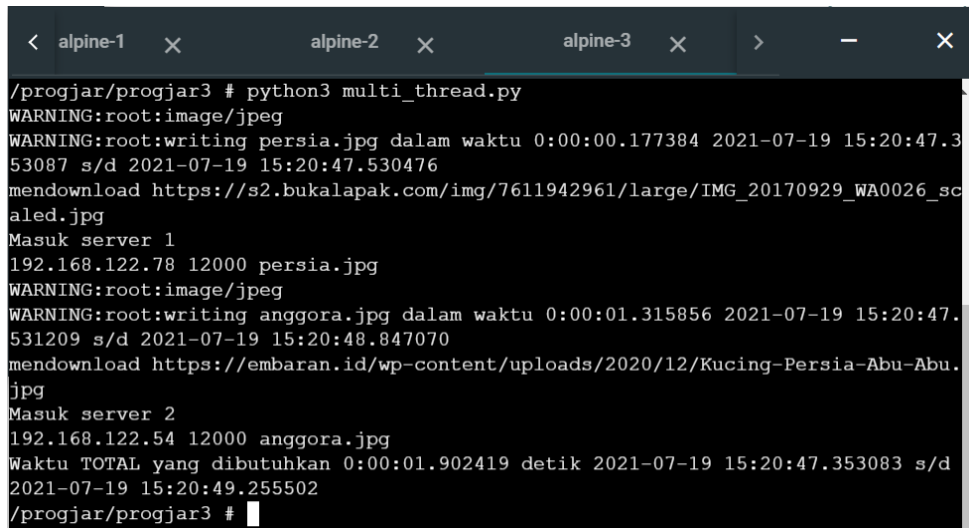
11. Pada alpine-3, install module request dengan command **-m pip install requests**

```
< alpine-1 x alpine-2 x alpine-3 x > - x
/bin/sh: -m: not found
/progjar/progjar3 # pip install requests
Collecting requests
  Downloading requests-2.26.0-py2.py3-none-any.whl (62 kB)
    | 62 kB 306 kB/s
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.6-py2.py3-none-any.whl (138 kB)
    | 138 kB 989 kB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2021.5.30-py2.py3-none-any.whl (145 kB)
    | 145 kB 371 kB/s
Collecting idna<4,>=2.5
  Downloading idna-3.2-py3-none-any.whl (59 kB)
    | 59 kB 439 kB/s
Collecting charset-normalizer~=2.0.0
  Downloading charset-normalizer-2.0.3-py3-none-any.whl (35 kB)
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2021.5.30 charset-normalizer-2.0.3 idna-3.2 requests-2.26.0 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/python3.8 -m pip install --upgrade pip' command.
```

12. Pada alpine-1, jalankan file server1.py dengan command **python3 server1.py**

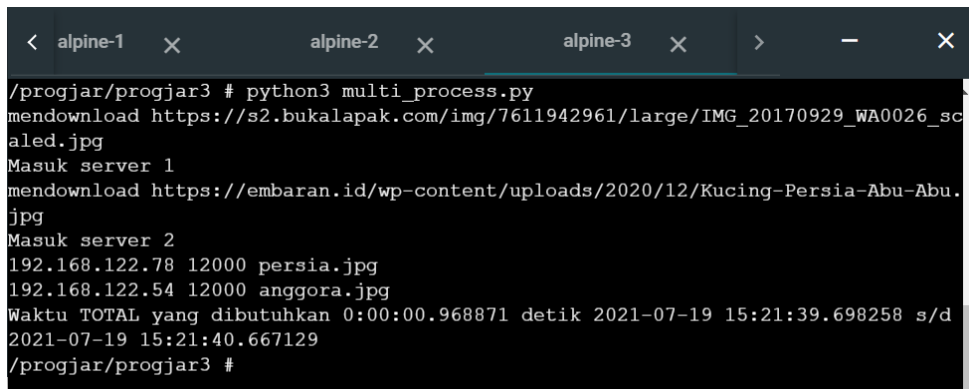
13. Pada alpine-2, jalankan file server2.py dengan command **python3 server2.py**

14. Pada alpine-3, jalankan file multi_thread.py dengan command **python3 multi_thread.py**



```
< alpine-1 x alpine-2 x alpine-3 x > - x
/progjar/progjar3 # python3 multi_thread.py
WARNING:root:image/jpeg
WARNING:root:writing persia.jpg dalam waktu 0:00:00.177384 2021-07-19 15:20:47.3
53087 s/d 2021-07-19 15:20:47.530476
mendownload https://s2.bukalapak.com/img/7611942961/large/IMG_20170929_WA0026_sc
aled.jpg
Masuk server 1
192.168.122.78 12000 persia.jpg
WARNING:root:image/jpeg
WARNING:root:writing anggora.jpg dalam waktu 0:00:01.315856 2021-07-19 15:20:47.
531209 s/d 2021-07-19 15:20:48.847070
mendownload https://embaran.id/wp-content/uploads/2020/12/Kucing-Persia-Abu-Abu.
jpg
Masuk server 2
192.168.122.54 12000 anggora.jpg
Waktu TOTAL yang dibutuhkan 0:00:01.902419 detik 2021-07-19 15:20:47.353083 s/d
2021-07-19 15:20:49.255502
/progjar/progjar3 #
```

15. Pada alpine-3, jalankan file multi_thread.py dengan command **python3 multi_process.py**



```
< alpine-1 x alpine-2 x alpine-3 x > - x
/progjar/progjar3 # python3 multi_process.py
mendownload https://s2.bukalapak.com/img/7611942961/large/IMG_20170929_WA0026_sc
aled.jpg
Masuk server 1
mendownload https://embaran.id/wp-content/uploads/2020/12/Kucing-Persia-Abu-Abu.
jpg
Masuk server 2
192.168.122.78 12000 persia.jpg
192.168.122.54 12000 anggora.jpg
Waktu TOTAL yang dibutuhkan 0:00:00.968871 detik 2021-07-19 15:21:39.698258 s/d
2021-07-19 15:21:40.667129
/progjar/progjar3 #
```


16. Pada alpine-3, jalankan file `multi_thread.py` dengan command **python3 multi_thread_async.py**

```
< alpine-1 x alpine-2 x alpine-3 x > - x
/progjar/progjar3 # python3 multi_thread_async.py
WARNING:root:image/jpeg
WARNING:root:writing persia.jpg dalam waktu 0:00:00.504525 2021-07-19 15:22:15.6
02502 s/d 2021-07-19 15:22:16.107032
mendownload https://s2.bukalapak.com/img/7611942961/large/IMG_20170929_WA0026_sc
aled.jpg
192.168.122.78 12000 persia.jpg
Masuk server 1
WARNING:root:image/jpeg
WARNING:root:writing anggora.jpg dalam waktu 0:00:02.387169 2021-07-19 15:22:16.
107815 s/d 2021-07-19 15:22:18.494989
mendownload https://embaran.id/wp-content/uploads/2020/12/Kucing-Persia-Abu-Abu.
jpg
Masuk server 2
192.168.122.54 12000 anggora.jpg
Waktu TOTAL yang dibutuhkan 0:00:03.298983 detik 2021-07-19 15:22:15.602497 s/d
2021-07-19 15:22:18.901480
hasil task yang dijalankan
{'persia': None, 'anggora': None}
/progjar/progjar3 #
```

17. Pada alpine-3, jalankan file `multi_thread.py` dengan command **python3 multi_process_async.py**

```
< alpine-1 x alpine-2 x alpine-3 x > - x
/progjar/progjar3 # python3 multi_process_async.py
WARNING:root:image/jpeg
WARNING:root:writing persia.jpg dalam waktu 0:00:00.416117 2021-07-19 15:22:53.6
13910 s/d 2021-07-19 15:22:54.030032
mendownload https://s2.bukalapak.com/img/7611942961/large/IMG_20170929_WA0026_sc
aled.jpg
192.168.122.78 12000 persia.jpg
Masuk server 1
WARNING:root:image/jpeg
WARNING:root:writing anggora.jpg dalam waktu 0:00:01.043600 2021-07-19 15:22:54.
051629 s/d 2021-07-19 15:22:55.095234
mendownload https://embaran.id/wp-content/uploads/2020/12/Kucing-Persia-Abu-Abu.
jpg
Masuk server 2
192.168.122.54 12000 anggora.jpg
Waktu TOTAL yang dibutuhkan 0:00:01.916747 detik 2021-07-19 15:22:53.613827 s/d
2021-07-19 15:22:55.530574
status TASK
{'persia': None, 'anggora': None}
/progjar/progjar3 #
```

D. Hasil

Setelah semua telah dijalankan dengan urut, maka alpine-1 dan alpine-2 selaku server akan menghasilkan seperti berikut

Alpine-1 (Server 1)

< role	alpine-1	×	alpine-2	×	alpine-3	>	—	×
('192.168.122.26', 53494)	20586	1	b'\x9a'					
('192.168.122.26', 53494)	20587	1	b'Y'					
('192.168.122.26', 53494)	20588	1	b'\xfa'					
('192.168.122.26', 53494)	20589	1	b'\x84'					
('192.168.122.26', 53494)	20590	1	b'\xf3'					
('192.168.122.26', 53494)	20591	1	b'=''					
('192.168.122.26', 53494)	20592	1	b'\x8b'					
('192.168.122.26', 53494)	20593	1	b'\xaf'					
('192.168.122.26', 53494)	20594	1	b'\xa8'					
('192.168.122.26', 53494)	20595	1	b'\x04'					
('192.168.122.26', 53494)	20596	1	b'j'					
('192.168.122.26', 53494)	20597	1	b'-'					
('192.168.122.26', 53494)	20598	1	b'\xa8'					
('192.168.122.26', 53494)	20599	1	b' '					
('192.168.122.26', 53494)	20600	1	b'6'					
('192.168.122.26', 53494)	20601	1	b'~'					
('192.168.122.26', 53494)	20602	1	b'\xa2'					
('192.168.122.26', 53494)	20603	1	b't'					
('192.168.122.26', 53494)	20604	1	b'\t'					
('192.168.122.26', 53494)	20605	1	b'\xb4'					
('192.168.122.26', 53494)	20606	1	b'?'					
('192.168.122.26', 53494)	20607	1	b'd'					
('192.168.122.26', 53494)	20608	1	b'\x0b'					

Alpine-2 (Server-2)

< role	alpine-1	×	alpine-2	×	alpine-3	>	—	×
('192.168.122.26', 49478)	14505	1	b'\x0b'					
('192.168.122.26', 49478)	14506	1	b'\x0f'					
('192.168.122.26', 49478)	14507	1	b'\x85'					
('192.168.122.26', 49478)	14508	1	b'4'					
('192.168.122.26', 49478)	14509	1	b'\xd6'					
('192.168.122.26', 49478)	14510	1	b'\x86'					
('192.168.122.26', 49478)	14511	1	b'\x9f'					
('192.168.122.26', 49478)	14512	1	b'\x88'					
('192.168.122.26', 49478)	14513	1	b'`'					
('192.168.122.26', 49478)	14514	1	b'4'					
('192.168.122.26', 49478)	14515	1	b'\xfb'					
('192.168.122.26', 49478)	14516	1	b'\xc2'					
('192.168.122.26', 49478)	14517	1	b'9'					
('192.168.122.26', 49478)	14518	1	b'\x8b'					
('192.168.122.26', 49478)	14519	1	b'\xdb'					
('192.168.122.26', 49478)	14520	1	b'a'					
('192.168.122.26', 49478)	14521	1	b'\xb9'					
('192.168.122.26', 49478)	14522	1	b'\xbe'					
('192.168.122.26', 49478)	14523	1	b' '					
('192.168.122.26', 49478)	14524	1	b'\x92'					
('192.168.122.26', 49478)	14525	1	b'h'					
('192.168.122.26', 49478)	14526	1	b'6'					
('192.168.122.26', 49478)	14527	1	b'p'					