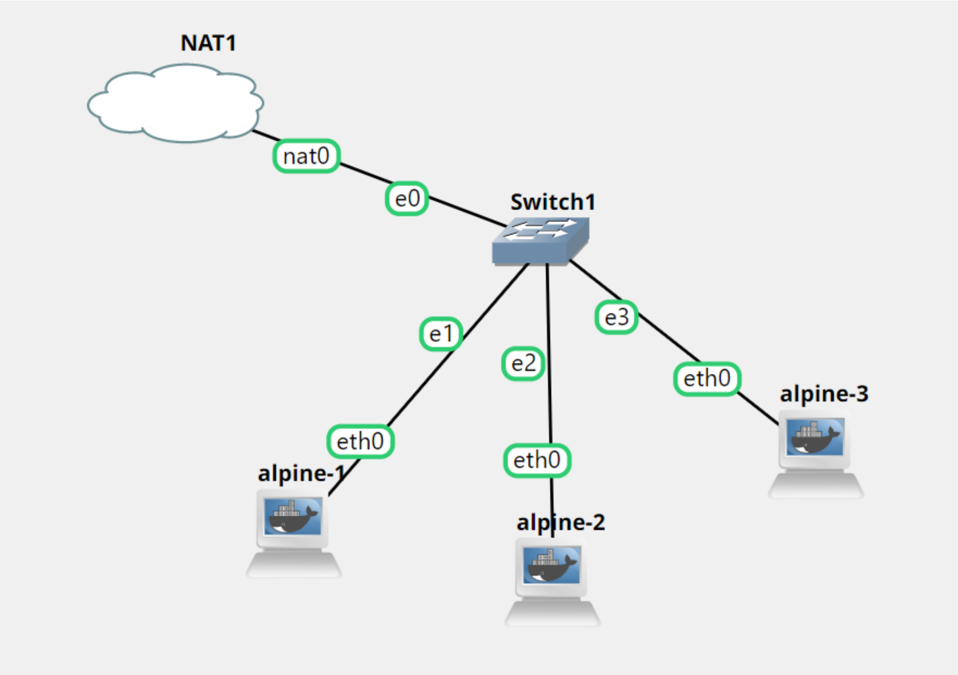
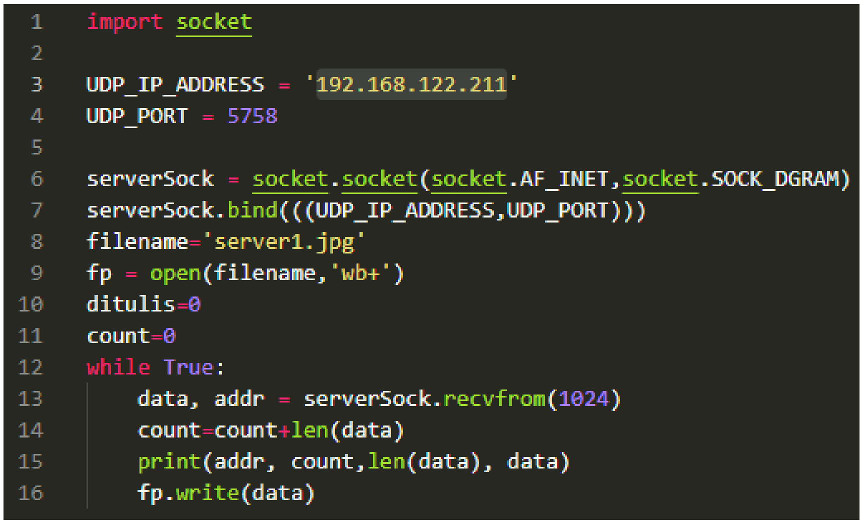
Nama : Michael Ricky
NRP : 05111840000078

Tugas Kasus Concurrency

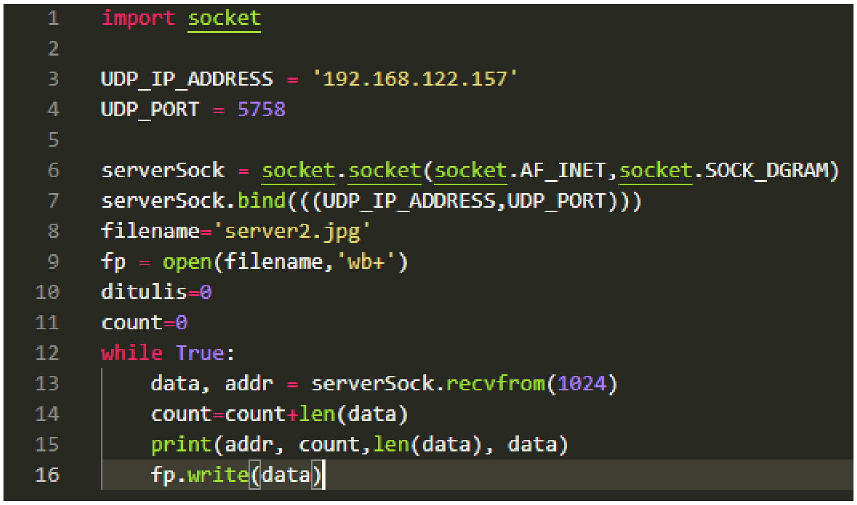
1. Kasus yang diambil adalah pengambilan gambar dari server. Dalam hal ini, alpine-1 dan alpine-2 akan bertindak sebagai server 1 dan 2. Alpine-3 akan menjadi client. Akan dilakukan 4 kali percobaan, menggunakan 4 tipe kasus, yaitu multi thread, multi thread async, multi process, dan multi process async.
2. 

3. Screenshots untuk tiap program:
    a. file_server1



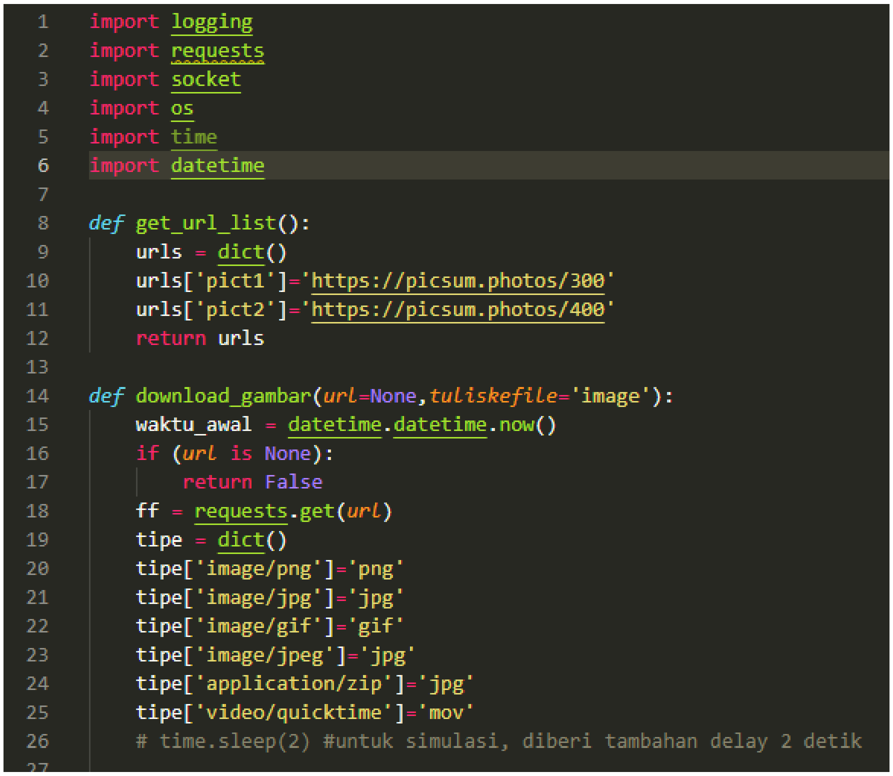
```python
import socket

UDP_IP_ADDRESS = '192.168.122.211'
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)
```

b. file_server2

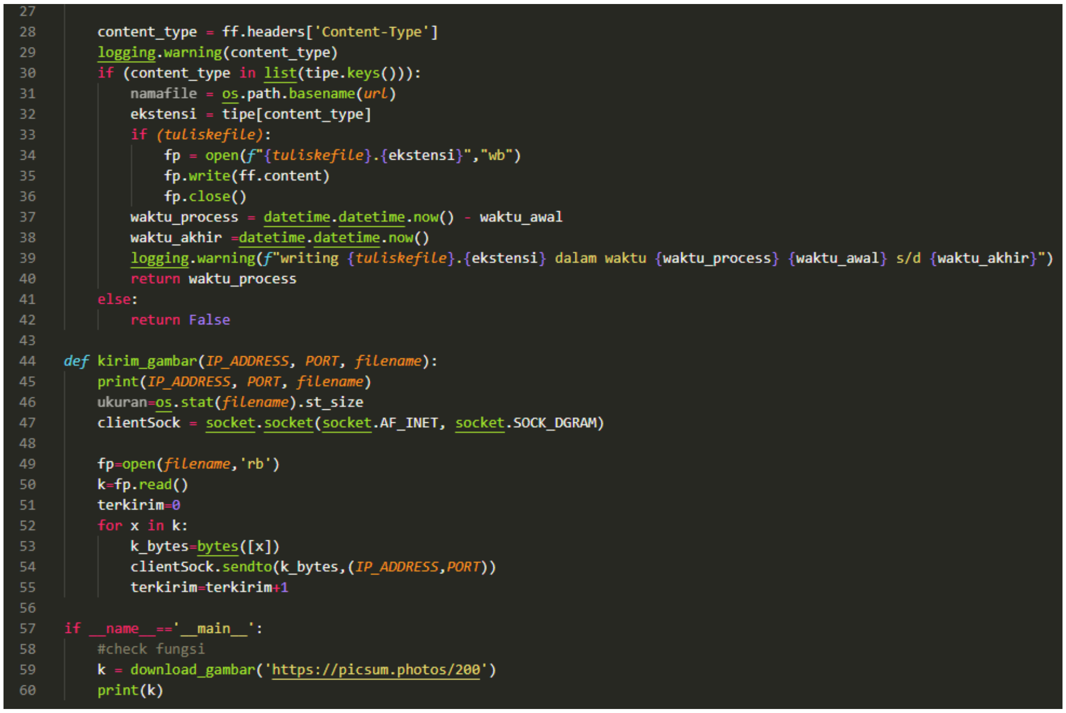


```python
import socket

UDP_IP_ADDRESS = '192.168.122.157'
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)
```

c. library

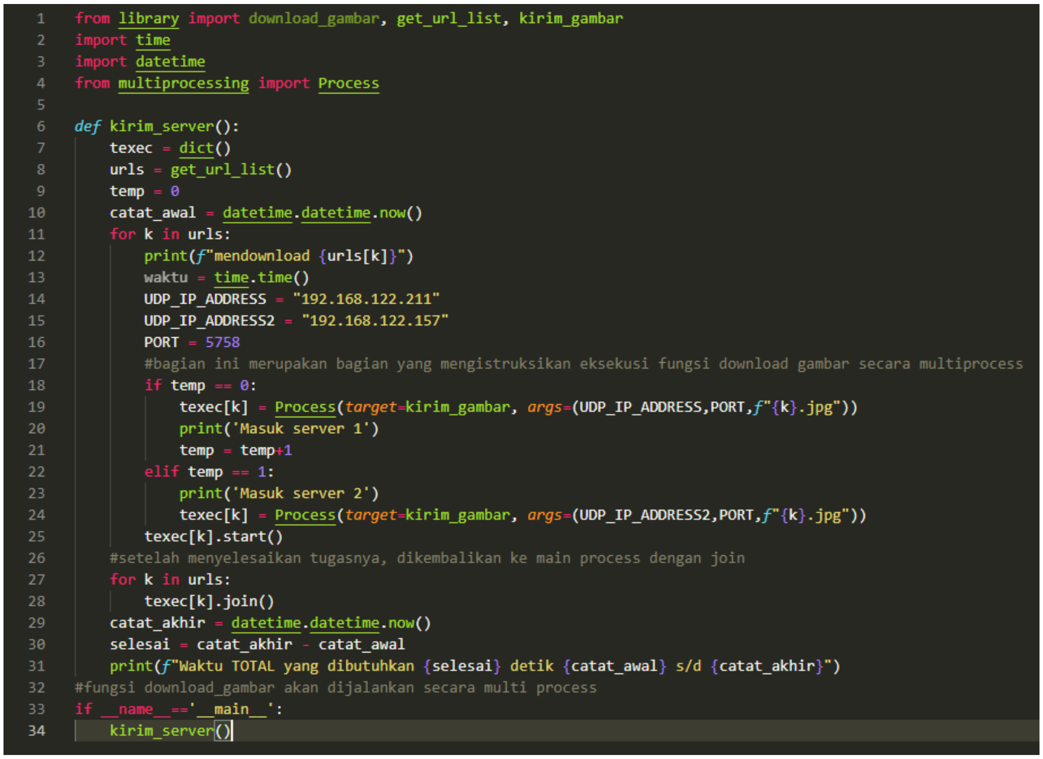


```python
import logging
import requests
import socket
import os
import time
import datetime

def get_url_list():
    urls = dict()
    urls['pict1']='https://picsum.photos/300'
    urls['pict2']='https://picsum.photos/400'
    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/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
```



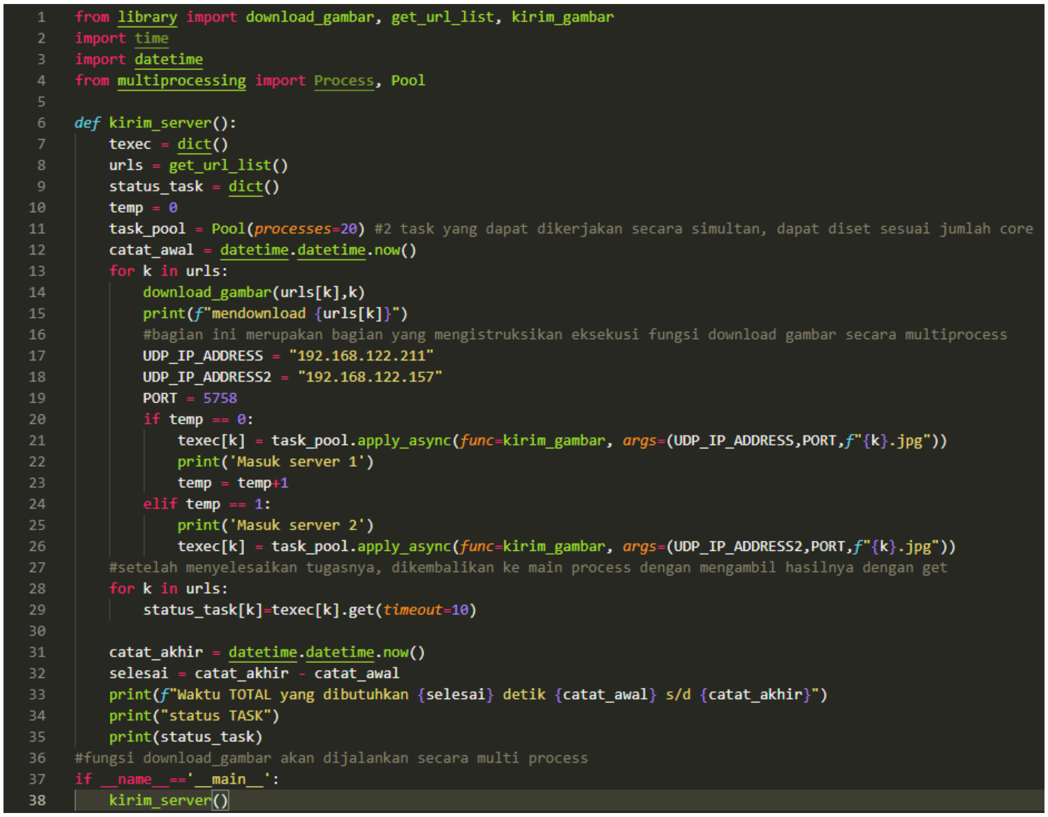
```python

    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://picsum.photos/200')
    print(k)
```

d. multi_process



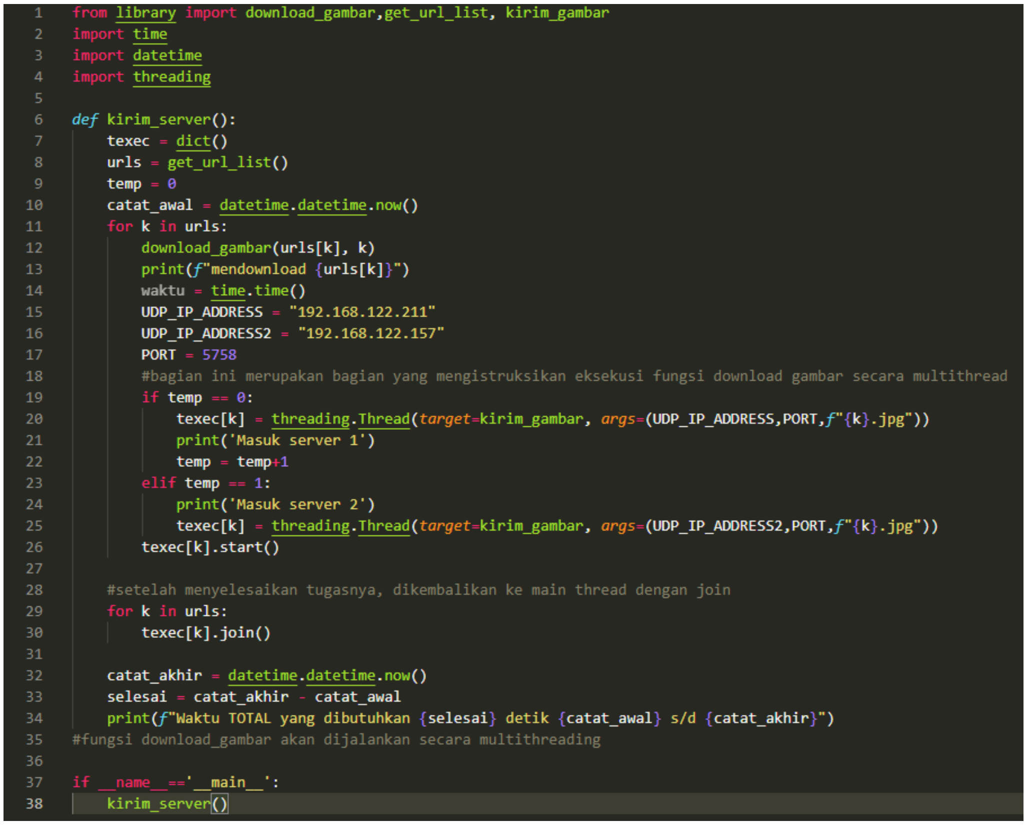
```python
from library import download_gambar, get_url_list, kirim_gambar
import time
import datetime
from multiprocessing import Process

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 = "192.168.122.211"
        UDP_IP_ADDRESS2 = "192.168.122.157"
        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"))
            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()
```

e. multi_process_async



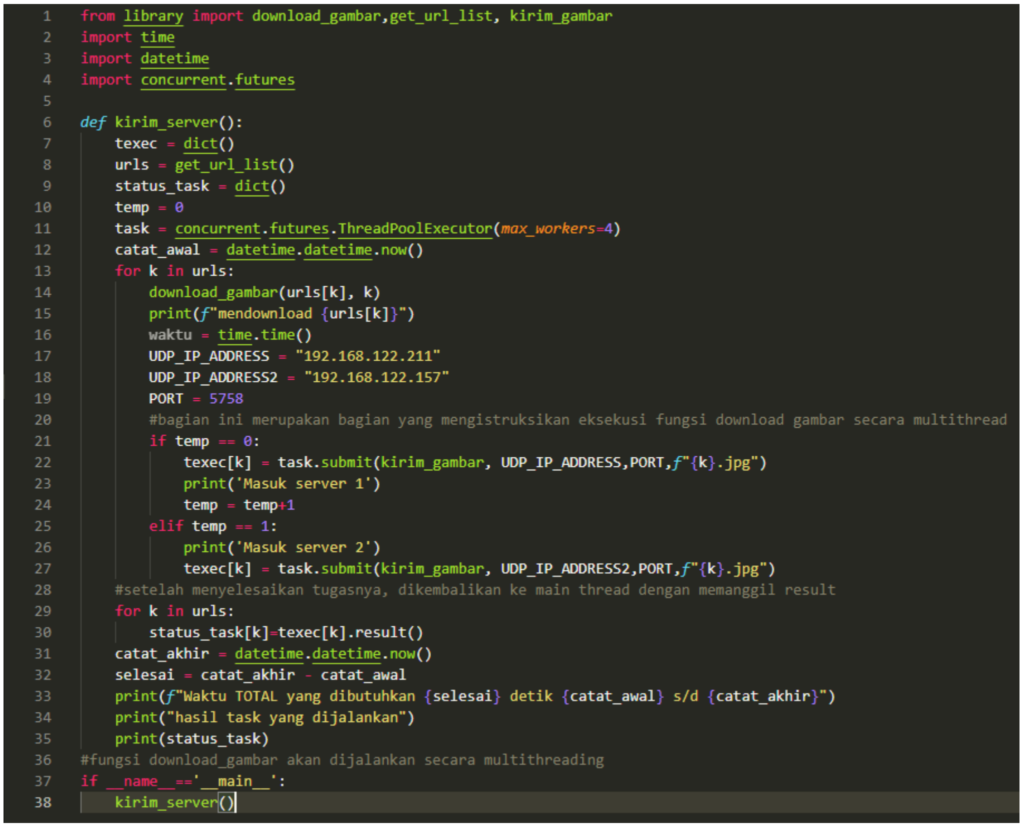
```python
from library import download_gambar, get_url_list, kirim_gambar
import time
import datetime
from multiprocessing import Process, Pool

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 mengistruksikan eksekusi fungsi download gambar secara multiprocess
        UDP_IP_ADDRESS = "192.168.122.211"
        UDP_IP_ADDRESS2 = "192.168.122.157"
        PORT = 5758
        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()
```

f. multi_thread



```
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 = "192.168.122.211"
        UDP_IP_ADDRESS2 = "192.168.122.157"
        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()
```

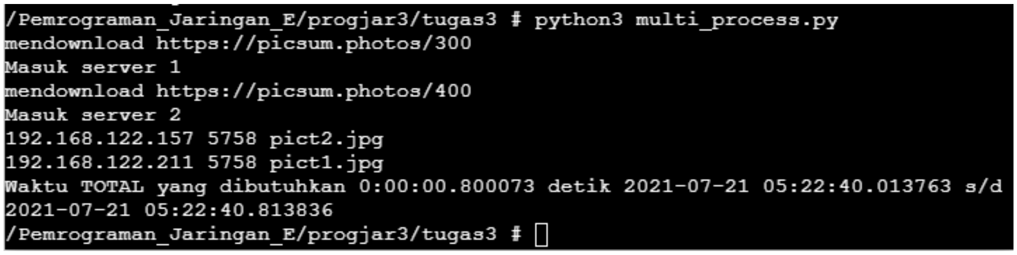
g. multi_thread_async



```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 = "192.168.122.211"
        UDP_IP_ADDRESS2 = "192.168.122.157"
        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")
    #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()
```

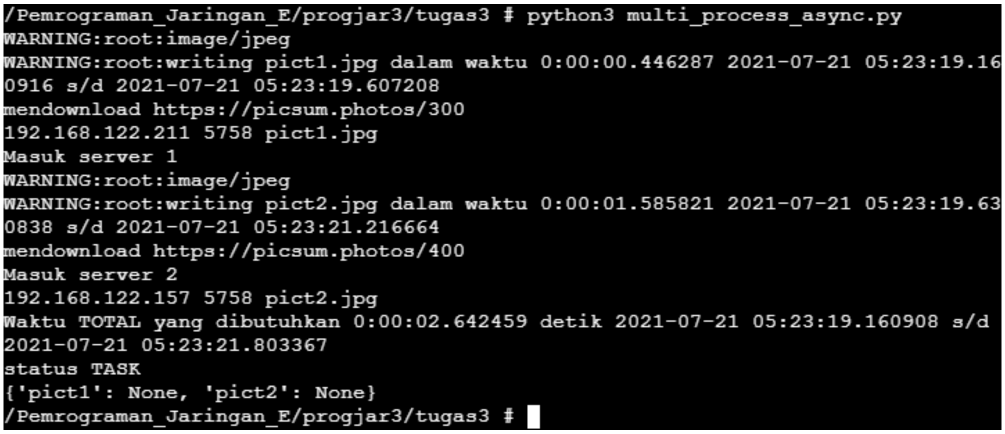
4. Hasil penjalanan tiap program

a. Multi_process



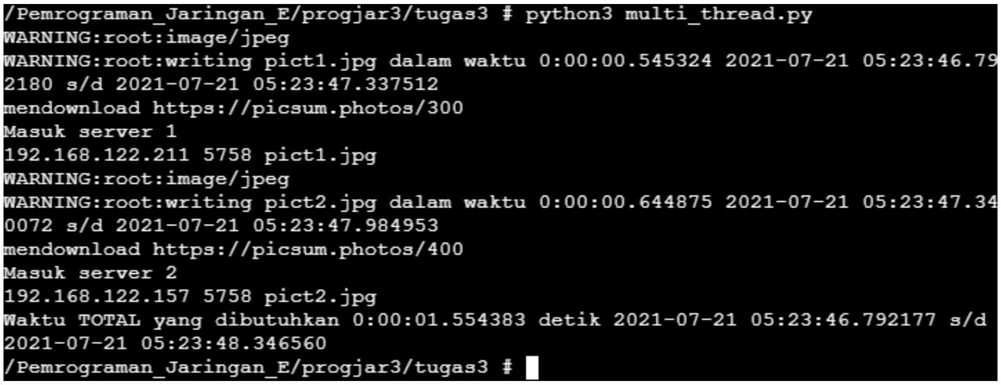
```
/Pemrograman_Jaringan_E/progjar3/tugas3 # python3 multi_process.py
mendownload https://picsum.photos/300
Masuk server 1
mendownload https://picsum.photos/400
Masuk server 2
192.168.122.157 5758 pict2.jpg
192.168.122.211 5758 pict1.jpg
Waktu TOTAL yang dibutuhkan 0:00:00.800073 detik 2021-07-21 05:22:40.013763 s/d
2021-07-21 05:22:40.813836
/Pemrograman_Jaringan_E/progjar3/tugas3 # 
```

b. Multi_process_async



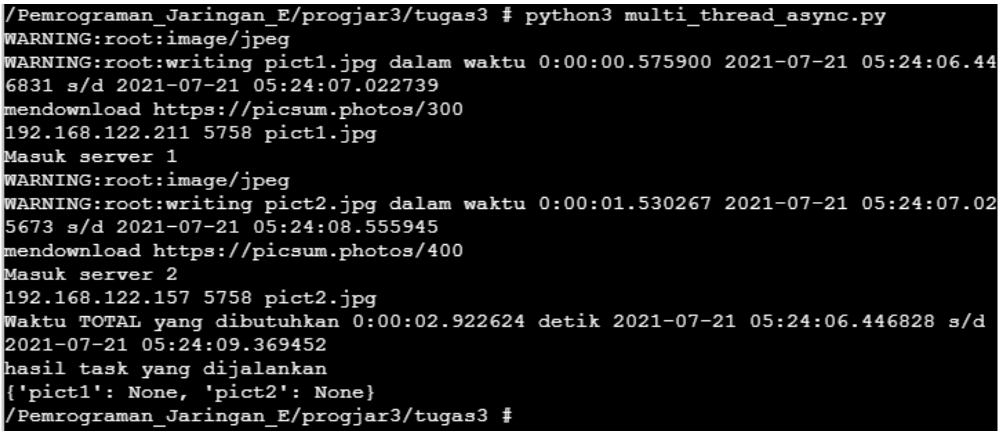
```
/Pemrograman_Jaringan_E/progjar3/tugas3 # python3 multi_process_async.py
WARNING:root:image/jpeg
WARNING:root:writing pict1.jpg dalam waktu 0:00:00.446287 2021-07-21 05:23:19.16
0916 s/d 2021-07-21 05:23:19.607208
mendownload https://picsum.photos/300
192.168.122.211 5758 pict1.jpg
Masuk server 1
WARNING:root:image/jpeg
WARNING:root:writing pict2.jpg dalam waktu 0:00:01.585821 2021-07-21 05:23:19.63
0838 s/d 2021-07-21 05:23:21.216664
mendownload https://picsum.photos/400
Masuk server 2
192.168.122.157 5758 pict2.jpg
Waktu TOTAL yang dibutuhkan 0:00:02.642459 detik 2021-07-21 05:23:19.160908 s/d
2021-07-21 05:23:21.803367
status TASK
{'pict1': None, 'pict2': None}
/Pemrograman_Jaringan_E/progjar3/tugas3 #
```

c. Multi_thread



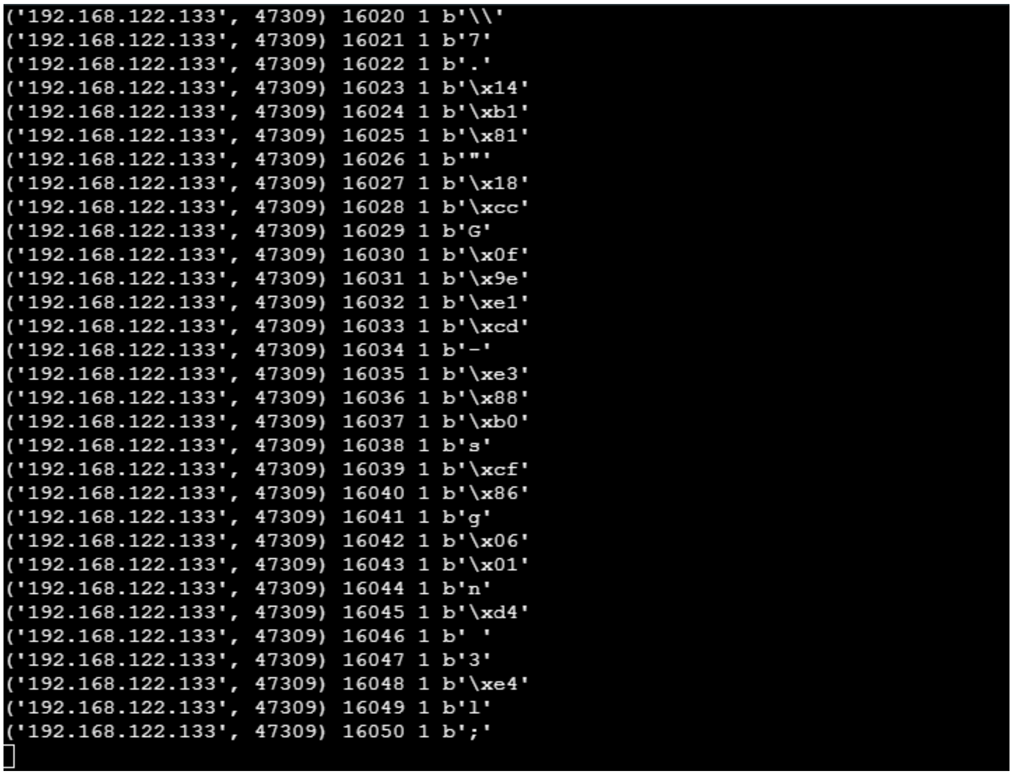
```
/Pemrograman_Jaringan_E/progjar3/tugas3 # python3 multi_thread.py
WARNING:root:image/jpeg
WARNING:root:writing pict1.jpg dalam waktu 0:00:00.545324 2021-07-21 05:23:46.79
2180 s/d 2021-07-21 05:23:47.337512
mendownload https://picsum.photos/300
Masuk server 1
192.168.122.211 5758 pict1.jpg
WARNING:root:image/jpeg
WARNING:root:writing pict2.jpg dalam waktu 0:00:00.644875 2021-07-21 05:23:47.34
0072 s/d 2021-07-21 05:23:47.984953
mendownload https://picsum.photos/400
Masuk server 2
192.168.122.157 5758 pict2.jpg
Waktu TOTAL yang dibutuhkan 0:00:01.554383 detik 2021-07-21 05:23:46.792177 s/d
2021-07-21 05:23:48.346560
/Pemrograman_Jaringan_E/progjar3/tugas3 #
```

d. Multi_thread_async



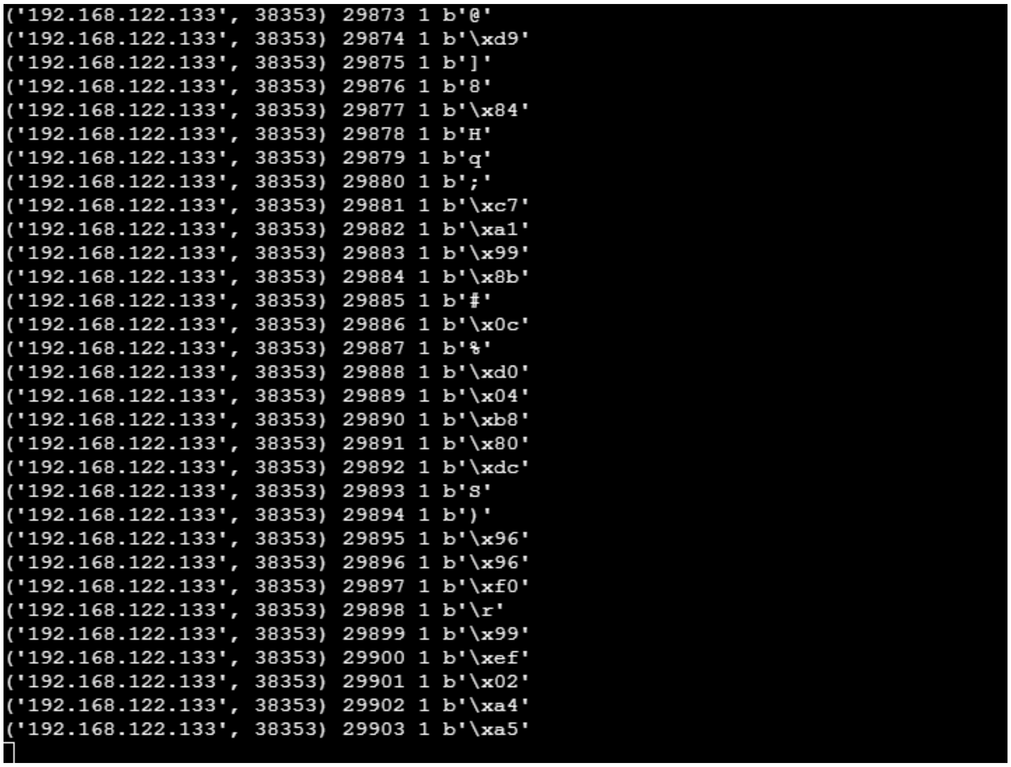
```
/Pemrograman_Jaringan_E/progjar3/tugas3 # python3 multi_thread_async.py
WARNING:root:image/jpeg
WARNING:root:writing pict1.jpg dalam waktu 0:00:00.575900 2021-07-21 05:24:06.44
6831 s/d 2021-07-21 05:24:07.022739
mendownload https://picsum.photos/300
192.168.122.211 5758 pict1.jpg
Masuk server 1
WARNING:root:image/jpeg
WARNING:root:writing pict2.jpg dalam waktu 0:00:01.530267 2021-07-21 05:24:07.02
5673 s/d 2021-07-21 05:24:08.555945
mendownload https://picsum.photos/400
Masuk server 2
192.168.122.157 5758 pict2.jpg
Waktu TOTAL yang dibutuhkan 0:00:02.922624 detik 2021-07-21 05:24:06.446828 s/d
2021-07-21 05:24:09.369452
hasil task yang dijalankan
{'pict1': None, 'pict2': None}
/Pemrograman_Jaringan_E/progjar3/tugas3 #
```

5. Hasil di server
    a. Server 1



```
('192.168.122.133', 47309) 16020 1 b'\\'
('192.168.122.133', 47309) 16021 1 b'7'
('192.168.122.133', 47309) 16022 1 b'.'
('192.168.122.133', 47309) 16023 1 b'\x14'
('192.168.122.133', 47309) 16024 1 b'\xb1'
('192.168.122.133', 47309) 16025 1 b'\x81'
('192.168.122.133', 47309) 16026 1 b'"'
('192.168.122.133', 47309) 16027 1 b'\x18'
('192.168.122.133', 47309) 16028 1 b'\xcc'
('192.168.122.133', 47309) 16029 1 b'G'
('192.168.122.133', 47309) 16030 1 b'\x0f'
('192.168.122.133', 47309) 16031 1 b'\x9e'
('192.168.122.133', 47309) 16032 1 b'\xe1'
('192.168.122.133', 47309) 16033 1 b'\xcd'
('192.168.122.133', 47309) 16034 1 b'-'
('192.168.122.133', 47309) 16035 1 b'\xe3'
('192.168.122.133', 47309) 16036 1 b'\x88'
('192.168.122.133', 47309) 16037 1 b'\xb0'
('192.168.122.133', 47309) 16038 1 b's'
('192.168.122.133', 47309) 16039 1 b'\xcf'
('192.168.122.133', 47309) 16040 1 b'\x86'
('192.168.122.133', 47309) 16041 1 b'g'
('192.168.122.133', 47309) 16042 1 b'\x06'
('192.168.122.133', 47309) 16043 1 b'\x01'
('192.168.122.133', 47309) 16044 1 b'n'
('192.168.122.133', 47309) 16045 1 b'\xd4'
('192.168.122.133', 47309) 16046 1 b' '
('192.168.122.133', 47309) 16047 1 b'3'
('192.168.122.133', 47309) 16048 1 b'\xe4'
('192.168.122.133', 47309) 16049 1 b'l'
('192.168.122.133', 47309) 16050 1 b';'
```

b. Server 2



```
('192.168.122.133', 38353) 29873 1 b'@'
('192.168.122.133', 38353) 29874 1 b'\xd9'
('192.168.122.133', 38353) 29875 1 b']'
('192.168.122.133', 38353) 29876 1 b'8'
('192.168.122.133', 38353) 29877 1 b'\x84'
('192.168.122.133', 38353) 29878 1 b'H'
('192.168.122.133', 38353) 29879 1 b'q'
('192.168.122.133', 38353) 29880 1 b';'
('192.168.122.133', 38353) 29881 1 b'\xc7'
('192.168.122.133', 38353) 29882 1 b'\xa1'
('192.168.122.133', 38353) 29883 1 b'\x99'
('192.168.122.133', 38353) 29884 1 b'\x8b'
('192.168.122.133', 38353) 29885 1 b'#'
('192.168.122.133', 38353) 29886 1 b'\x0c'
('192.168.122.133', 38353) 29887 1 b'%'
('192.168.122.133', 38353) 29888 1 b'\xd0'
('192.168.122.133', 38353) 29889 1 b'\x04'
('192.168.122.133', 38353) 29890 1 b'\xb8'
('192.168.122.133', 38353) 29891 1 b'\x80'
('192.168.122.133', 38353) 29892 1 b'\xdc'
('192.168.122.133', 38353) 29893 1 b'S'
('192.168.122.133', 38353) 29894 1 b')'
('192.168.122.133', 38353) 29895 1 b'\x96'
('192.168.122.133', 38353) 29896 1 b'\x96'
('192.168.122.133', 38353) 29897 1 b'\xf0'
('192.168.122.133', 38353) 29898 1 b'\r'
('192.168.122.133', 38353) 29899 1 b'\x99'
('192.168.122.133', 38353) 29900 1 b'\xef'
('192.168.122.133', 38353) 29901 1 b'\x02'
('192.168.122.133', 38353) 29902 1 b'\xa4'
('192.168.122.133', 38353) 29903 1 b'\xa5'
```