

Dokumentasi Praktikum Implementasi Kasus menggunakan  
Concurrency



Pemrograman Jaringan  
Kelas E

Muhammad Afif Fadhlurrahman  
0511184000093

Institut Teknologi Sepuluh Nopember  
Surabaya  
2021

## Praktikum :

Buatlah program yang mengimplementasikan

1. multi process
2. multi thread
3. multi process asynchronous
4. multi thread asynchronous

Dengan menggunakan protokol transport UDP. kasus dapat didefinisikan sendiri. dan Buatlah arsitektur jaringan anda sendiri di simulator GNS3

Buatlah laporan dalam bentuk PDF yang berisikan screenshot dari

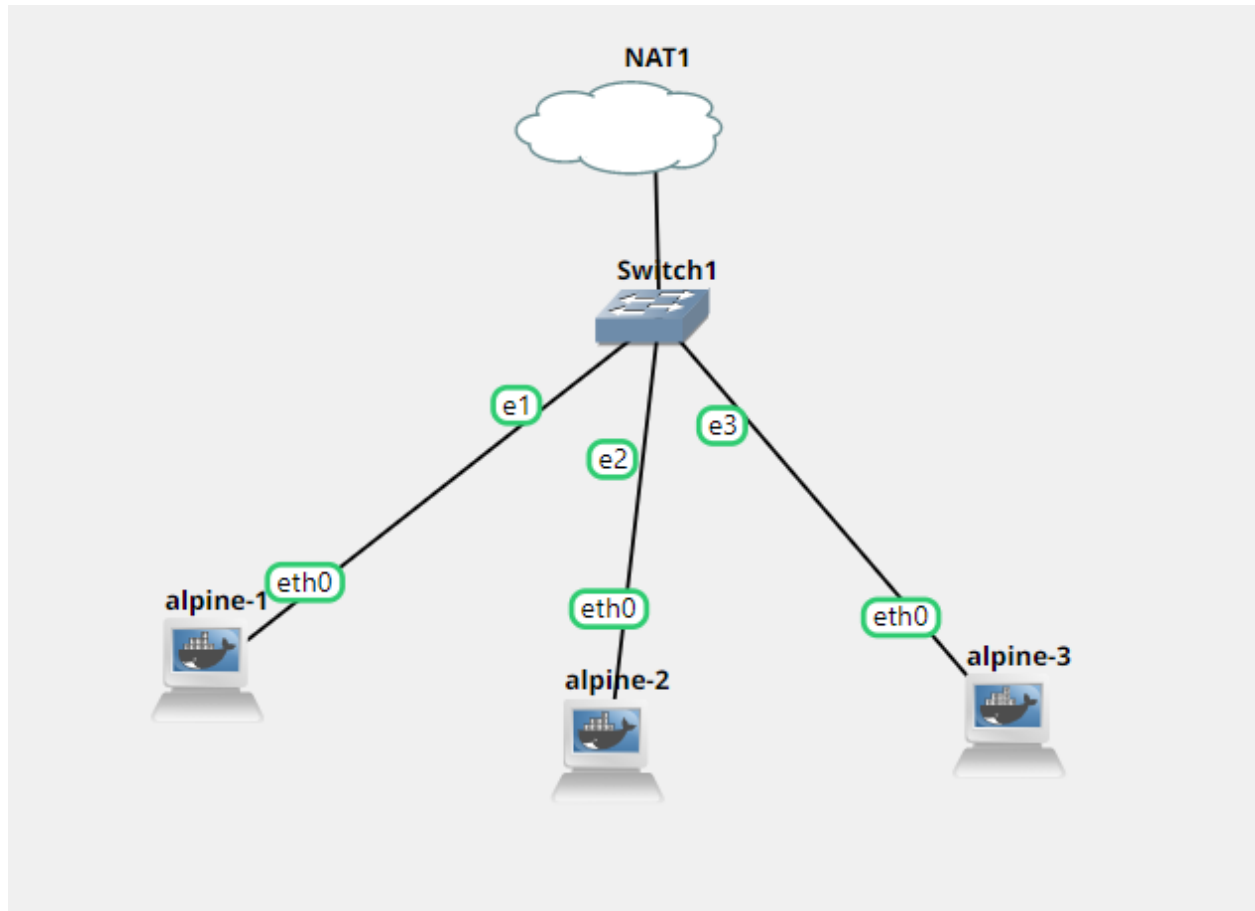
1. deskripsi kasus yang dibuat
2. gambar arsitektur jaringan (dalam simulator GNS3)
3. program yang dibuat (1-4)
4. hasil outputnya

## Deskripsi Kasus

Pada project GNS3 akan terdapat 3 alpine, alpine 1 dan alpine 2 sebagai server serta alpine 3 akan menjadi client. Cara kerjanya alpine 3 mendownload 2 file terlebih dahulu, kemudian mengirimkan file tersebut ke alpine 1 dan alpine 2 sebagai server menggunakan protokol UDP. Kemudian dari 2 file tersebut akan dikirimkan ke alpine 1 (server) dan alpine 2 (server) masing-masing 1 file. Pengiriman ke 2 server dilakukan dengan 4 program yaitu :

1. multi process
2. multi thread
3. multi process asynchronous
4. multi thread asynchronous

## Arsitektur Jaringan



### Langkah-langkah :

1. Check IP address pada alpine 1

```
alpine-1  alpine-2  alpine-3
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr DA:A7:E5:37:A9:EF
          inet addr:192.168.122.129  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:62 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11630 (11.3 KiB)  TX bytes:3642 (3.5 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)

/ #
```

2. Cek IP address pada alpine 2

```
alpine-1  alpine-2  alpine-3
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 66:7D:90:DF:8C:C7
          inet addr:192.168.122.77  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:147 errors:0 dropped:4 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16378 (15.9 KiB)  TX bytes:3642 (3.5 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)

/ #
```

3. Sebelum memulai, siapkan 7 file code python, yang terdiri dari :

- a. server1.py

```
import socket

SERVER_IP = '192.168.122.129'
SERVER_PORT = 5005
```

```

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((SERVER_IP, SERVER_PORT))
filename = 'server1.jpg'
fp = open(filename, 'wb+')
count = 0

while True:
    print("SERVER 1")
    data, addr = sock.recvfrom(1024)
    count += len(data)
    # buffer size 1024
    print("dikirim oleh [" + addr + "] count: ",
          count, " panjang: ", len(data), data)
    fp.write(data)
{"mode": "full", "isActive": false}

```

b. server2.py

```

import socket

SERVER_IP = '192.168.122.77'
SERVER_PORT = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((SERVER_IP, SERVER_PORT))
filename = 'server1.jpg'
fp = open(filename, 'wb+')
count = 0

while True:
    print("SERVER 2")
    data, addr = sock.recvfrom(1024)
    count += len(data)
    # buffer size 1024
    print("dikirim oleh [" + addr + "] count: ",
          count, " panjang: ", len(data), data)
    fp.write(data)
{"mode": "full", "isActive": false}

```

c. udp\_multi\_process.py

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

def download_semua():
    texec = dict()
    urls = get_url_list()
    catat_awal = datetime.datetime.now()
    flag = 0
    for k in urls:
        download_gambar(urls[k], k)
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        IP_SERVER1 = "192.168.122.129"
        IP_SERVER2 = "192.168.122.77"
        if flag == 0:
            # bagian ini merupakan bagian yang menginstruksikan
            # eksekusi fungsi download gambar secara multiprocess
            print("SERVER 1")
            texec[k] =
Process(target=send_image,args=(IP_SERVER1, 5005, f"{k}.jpg"))
            flag += 1
        elif flag == 1:
            print("SERVER 2")
            texec[k] =
Process(target=send_image,args=(IP_SERVER2, 5005, 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__':
    download_semua()
{"mode":"full","isActive":false}
```

d. udp\_multi\_process\_async.py

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

def download_semua():
    IP_SERVER1 = "192.168.122.129"
    IP_SERVER2 = "192.168.122.77"
    texec = dict()
    urls = get_url_list()
    status_task = dict()
    flag = 0
    # 2 task yang dapat dikerjakan secara simultan, dapat
    # diset sesuai jumlah core
    task_pool = Pool(processes=20)
    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 multiprocess
        if flag == 0:
            print("SERVER 1")
            texec[k] = task_pool.apply_async(
                func=send_image, args=(IP_SERVER1, 5005,
f"{k}.jpg"))
            flag += 1
        elif flag == 1:
            print("SERVER 2")
            texec[k] = task_pool.apply_async(
                func=send_image, args=(IP_SERVER2, 5005,
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__':
    download_semua()
{"mode": "full", "isActive": false}

```

e. udp\_multi\_thread.py

```

from library import download_gambar, get_url_list, send_image
import time
import datetime
import threading

def download_semua():
    IP_SERVER1 = "192.168.122.129"
    IP_SERVER2 = "192.168.122.77"
    texec = dict()
    urls = get_url_list()
    flag = 0
    catat_awal = datetime.datetime.now()
    for k in urls:
        download_gambar(urls[k], k)
        print(f"mendownload {urls[k]}")
        waktu = time.time()

```



```

        if flag == 0:
            print("SERVER 1")
            # bagian ini merupakan bagian yang menginstruksikan
            # eksekusi fungsi download gambar secara multithread
            texec[k] = threading.Thread(
                target=send_image, args=(IP_SERVER1, 5005,
f"{k}.jpg"))
            flag += 1
        elif flag == 1:
            print("SERVER 2")
            texec[k] = threading.Thread(
                target=send_image, args=(IP_SERVER2, 5005,
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__':
    download_semua()
{"mode": "full", "isActive": false}

```

f. udp\_multi\_thread\_async.py

```

from library import download_gambar, get_url_list, send_image
import time
import datetime
import concurrent.futures

```

```

def download_semua():
    IP_SERVER1 = "192.168.122.129"
    IP_SERVER2 = "192.168.122.77"
    texec = dict()
    urls = get_url_list()
    status_task = dict()
    task =
concurrent.futures.ThreadPoolExecutor(max_workers=4)
    catat_awal = datetime.datetime.now()
    flag = 0
    for k in urls:
        download_gambar(urls[k], k)
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        if flag == 0:
            print("SERVER 1")
            # bagian ini merupakan bagian yang menginstruksikan
eksekusi fungsi download gambar secara multithread
            texec[k] = task.submit(send_image, IP_SERVER1,
5005, f"{k}.jpg")
            flag += 1
        elif flag == 1:
            print("SERVER 2")
            texec[k] = task.submit(send_image, IP_SERVER2,
5005, 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__':
    download_semua()
{"mode": "full", "isActive": false}
```

g. library.py

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

def get_url_list():
    urls = dict()
    urls['kompas'] =
'https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0
x0:998x665/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg'
    urls['detik'] =
'https://awscdn.detik.net.id/detik2/images/logo.jpg'
    # urls['its'] =
'https://www.its.ac.id/wp-content/uploads/2017/09/Gambar2111-1
-1024x683.jpg'
    # urls['file1'] =
'https://file-examples-com.github.io/uploads/2018/04/file_exam
ple_MOV_480_700kB.mov'
    #
urls['file2']='https://file-examples-com.github.io/uploads/201
8/04/file_example_MOV_1280_1_4MB.mov'
    # urls['file3'] =
'https://file-examples-com.github.io/uploads/2017/02/zip_2MB.z
ip'
    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

```

```

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 send_image(IP_ADDRESS, PORT, filename):
    print(IP_ADDRESS, PORT, filename)
    ukuran = os.stat(filename).st_size
    sockclient = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
    sent = 0
    fp = open(filename, 'rb')

```

```

        file = fp.read()
        for i in file:
            file_bytes = bytes([i])
            sockclient.sendto(file_bytes, (IP_ADDRESS, PORT))
            sent += 1

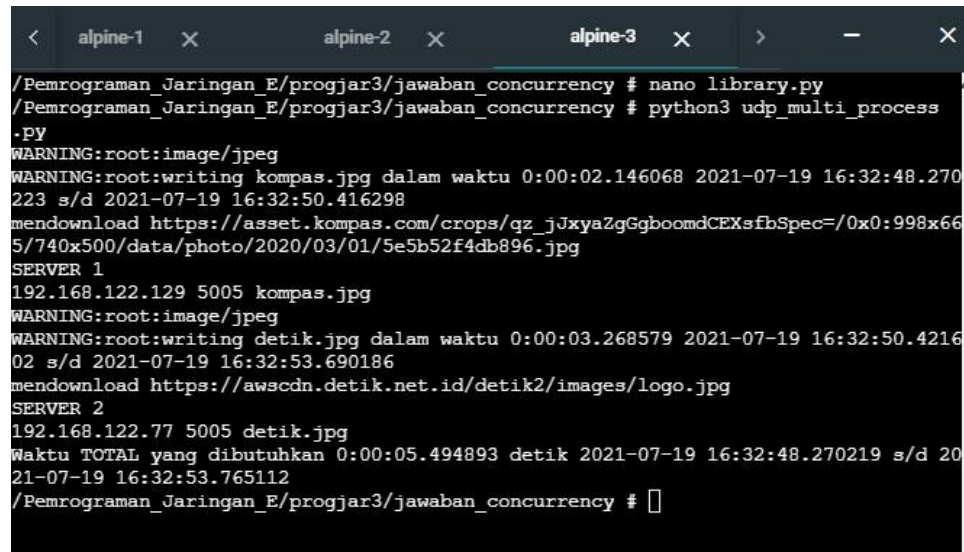
if __name__ == '__main__':
    # check fungsi
    k = download_gambar(

'https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0
x0:998x665/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg')
    print(k)
{"mode": "full", "isActive": false}

```

#### 4. Multi-process : Jalankan code `udp_multi_process.py` pada alpine 3

- a. Hasil output di alpine 3



```

< alpine-1 x alpine-2 x alpine-3 x > - x
/Pemrograman_Jaringan_E/progjar3/jawaban_concurrency # nano library.py
/Pemrograman_Jaringan_E/progjar3/jawaban_concurrency # python3 udp_multi_process
.PY
WARNING:root:image/jpeg
WARNING:root:writing kompas.jpg dalam waktu 0:00:02.146068 2021-07-19 16:32:48.270
223 s/d 2021-07-19 16:32:50.416298
mendownload https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0x0:998x66
5/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg
SERVER 1
192.168.122.129 5005 kompas.jpg
WARNING:root:image/jpeg
WARNING:root:writing detik.jpg dalam waktu 0:00:03.268579 2021-07-19 16:32:50.4216
02 s/d 2021-07-19 16:32:53.690186
mendownload https://awscdn.detik.net.id/detik2/images/logo.jpg
SERVER 2
192.168.122.77 5005 detik.jpg
Waktu TOTAL yang dibutuhkan 0:00:05.494893 detik 2021-07-19 16:32:48.270219 s/d 20
21-07-19 16:32:53.765112
/Pemrograman_Jaringan_E/progjar3/jawaban_concurrency # 

```

b. Hasil output di alpine 2 (server)

< ole	alpine-1	×	alpine-2	×	alpin	>	-	×
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	603	panjang:	1 b'\x8a'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	604	panjang:	1 b'\x95'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	605	panjang:	1 b'\xef'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	606	panjang:	1 b'\xc0'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	607	panjang:	1 b'\xdb'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	608	panjang:	1 b'8'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	609	panjang:	1 b'D'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	610	panjang:	1 b'\xc6'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	611	panjang:	1 b'\x1d'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	612	panjang:	1 b'\x16'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	613	panjang:	1 b'\xa8'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	614	panjang:	1 b'g'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	615	panjang:	1 b'\x12'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	616	panjang:	1 b'A'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	617	panjang:	1 b'*'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	618	panjang:	1 b'a'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	619	panjang:	1 b'''		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	620	panjang:	1 b'B'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	621	panjang:	1 b'<'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	622	panjang:	1 b'a'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	623	panjang:	1 b'\xd6'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	624	panjang:	1 b'\xb3'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	625	panjang:	1 b'\xd2'		
dikirim oleh	[ ('192.168.122.179', 35278) ]		count:	626	panjang:	1 b'\x0e'		

c. Hasil output di alpine 1 (server)

< ole	alpine-1	×	alpine-2	×	alpin	>	-	×
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5381	panjang:	1 b'\x97'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5382	panjang:	1 b'\xe8'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5383	panjang:	1 b'('		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5384	panjang:	1 b'\x87'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5385	panjang:	1 b'\x99'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5386	panjang:	1 b'P'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5387	panjang:	1 b'\x0b'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5388	panjang:	1 b'\x04'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5389	panjang:	1 b'3'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5390	panjang:	1 b'<'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5391	panjang:	1 b'\xb0'		
SERVER 1								
dikirim oleh	[ ('192.168.122.179', 59235) ]		count:	5392	panjang:	1 b','		
SERVER 1								

5. **Multi-thread** : Jalankan code **udp\_multi\_thread.py** pada alpine 3

a. Hasil output di alpine 3

```
< alpine-1 x alpine-2 x alpine-3 x > - x
/Pemrograman_Jaringan_E/progjar3/jawaban_concurrency # python3 udp_multi_thread.
py
WARNING:root:image/jpeg
WARNING:root:writing kompas.jpg dalam waktu 0:00:02.126320 2021-07-19 16:41:34.865
469 s/d 2021-07-19 16:41:36.991794
mendownload https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0x0:998x66
5/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg
SERVER 1
192.168.122.129 5005 kompas.jpg
WARNING:root:image/jpeg
WARNING:root:writing detik.jpg dalam waktu 0:00:03.923315 2021-07-19 16:41:36.9925
19 s/d 2021-07-19 16:41:40.915840
mendownload https://awscdn.detik.net.id/detik2/images/logo.jpg
SERVER 2
192.168.122.77 5005 detik.jpg
Waktu TOTAL yang dibutuhkan 0:00:06.122012 detik 2021-07-19 16:41:34.865465 s/d 20
21-07-19 16:41:40.987477
/Pemrograman_Jaringan_E/progjar3/jawaban_concurrency #
```

b. Hasil output di alpine 2 (server)

```
< alpine-1 x alpine-2 x alpine-3 x > - x
dikirim oleh [ ('192.168.122.179', 41516) ] count: 1999 panjang: 1 b'\r'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2000 panjang: 1 b'\x00'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2001 panjang: 1 b'\x01'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2002 panjang: 1 b'\x90'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2003 panjang: 1 b'\x03'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2004 panjang: 1 b' '
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2005 panjang: 1 b'\x17'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2006 panjang: 1 b'P'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2007 panjang: 1 b'\xa3'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2008 panjang: 1 b'?'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2009 panjang: 1 b'\x10'
SERVER 2
dikirim oleh [ ('192.168.122.179', 41516) ] count: 2010 panjang: 1 b'\xbf'
SERVER 2
```



- c. Hasil output di alpine 1 (server)

```
< ole alpine-1 X alpine-2 X alpin > - X
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10860 panjang: 1 b'g'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10861 panjang: 1 b'\xa4'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10862 panjang: 1 b'\xa5'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10863 panjang: 1 b'\xa5'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10864 panjang: 1 b'\xff'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10865 panjang: 1 b'\x00'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10866 panjang: 1 b'\xf2'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10867 panjang: 1 b'\xe5'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10868 panjang: 1 b'\xc6'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10869 panjang: 1 b's'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10870 panjang: 1 b'\xc1'
SERVER 1
dikirim oleh [ ('192.168.122.179', 43426) ] count: 10871 panjang: 1 b'\x04'
SERVER 1
```

6. Multi-process-async : Jalankan code `udp_multi_process_async.py` pada alpine 3

- a. Hasil output di alpine 3

```
< alpine-1 X alpine-2 X alpine-3 X > - X
mendownload https://awscdn.detik.net.id/detik2/images/logo.jpg
SERVER 2
192.168.122.77 5005 detik.jpg
Waktu TOTAL yang dibutuhkan 0:00:05.494893 detik 2021-07-19 16:32:48.270219 s/d 20
21-07-19 16:32:53.765112
/Pemrograman_Jaringan_E/progjar3/jawaban_concurrency # python3 udp_multi_process
_async.py
WARNING:root:image/jpeg
WARNING:root:writing kompas.jpg dalam waktu 0:00:02.158755 2021-07-19 16:35:58.717
290 s/d 2021-07-19 16:36:00.876049
mendownload https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0x0:998x66
5/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg
SERVER 1
192.168.122.129 5005 kompas.jpg
WARNING:root:image/jpeg
WARNING:root:writing detik.jpg dalam waktu 0:00:03.352325 2021-07-19 16:36:00.8768
43 s/d 2021-07-19 16:36:04.229173
mendownload https://awscdn.detik.net.id/detik2/images/logo.jpg
SERVER 2
192.168.122.77 5005 detik.jpg
Waktu TOTAL yang dibutuhkan 0:00:05.590307 detik 2021-07-19 16:35:58.717283 s/d 20
21-07-19 16:36:04.307590
status TASK
{'kompas': None, 'detik': None}
```



b. Hasil output di alpine 2 (server)

	<	alpine-1	×	alpine-2	×	alpine-3	×	>	-	×
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	936	panjang:	1	b'\xe0'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	937	panjang:	1	b'\xdd'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	938	panjang:	1	b'\x90'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	939	panjang:	1	b'\xe9'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	940	panjang:	1	b'''	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	941	panjang:	1	b'\xc7'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	942	panjang:	1	b'\xfa'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	943	panjang:	1	b'\xcd'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	944	panjang:	1	b'\xb4'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	945	panjang:	1	b'\x8d'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	946	panjang:	1	b'\xf6'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	947	panjang:	1	b'5'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	948	panjang:	1	b'\xdd'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	949	panjang:	1	b'\xea'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	950	panjang:	1	b'\xfb'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	951	panjang:	1	b'Y'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	952	panjang:	1	b'\xde'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	953	panjang:	1	b'\xa9'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	954	panjang:	1	b'\x1c'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	955	panjang:	1	b'j'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	956	panjang:	1	b'\xb1'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	957	panjang:	1	b'\xf5'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	958	panjang:	1	b'K'	
dikirim oleh	[	('192.168.122.179',	46737)	]	count:	959	panjang:	1	b'\xfc'	

c. Hasil output di alpine 1 (server)

< role	alpine-1	×	alpine-2	×	alpin >	-	×
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4839		panjang: 1 b'\x84'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4840		panjang: 1 b'\x86'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4841		panjang: 1 b'\xb1'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4842		panjang: 1 b'{'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4843		panjang: 1 b'\xff'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4844		panjang: 1 b'\x00'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4845		panjang: 1 b'R'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4846		panjang: 1 b'\xd4'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4847		panjang: 1 b'\xe6'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4848		panjang: 1 b'('		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4849		panjang: 1 b'p'		
dikirim oleh SERVER 1	[ ('192.168.122.179', 34468) ]		count: 4850		panjang: 1 b'{'		

7. **Multi\_thread\_async** : Jalankan code **udp\_multi\_thread\_async.py** pada alpine 3
- a. Hasil output di alpine 3

```
< alpine-1 x alpine-2 x alpine-3 x > - x
/Pemrograman_Jaringan_E/progjar3/jawaban_concurrency # python3 udp_multi_thread_
async.py
WARNING:root:image/jpeg
WARNING:root:writing kompas.jpg dalam waktu 0:00:02.137012 2021-07-19 16:44:00.875
270 s/d 2021-07-19 16:44:03.012287
mendownload https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0x0:998x66
5/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg
SERVER 1
192.168.122.129 5005 kompas.jpg
WARNING:root:image/jpeg
WARNING:root:writing detik.jpg dalam waktu 0:00:03.393281 2021-07-19 16:44:03.0130
87 s/d 2021-07-19 16:44:06.406373
mendownload https://awscdn.detik.net.id/detik2/images/logo.jpg
SERVER 2
192.168.122.77 5005 detik.jpg
Waktu TOTAL yang dibutuhkan 0:00:05.575929 detik 2021-07-19 16:44:00.875266 s/d 20
21-07-19 16:44:06.451195
hasil task yang dijalankan
{'kompas': None, 'detik': None}
/Pemrograman_Jaringan_E/progjar3/jawaban_concurrency #
```

- b. Hasil output di alpine 2 (server)

```
< alpine-1 x alpine-2 x alpine-3 x > - x
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3710 panjang: 1 b'\xde'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3711 panjang: 1 b'\xad'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3712 panjang: 1 b'\xac'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3713 panjang: 1 b'&'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3714 panjang: 1 b'\x1d'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3715 panjang: 1 b'H'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3716 panjang: 1 b'""'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3717 panjang: 1 b'\x1f'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3718 panjang: 1 b'\x80'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3719 panjang: 1 b'\x13'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3720 panjang: 1 b'\xd6'
SERVER 2
dikirim oleh [ ('192.168.122.179', 39831) ] count: 3721 panjang: 1 b's'
SERVER 2
```

c. Hasil output di alpine 1 (server)

< ole	alpine-1	×	alpine-2	×	alpin >	—	×
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21774	panjang:	1 b'\xa0'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21775	panjang:	1 b'\$'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21776	panjang:	1 b'\x95'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21777	panjang:	1 b'\x8a'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21778	panjang:	1 b'\xc2'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21779	panjang:	1 b'0'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21780	panjang:	1 b'^'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21781	panjang:	1 b'Y'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21782	panjang:	1 b'['	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21783	panjang:	1 b'F'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21784	panjang:	1 b'\t'	
SERVER 1							
dikirim oleh	[ ('192.168.122.179', 42898) ]		count:	21785	panjang:	1 b'\x98'	
SERVER 1							