Dokumentasi
Tugas Concurrency

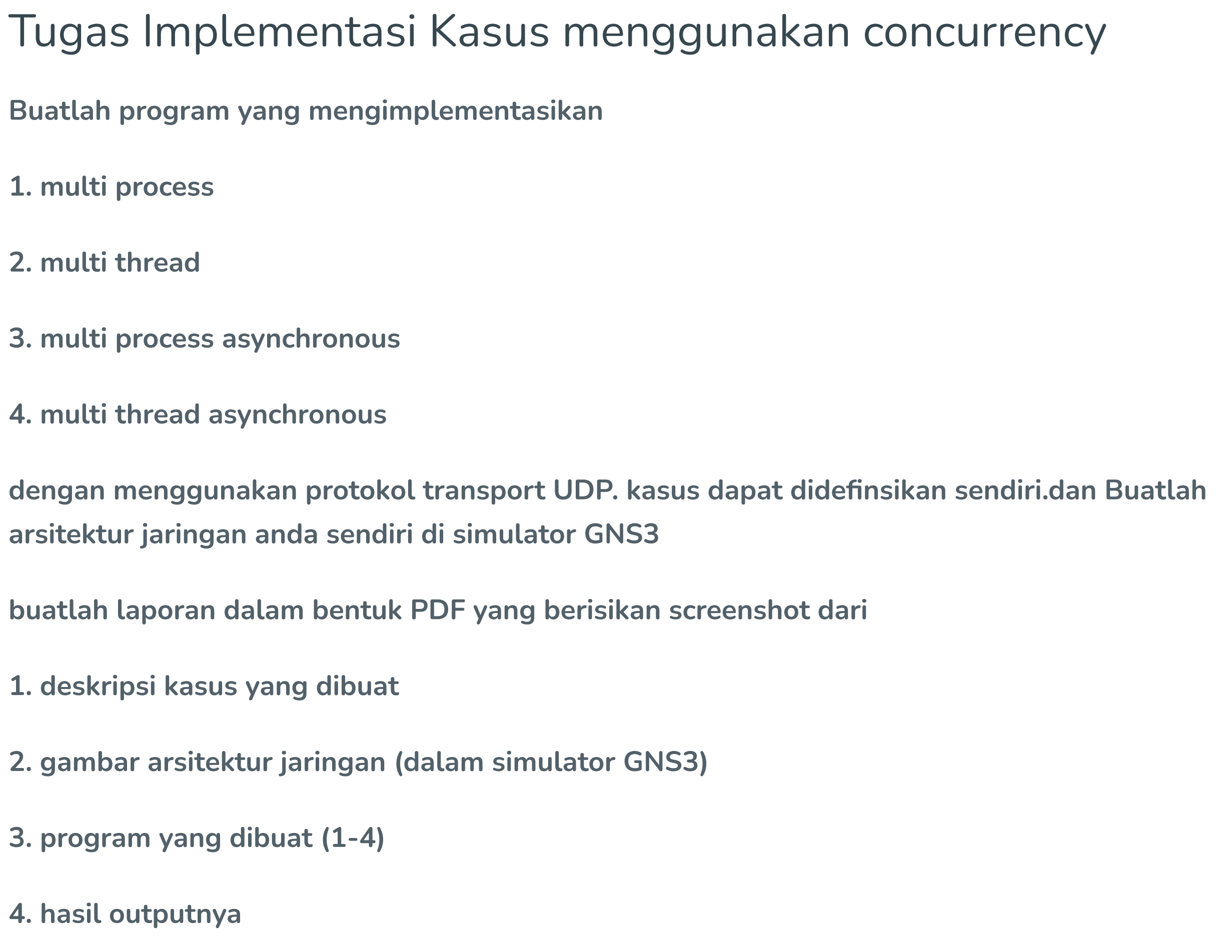
Pemrograman Jaringan E

Brananda Denta Wira Pranata - 05111840000143

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

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya

Perintah



Tugas Implementasi Kasus menggunakan concurrency

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 didefinsikan 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)

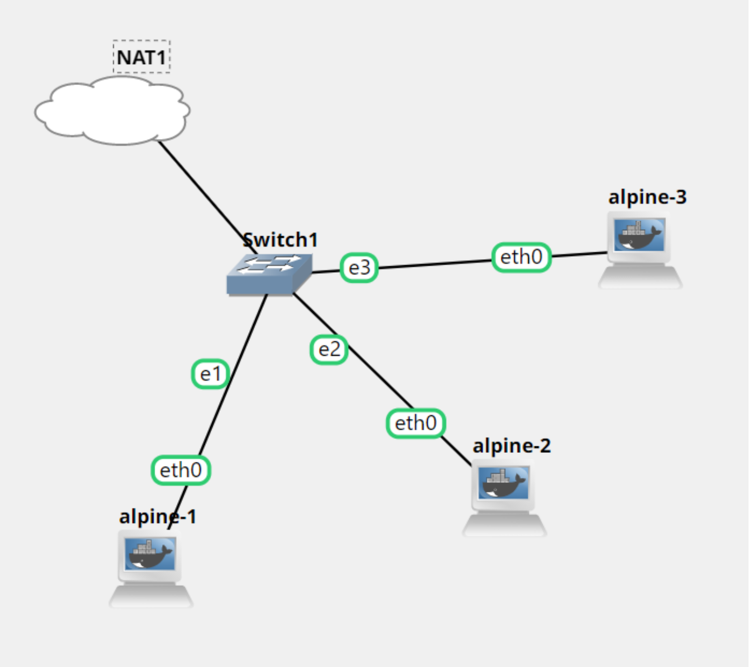
4. hasil outputnya

Deskripsi kasus

Pada tugas ini, diperintahkan untuk membuat program yang dapat melakukan pengiriman file menggunakan protokol transport UDP. Disini program saya mendownload 2 gambar terlebih dahulu lalu mengirim masing2 gambar ke server A dan server B. Pengiriman file ini diharuskan memakai 4 program yang berbeda yaitu dengan cara multi process, multi thread, multi process asynchronous, dan multi thread asynchronous.

Langkah Pengerjaan 1 :

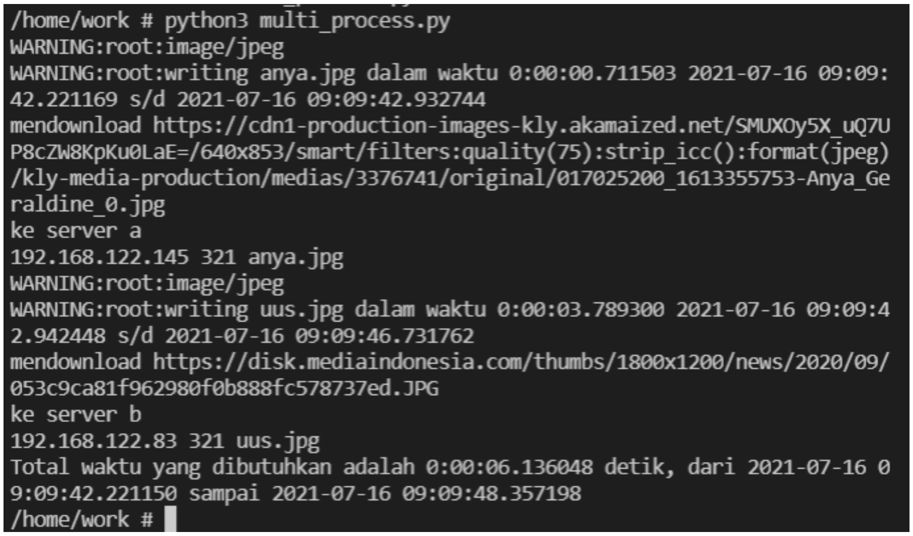
- **Untuk menyiapkan 1 client dan 1 server maka saya menggunakan 3 Alpine yaitu Alpine 1, Alpine 2, dan Alpine 3.**



- **Alpine 1 sebagai server a, Alpine 2 sebagai server b, dan Alpine 3 sebagai client.**
- **Setelah melakukan telnet pada terminal, selanjutnya membuat folder work di dalam folder home di semua alpine.**
- **Lalu untuk alpine 1 membuat program file bernama server_a.py.**
- **Untuk alpine 2 membuat program file bernama server_b.py.**
- **Untuk alpine 3 membuat file bernama library.py, multi_process.py, multi_process_async.py, multi_thread.py, dan multi_thread_async.py.**

Multi process

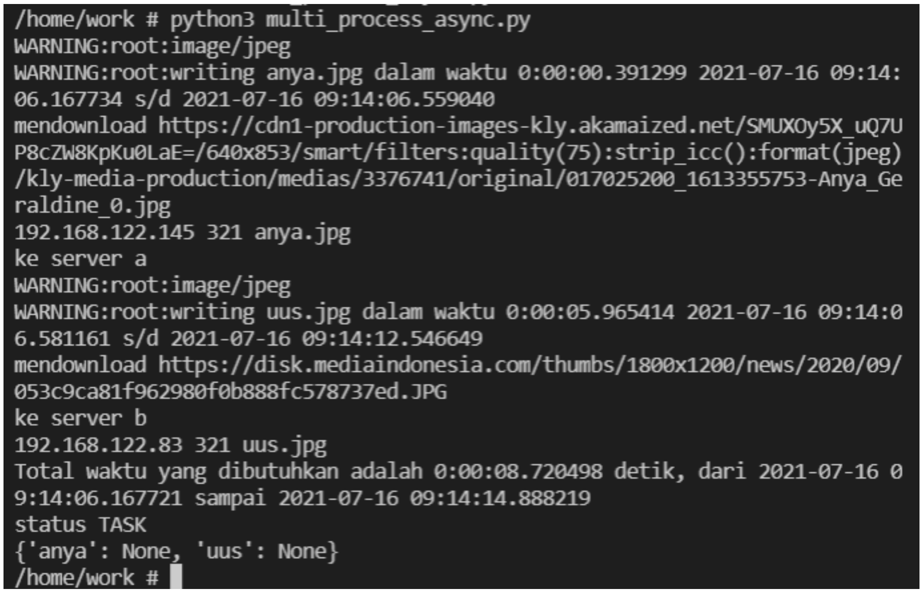
- **Selanjutnya jalankan program server_a pada Alpine 1**
- **lalu jalankan program server_b pada Alpine 2**
- **Lalu pada alpine 2 jalankan program multi_process.py.**



```
/home/work # python3 multi_process.py
WARNING:root:image/jpeg
WARNING:root:writing anya.jpg dalam waktu 0:00:00.711503 2021-07-16 09:09:
42.221169 s/d 2021-07-16 09:09:42.932744
mendownload https://cdn1-production-images-kly.akamaized.net/SMUXOy5X_uQ7U
P8cZW8KpKu0LaE=/640x853/smart/filters:quality(75):strip_icc():format(jpeg)
/kly-media-production/medias/3376741/original/017025200_1613355753-Anya_Ge
raldine_0.jpg
ke server a
192.168.122.145 321 anya.jpg
WARNING:root:image/jpeg
WARNING:root:writing uus.jpg dalam waktu 0:00:03.789300 2021-07-16 09:09:4
2.942448 s/d 2021-07-16 09:09:46.731762
mendownload https://disk.mediaindonesia.com/thumbs/1800x1200/news/2020/09/
053c9ca81f962980f0b888fc578737ed.JPG
ke server b
192.168.122.83 321 uus.jpg
Total waktu yang dibutuhkan adalah 0:00:06.136048 detik, dari 2021-07-16 0
9:09:42.221150 sampai 2021-07-16 09:09:48.357198
/home/work # 
```

Multi process asynchronous

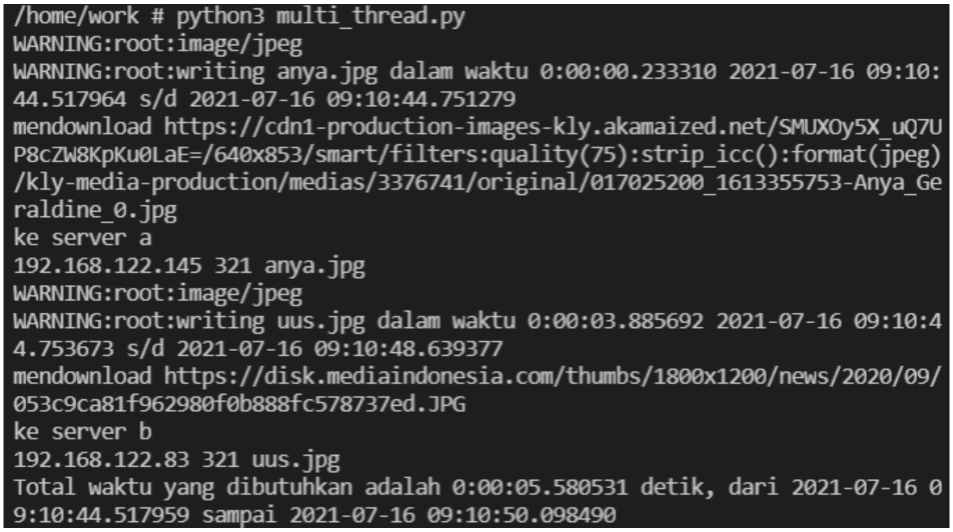
- **Selanjutnya jalankan program server_a pada Alpine 1**
- **lalu jalankan program server_b pada Alpine 2**
- **Lalu pada alpine 2 jalankan program multi_process_async.py.**



```
/home/work # python3 multi_process_async.py
WARNING:root:image/jpeg
WARNING:root:writing anya.jpg dalam waktu 0:00:00.391299 2021-07-16 09:14:
06.167734 s/d 2021-07-16 09:14:06.559040
mendownload https://cdn1-production-images-kly.akamaized.net/SMUXOy5X_uQ7U
P8cZW8KpKu0LaE=/640x853/smart/filters:quality(75):strip_icc():format(jpeg)
/kly-media-production/medias/3376741/original/017025200_1613355753-Anya_Ge
raldine_0.jpg
192.168.122.145 321 anya.jpg
ke server a
WARNING:root:image/jpeg
WARNING:root:writing uus.jpg dalam waktu 0:00:05.965414 2021-07-16 09:14:0
6.581161 s/d 2021-07-16 09:14:12.546649
mendownload https://disk.mediaindonesia.com/thumbs/1800x1200/news/2020/09/
053c9ca81f962980f0b888fc578737ed.JPG
ke server b
192.168.122.83 321 uus.jpg
Total waktu yang dibutuhkan adalah 0:00:08.720498 detik, dari 2021-07-16 0
9:14:06.167721 sampai 2021-07-16 09:14:14.888219
status TASK
{'anya': None, 'uus': None}
/home/work # 
```

Multi thread

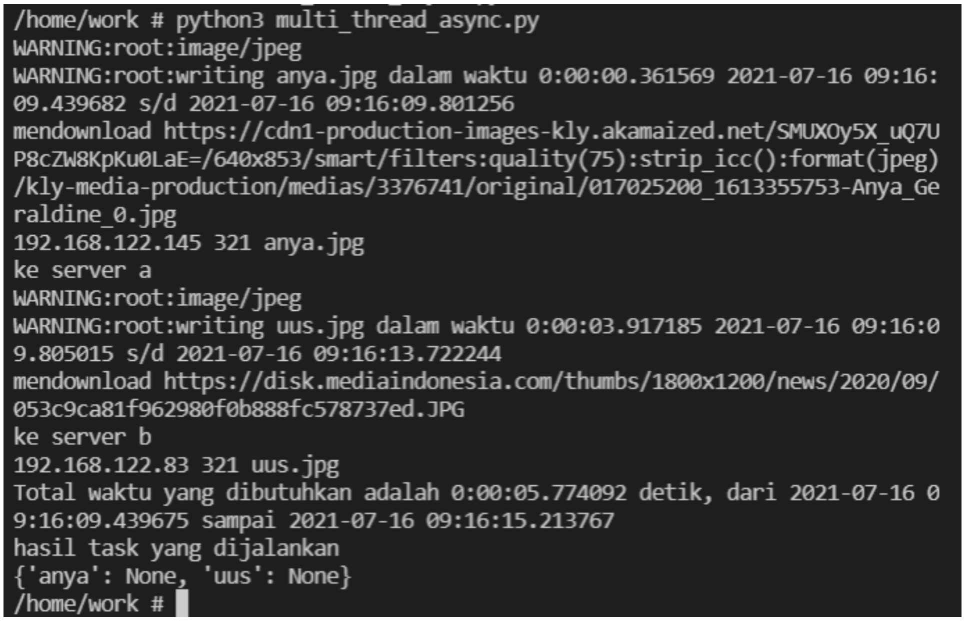
- **Selanjutnya jalankan program server_a pada Alpine 1**
- **lalu jalankan program server_b pada Alpine 2**
- **Lalu pada alpine 2 jalankan program multi_thread.py.**



```
/home/work # python3 multi_thread.py
WARNING:root:image/jpeg
WARNING:root:writing anya.jpg dalam waktu 0:00:00.233310 2021-07-16 09:10:
44.517964 s/d 2021-07-16 09:10:44.751279
mendownload https://cdn1-production-images-kly.akamaized.net/SMUXOy5X_uQ7U
P8cZW8KpKu0LaE=/640x853/smart/filters:quality(75):strip_icc():format(jpeg)
/kly-media-production/medias/3376741/original/017025200_1613355753-Anya_Ge
raldine_0.jpg
ke server a
192.168.122.145 321 anya.jpg
WARNING:root:image/jpeg
WARNING:root:writing uus.jpg dalam waktu 0:00:03.885692 2021-07-16 09:10:4
4.753673 s/d 2021-07-16 09:10:48.639377
mendownload https://disk.mediaindonesia.com/thumbs/1800x1200/news/2020/09/
053c9ca81f962980f0b888fc578737ed.JPG
ke server b
192.168.122.83 321 uus.jpg
Total waktu yang dibutuhkan adalah 0:00:05.580531 detik, dari 2021-07-16 0
9:10:44.517959 sampai 2021-07-16 09:10:50.098490
```

Multi thread asynchronous

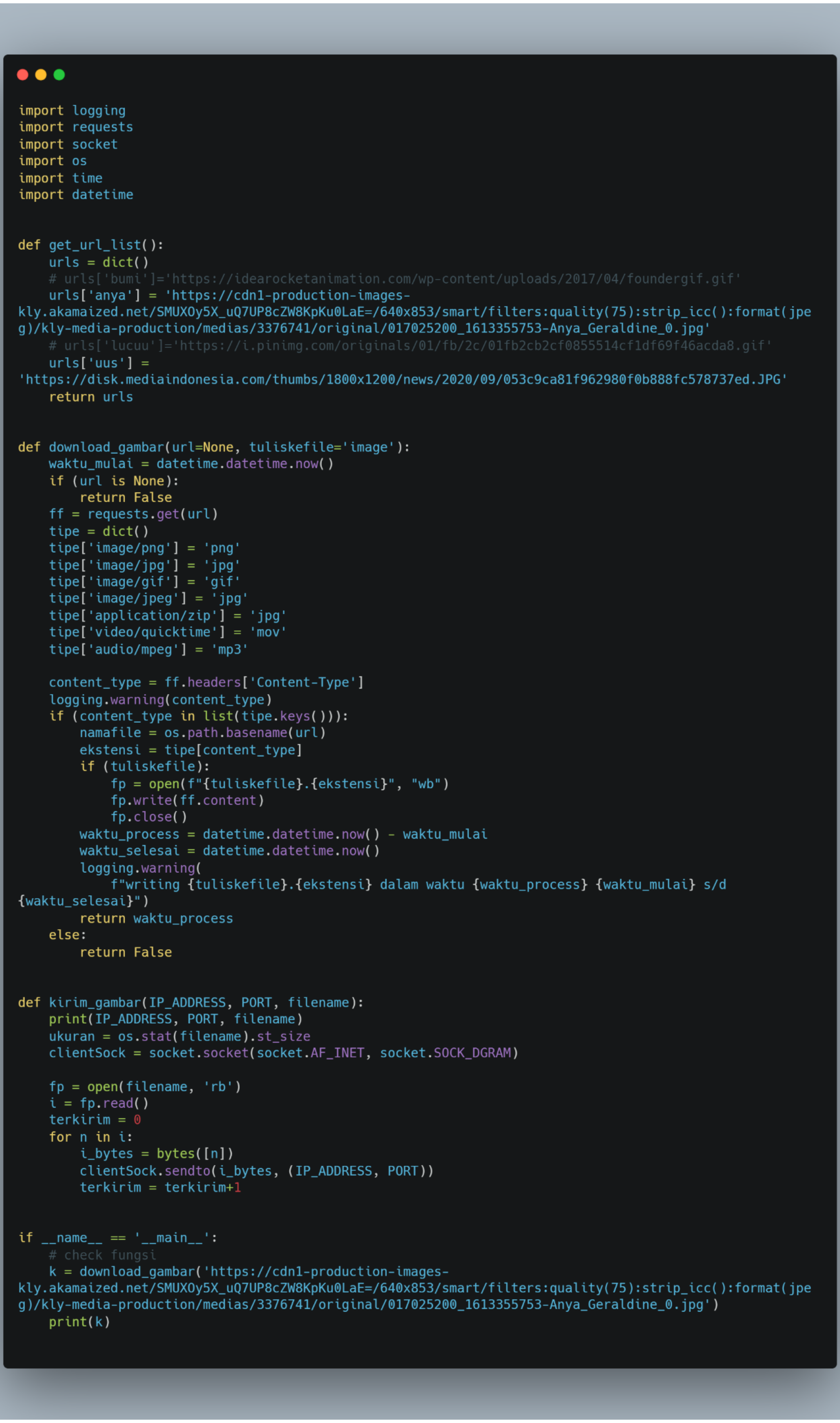
- **Selanjutnya jalankan program server_a pada Alpine 1**
- **lalu jalankan program server_b pada Alpine 2**
- **Lalu pada alpine 2 jalankan program multi_thread_async.py.**



```
/home/work # python3 multi_thread_async.py
WARNING:root:image/jpeg
WARNING:root:writing anya.jpg dalam waktu 0:00:00.361569 2021-07-16 09:16:
09.439682 s/d 2021-07-16 09:16:09.801256
mendownload https://cdn1-production-images-kly.akamaized.net/SMUXOy5X_uQ7U
P8cZW8KpKu0LaE=/640x853/smart/filters:quality(75):strip_icc():format(jpeg)
/kly-media-production/medias/3376741/original/017025200_1613355753-Anya_Ge
raldine_0.jpg
192.168.122.145 321 anya.jpg
ke server a
WARNING:root:image/jpeg
WARNING:root:writing uus.jpg dalam waktu 0:00:03.917185 2021-07-16 09:16:0
9.805015 s/d 2021-07-16 09:16:13.722244
mendownload https://disk.mediaindonesia.com/thumbs/1800x1200/news/2020/09/
053c9ca81f962980f0b888fc578737ed.JPG
ke server b
192.168.122.83 321 uus.jpg
Total waktu yang dibutuhkan adalah 0:00:05.774092 detik, dari 2021-07-16 0
9:16:09.439675 sampai 2021-07-16 09:16:15.213767
hasil task yang dijalankan
{'anya': None, 'uus': None}
/home/work #
```

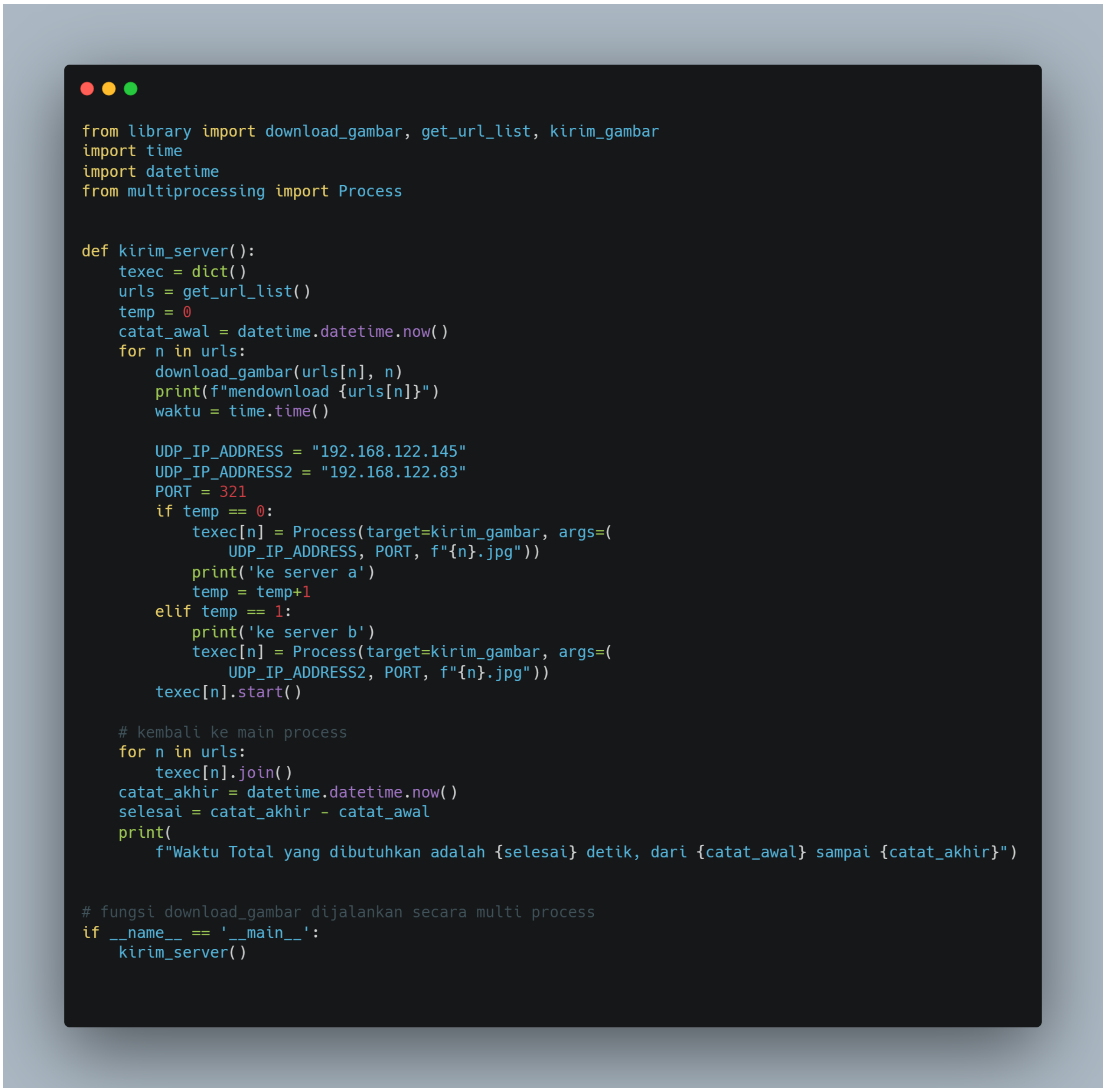
Program

Library.py



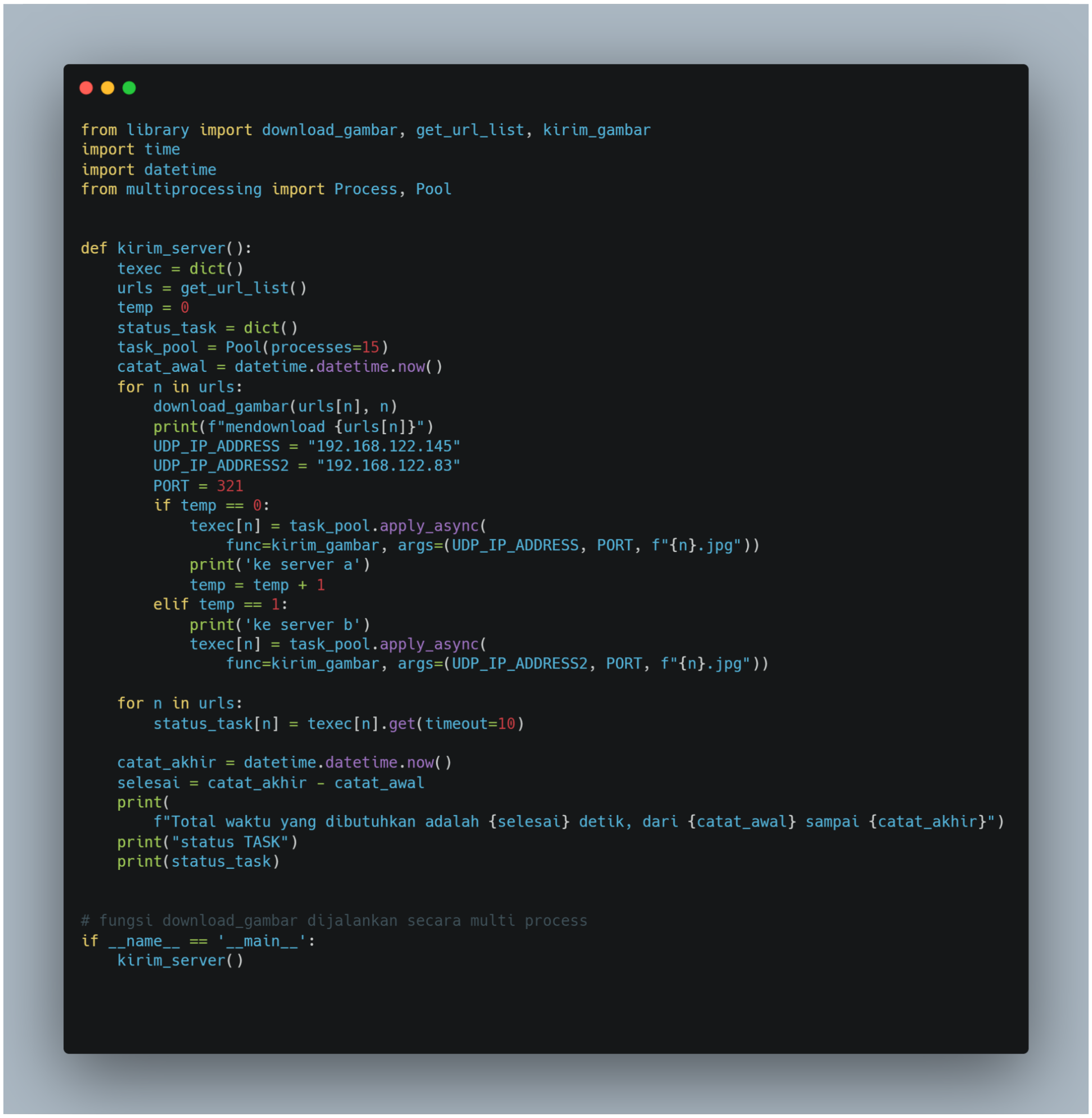
```python
import logging
import requests
import socket
import os
import time
import datetime


def get_url_list():
    urls = dict()
    # urls['bumi']='https://idearocketanimation.com/wp-content/uploads/2017/04/foundergif.gif'
    urls['anya'] = 'https://cdn1-production-images-
kly.akamaized.net/SMUXOy5X_uQ7UP8cZW8KpKu0LaE=/640x853/smart/filters:quality(75):strip_icc():format(jpe
g)/kly-media-production/medias/3376741/original/017025200_1613355753-Anya_Geraldine_0.jpg'
    # urls['lucuu']='https://i.pinimg.com/originals/01/fb/2c/01fb2cb2cf0855514cf1df69f46acda8.gif'
    urls['uus'] =
'https://disk.mediaindonesia.com/thumbs/1800x1200/news/2020/09/053c9ca81f962980f0b888fc578737ed.JPG'
    return urls


def download_gambar(url=None, tuliskefile='image'):
    waktu_mulai = 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'
    tipe['audio/mpeg'] = 'mp3'

    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_mulai
        waktu_selesai = datetime.datetime.now()
        logging.warning(
            f"writing {tuliskefile}.{ekstensi} dalam waktu {waktu_process} {waktu_mulai} s/d
{waktu_selesai}")
        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')
    i = fp.read()
    terkirim = 0
    for n in i:
        i_bytes = bytes([n])
        clientSock.sendto(i_bytes, (IP_ADDRESS, PORT))
        terkirim = terkirim+1


if __name__ == '__main__':
    # check fungsi
    k = download_gambar('https://cdn1-production-images-
kly.akamaized.net/SMUXOy5X_uQ7UP8cZW8KpKu0LaE=/640x853/smart/filters:quality(75):strip_icc():format(jpe
g)/kly-media-production/medias/3376741/original/017025200_1613355753-Anya_Geraldine_0.jpg')
    print(k)
```

Multi_process.py



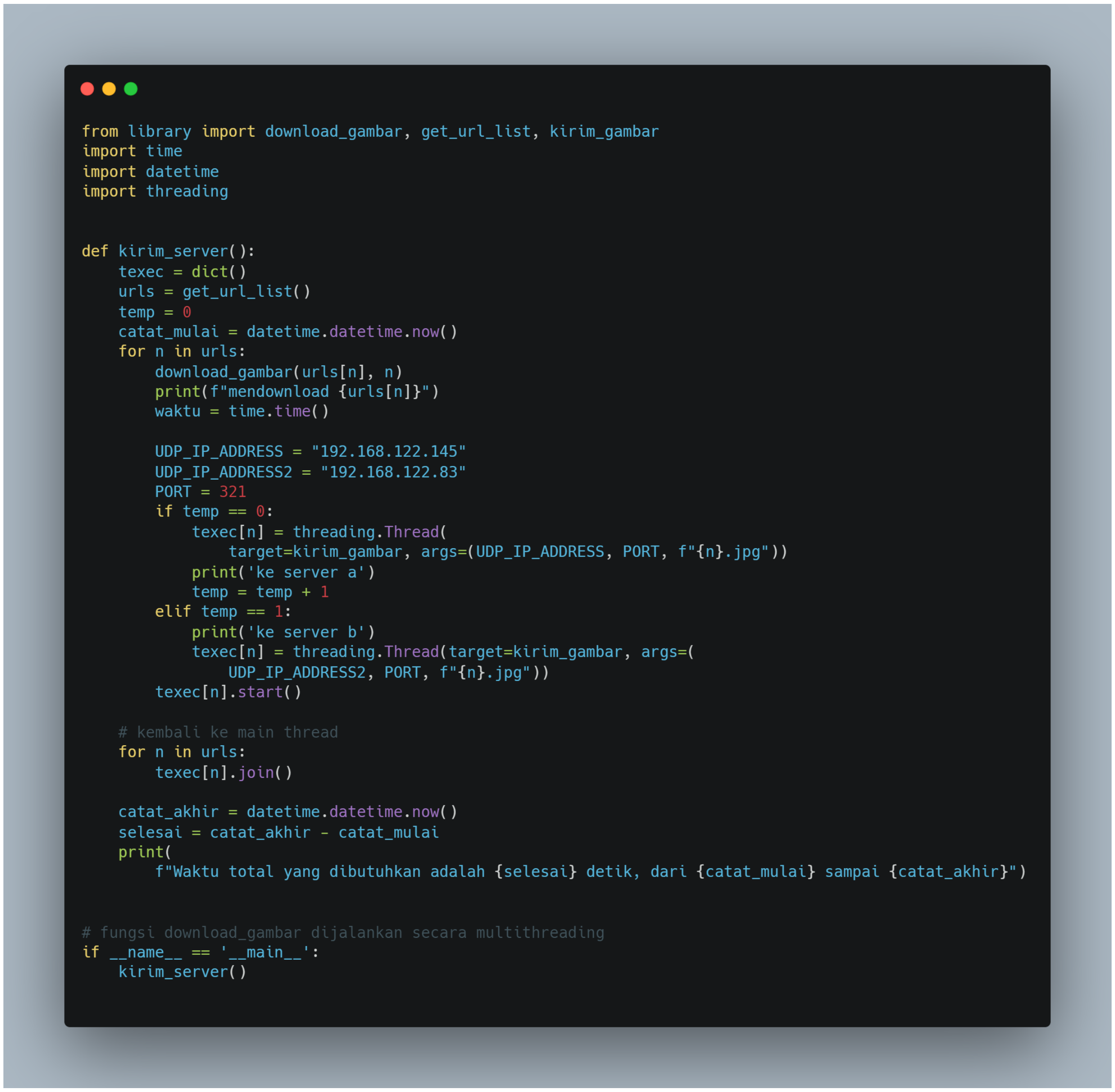
```python
from library import download_gambar, get_url_list, kirim_gambar
import time
import datetime
from multiprocessing import Process


def kirim_server():
    texec = dict()
    urls = get_url_list()
    temp = 0
    catat_awal = datetime.datetime.now()
    for n in urls:
        download_gambar(urls[n], n)
        print(f"mendownload {urls[n]}")
        waktu = time.time()

        UDP_IP_ADDRESS = "192.168.122.145"
        UDP_IP_ADDRESS2 = "192.168.122.83"
        PORT = 321
        if temp == 0:
            texec[n] = Process(target=kirim_gambar, args=(
                UDP_IP_ADDRESS, PORT, f"{n}.jpg"))
            print('ke server a')
            temp = temp+1
        elif temp == 1:
            print('ke server b')
            texec[n] = Process(target=kirim_gambar, args=(
                UDP_IP_ADDRESS2, PORT, f"{n}.jpg"))
        texec[n].start()

    # kembali ke main process
    for n in urls:
        texec[n].join()
    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
    print(
        f"Waktu Total yang dibutuhkan adalah {selesai} detik, dari {catat_awal} sampai {catat_akhir}")


# fungsi download_gambar dijalankan secara multi process
if __name__ == '__main__':
    kirim_server()
```

Multi_process_asyn.py



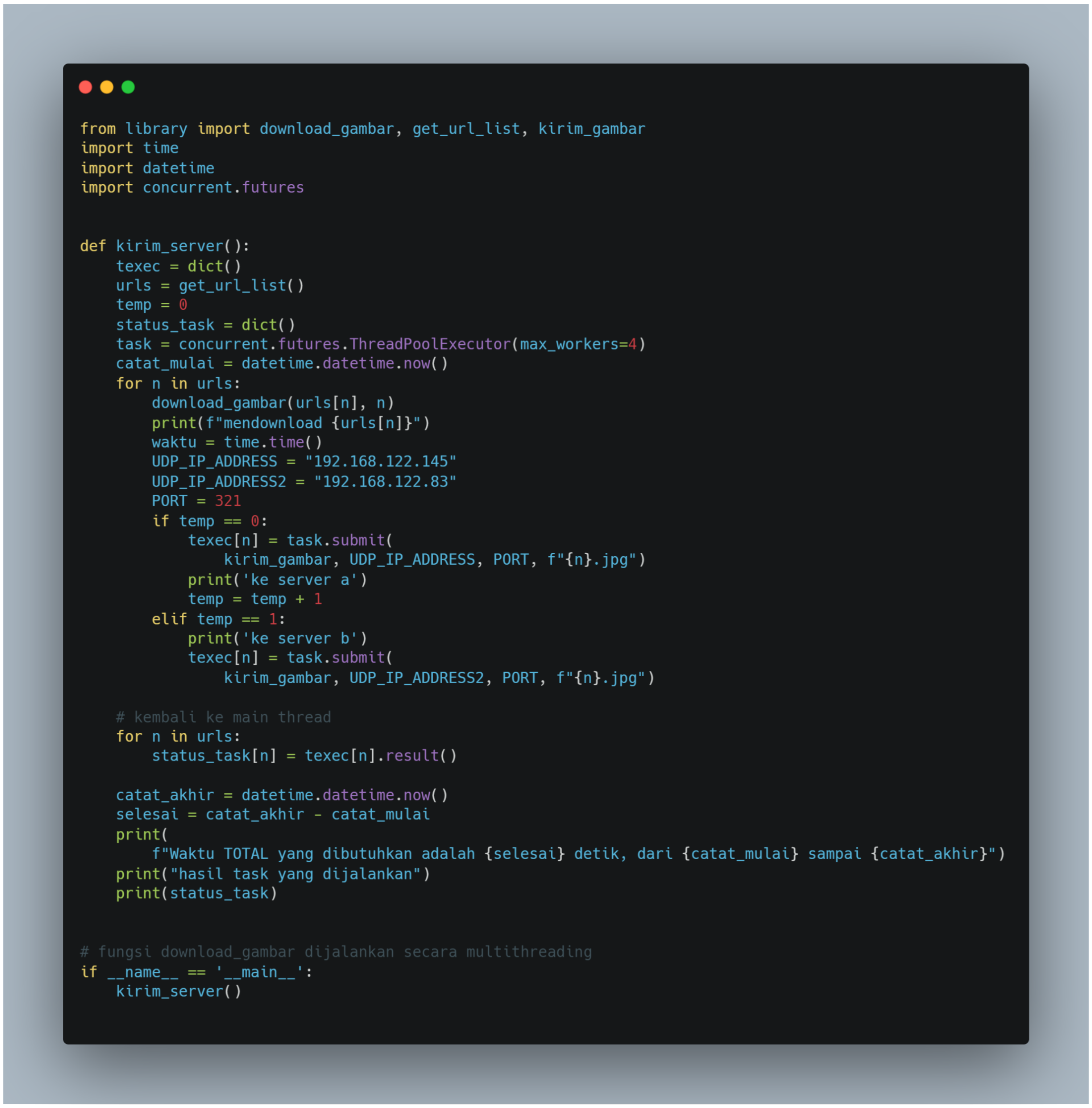
```python
from library import download_gambar, get_url_list, kirim_gambar
import time
import datetime
from multiprocessing import Process, Pool


def kirim_server():
    texec = dict()
    urls = get_url_list()
    temp = 0
    status_task = dict()
    task_pool = Pool(processes=15)
    catat_awal = datetime.datetime.now()
    for n in urls:
        download_gambar(urls[n], n)
        print(f"mendownload {urls[n]}")
        UDP_IP_ADDRESS = "192.168.122.145"
        UDP_IP_ADDRESS2 = "192.168.122.83"
        PORT = 321
        if temp == 0:
            texec[n] = task_pool.apply_async(
                func=kirim_gambar, args=(UDP_IP_ADDRESS, PORT, f"{n}.jpg"))
            print('ke server a')
            temp = temp + 1
        elif temp == 1:
            print('ke server b')
            texec[n] = task_pool.apply_async(
                func=kirim_gambar, args=(UDP_IP_ADDRESS2, PORT, f"{n}.jpg"))

    for n in urls:
        status_task[n] = texec[n].get(timeout=10)

    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
    print(
        f"Total waktu yang dibutuhkan adalah {selesai} detik, dari {catat_awal} sampai {catat_akhir}")
    print("status TASK")
    print(status_task)


# fungsi download_gambar dijalankan secara multi process
if __name__ == '__main__':
    kirim_server()
```

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_mulai = datetime.datetime.now()
    for n in urls:
        download_gambar(urls[n], n)
        print(f"mendownload {urls[n]}")
        waktu = time.time()

        UDP_IP_ADDRESS = "192.168.122.145"
        UDP_IP_ADDRESS2 = "192.168.122.83"
        PORT = 321
        if temp == 0:
            texec[n] = threading.Thread(
                target=kirim_gambar, args=(UDP_IP_ADDRESS, PORT, f"{n}.jpg"))
            print('ke server a')
            temp = temp + 1
        elif temp == 1:
            print('ke server b')
            texec[n] = threading.Thread(target=kirim_gambar, args=(
                UDP_IP_ADDRESS2, PORT, f"{n}.jpg"))
        texec[n].start()

    # kembali ke main thread
    for n in urls:
        texec[n].join()

    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_mulai
    print(
        f"Waktu total yang dibutuhkan adalah {selesai} detik, dari {catat_mulai} sampai {catat_akhir}")


# fungsi download_gambar dijalankan secara multithreading
if __name__ == '__main__':
    kirim_server()
```

Multi_thread_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()
    temp = 0
    status_task = dict()
    task = concurrent.futures.ThreadPoolExecutor(max_workers=4)
    catat_mulai = datetime.datetime.now()
    for n in urls:
        download_gambar(urls[n], n)
        print(f"mendownload {urls[n]}")
        waktu = time.time()
        UDP_IP_ADDRESS = "192.168.122.145"
        UDP_IP_ADDRESS2 = "192.168.122.83"
        PORT = 321
        if temp == 0:
            texec[n] = task.submit(
                kirim_gambar, UDP_IP_ADDRESS, PORT, f"{n}.jpg")
            print('ke server a')
            temp = temp + 1
        elif temp == 1:
            print('ke server b')
            texec[n] = task.submit(
                kirim_gambar, UDP_IP_ADDRESS2, PORT, f"{n}.jpg")

    # kembali ke main thread
    for n in urls:
        status_task[n] = texec[n].result()

    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_mulai
    print(
        f"Waktu TOTAL yang dibutuhkan adalah {selesai} detik, dari {catat_mulai} sampai {catat_akhir}")
    print("hasil task yang dijalankan")
    print(status_task)


# fungsi download_gambar dijalankan secara multithreading
if __name__ == '__main__':
    kirim_server()
```