Tugas 4a Progjar D

Nama : Rimas Muhammad Irfansyah

NRP : 05111740000185

Tugas:

Implementasi program server sesuai contoh

Semua tugas berikut ini HARUS dijalankan di SIMULATOR

1. Jalankan program server seperti dalam pembahasan

2. Buatlah program client yang dapat melakukan 100 request get pada dalam satu saat untuk operasi get file "pokijan.jpg"

3. Capture dan submitlah poin 1 dan 2 dalam satu dokumen pdf. berikan DESKRIPSI dan PENJELASAN

Jawab:

Untuk melakukan 100 request get dalam satu saat dapat menggunakan multiprocessing.

Script file_server.py:

```
from socket import *
import socket
import threading
import logging
import time
import sys


from file_protocol import  FileProtocol
fp = FileProtocol()


class ProcessTheClient(threading.Thread):
    def __init__(self, connection, address):
        self.connection = connection
        self.address = address
        threading.Thread.__init__(self)

    def run(self):
        while True:
            data = self.connection.recv(32)
            if data:
                d = data.decode()
                hasil = fp.proses_string(d)
                hasil=hasil+"\r\n\r\n"
                self.connection.sendall(hasil.encode())
            else:
                break
        self.connection.close()
```

```
class Server(threading.Thread):
    def __init__(self,ipaddress='192.168.122.38',port=80):
        self.ipinfo=(ipaddress,port)
        self.the_clients = []
        self.my_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.my_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR,
1)
        threading.Thread.__init__(self)

    def run(self):
        logging.warning(f"server berjalan di ip address {self.ipinfo}")
        self.my_socket.bind(self.ipinfo)
        self.my_socket.listen(1)
        while True:
            self.connection, self.client_address = self.my_socket.accept()
            logging.warning(f"connection from {self.client_address}")

            clt = ProcessTheClient(self.connection, self.client_address)
            clt.start()
            self.the_clients.append(clt)


def main():
    svr = Server(ipaddress='192.168.122.38',port=80)
    svr.start()


if __name__ == "__main__":
    main()
```

Script file_client_cli.py:

```
import socket
import json
import base64
import logging

server_address=('192.168.122.38',80)

def send_command(command_str=""):
    global server_address
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect(server_address)
    logging.warning(f"connecting to {server_address}")
    try:
        logging.warning(f"sending message ")
        sock.sendall(command_str.encode())
        # Look for the response, waiting until socket is done (no more
data)
        data_received="" #empty string
        while True:
            #socket does not receive all data at once, data comes in part,
need to be concatenated at the end of process
            data = sock.recv(16)
            if data:
                #data is not empty, concat with previous content
                data_received += data.decode()
                if "\r\n\r\n" in data_received:
                    break
            else:
                # no more data, stop the process by break
                break
        # at this point, data_received (string) will contain all data
coming from the socket
        # to be able to use the data_received as a dict, need to load it
using json.loads()
        hasil = json.loads(data_received)
        logging.warning("data received from server:")
        return hasil
    except:
        logging.warning("error during data receiving")
        return False


def remote_list():
    command_str=f"LIST"
    hasil = send_command(command_str)
    if (hasil['status']=='OK'):
        print("daftar file : ")
        for nmfile in hasil['data']:
            print(f"- {nmfile}")
        return True
    else:
        print("Gagal")
        return False

def remote_get(filename=""):
    command_str=f"GET {filename}"
    hasil = send_command(command_str)
    if (hasil['status']=='OK'):
        #proses file dalam bentuk base64 ke bentuk bytes
        namafile= hasil['data_namafile']
        isifile = base64.b64decode(hasil['data_file'])
        fp = open(namafile,'wb+')
        fp.write(isifile)
        fp.close()
        return True
    else:
        print("Gagal")
        return False


if __name__=='__main__':
    server_address=('192.168.122.38',80)
    remote_list()
    #remote_get('donalbebek.jpg')
```

Script multi_process.py:

```
from file_client_cli import remote_get
import time
import datetime
from multiprocessing import Process

def download_semua():
    texec = dict()
    filename = pokijan.jpg
    catat_awal = datetime.datetime.now()
    for k in range(100):
        print(f"mendownload {k}")
        waktu = time.time()
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi 
download gambar secara multiprocess
        texec[k] = Process(target=remote_get, args=(filename,))
        texec[k].start()
    #setelah menyelesaikan tugasnya, dikembalikan ke main process dengan 
join
    for k in range(100):
        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()
```

Script file_protocol.py:

```
import json
import logging

from file_interface import FileInterface

"""
* class FileProtocol bertugas untuk memproses 
data yang masuk, dan menerjemahkannya apakah sesuai dengan
protokol/aturan yang dibuat

* data yang masuk dari client adalah dalam bentuk bytes yang 
pada akhirnya akan diproses dalam bentuk string

* class FileProtocol akan memproses data yang masuk dalam bentuk 
string
"""



class FileProtocol:
    def __init__(self):
        self.file = FileInterface()
    def proses_string(self,string_datamasuk=''):
        logging.warning(f"string diproses: {string_datamasuk}")
        c = string_datamasuk.split(" ")
        try:
            c_request = c[0].strip()
            logging.warning(f"memproses request: {c_request}")
            if (c_request=='LIST'):
                return json.dumps(self.file.list())
            elif (c_request=='GET'):
                param1 = c[1].strip()
                return json.dumps(self.file.get(param1))
            else:
                return json.dumps(dict(status='ERROR', data='request tidak 
dikenali'))
        except Exception:
            return json.dumps(dict(status='ERROR',data='request tidak 
dikenali'))
```

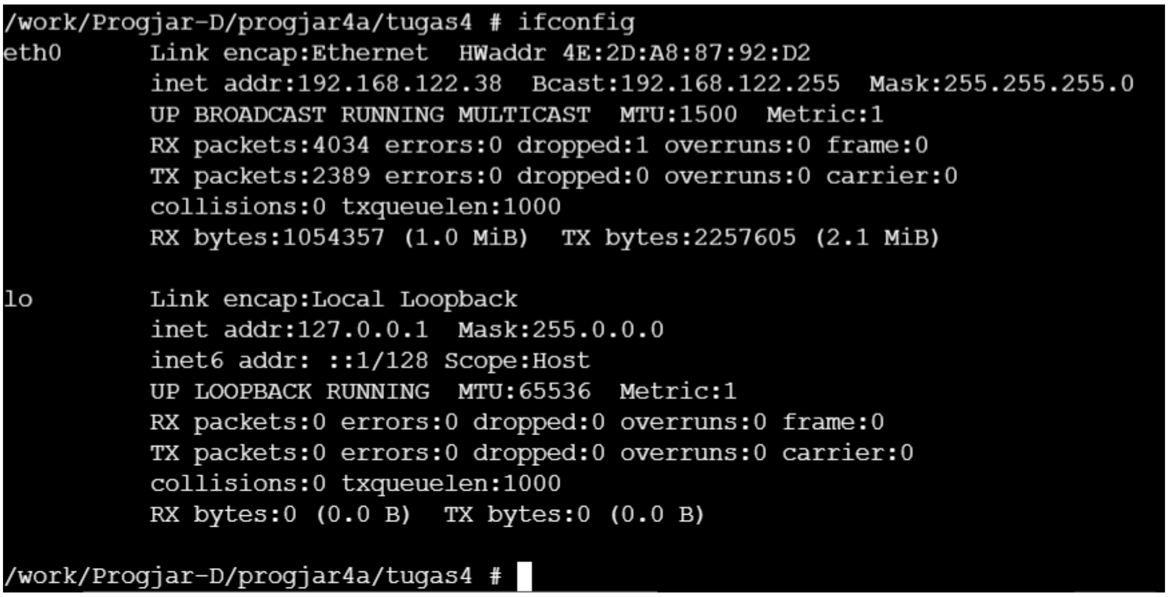
```
if __name__=='__main__':
    #contoh pemakaian
    fp = FileProtocol()
    print(fp.proses_string("LIST"))
    print(fp.proses_string("GET pokijan.jpg"))
```

Script file_interface.py:

```
import os
import json
import base64
from glob import glob


class FileInterface:
    def __init__(self):
        os.chdir('files/')

    def list(self):
        try:
            filelist = glob('*.*')
            return dict(status='OK',data=filelist)
        except Exception as e:
            return dict(status='ERROR',data=str(e))

    def get(self,filename=''):
        if(filename==''):
            return None
        try:
            fp = open(f"{filename}",'rb')
            isifile = base64.b64encode(fp.read()).decode()
            return
dict(status='OK',data_namafile=filename,data_file=isifile)
        except Exception as e:
            return dict(status='ERROR',data=str(e))



if __name__=='__main__':
    f = FileInterface()
    print(f.list())
    print(f.get('pokijan.jpg'))
```

Script diatas sudah diubah IP nya sesuai dengan ip yang digunakan pada node alpine-1 sebagai server.

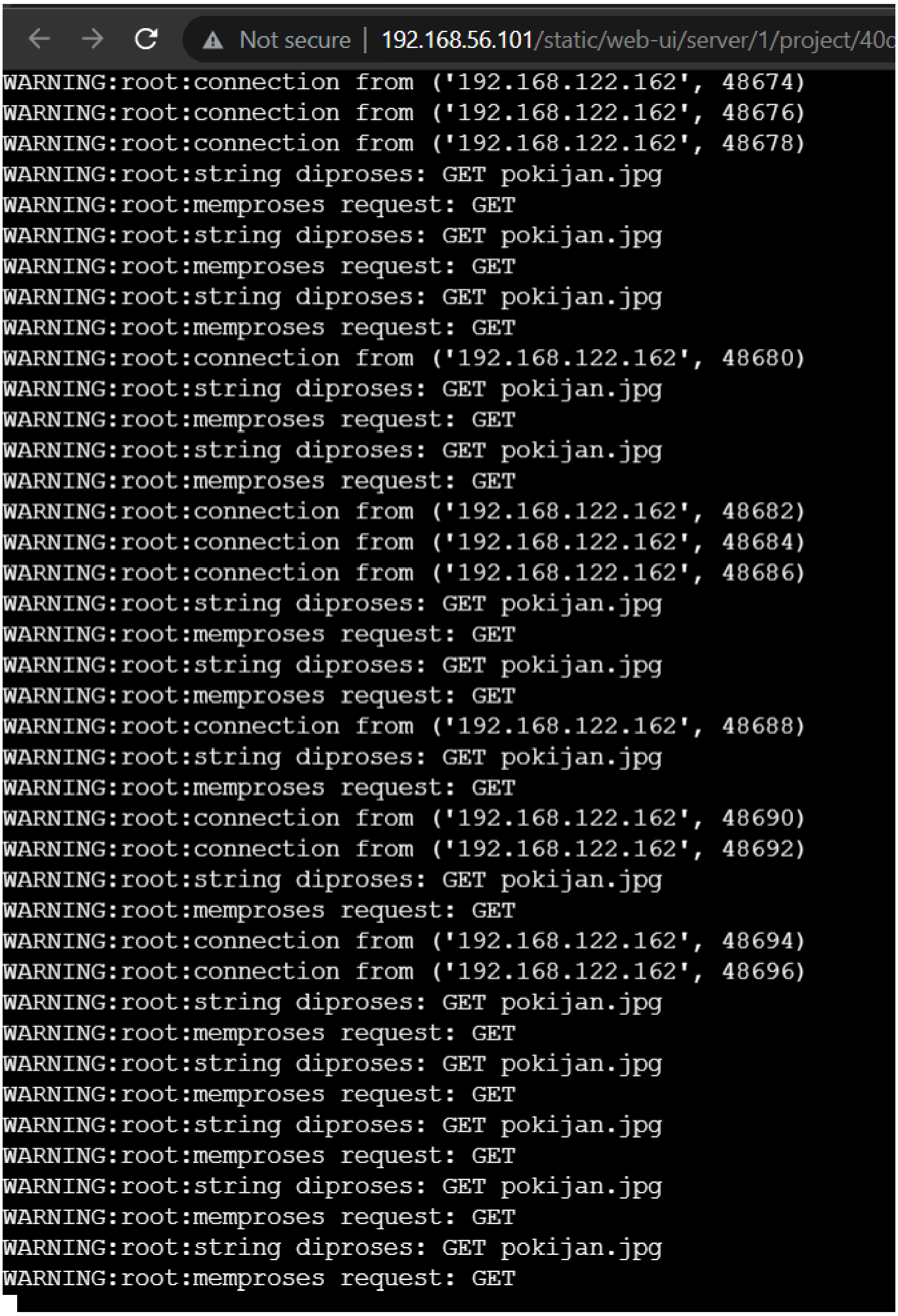
Ip pada alpine-1:



```
/work/Progjar-D/progjar4a/tugas4 # ifconfig
eth0      Link encap:Ethernet  HWaddr 4E:2D:A8:87:92:D2
          inet addr:192.168.122.38  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4034 errors:0 dropped:1 overruns:0 frame:0
          TX packets:2389 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1054357 (1.0 MiB)  TX bytes:2257605 (2.1 MiB)

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)

/work/Progjar-D/progjar4a/tugas4 # 
```

Output:

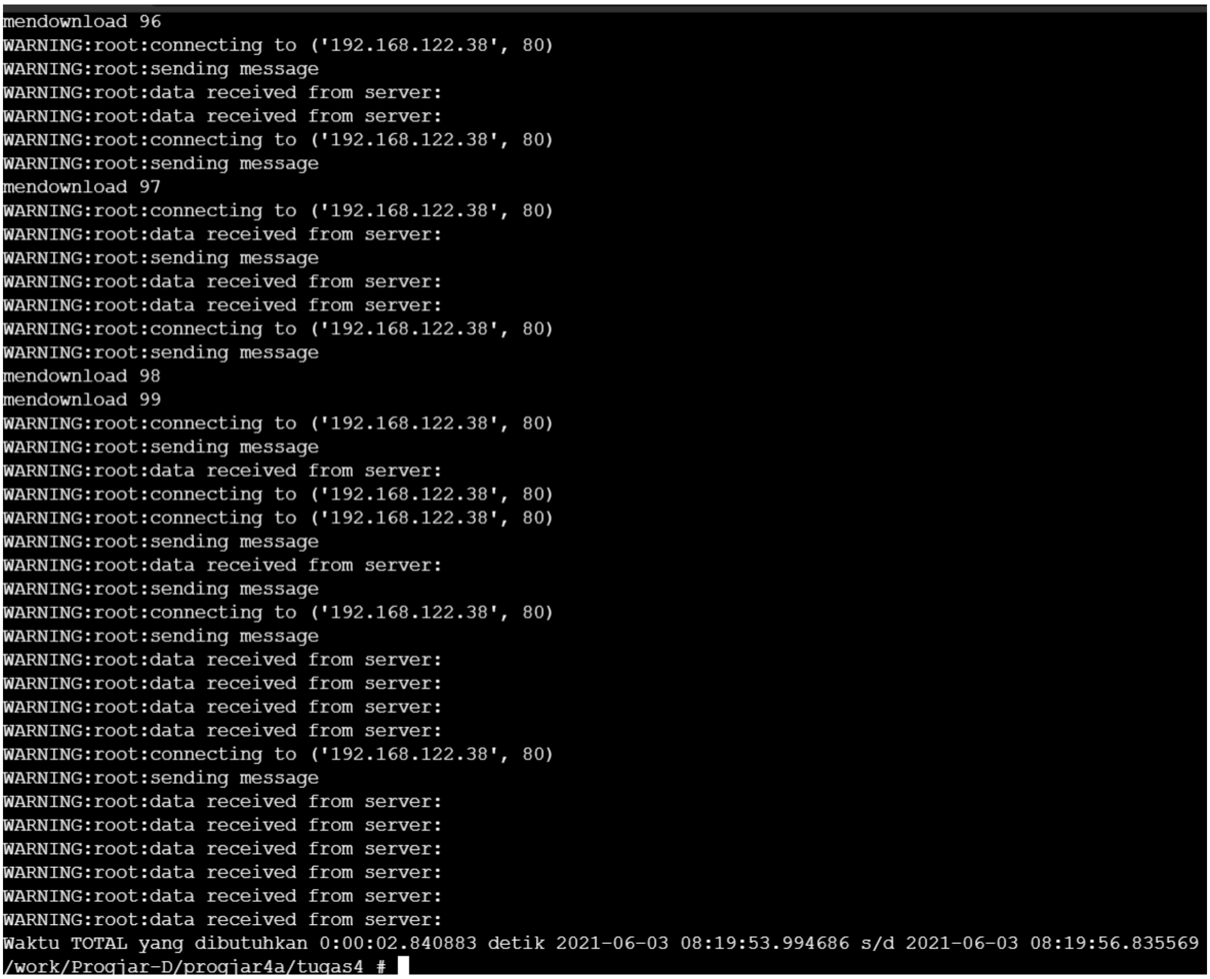
Alpine-1:


Not secure | 192.168.56.101/static/web-ui/server/1/project/40c

```
WARNING:root:connection from ('192.168.122.162', 48674)
WARNING:root:connection from ('192.168.122.162', 48676)
WARNING:root:connection from ('192.168.122.162', 48678)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:connection from ('192.168.122.162', 48680)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:connection from ('192.168.122.162', 48682)
WARNING:root:connection from ('192.168.122.162', 48684)
WARNING:root:connection from ('192.168.122.162', 48686)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:connection from ('192.168.122.162', 48688)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:connection from ('192.168.122.162', 48690)
WARNING:root:connection from ('192.168.122.162', 48692)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:connection from ('192.168.122.162', 48694)
WARNING:root:connection from ('192.168.122.162', 48696)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
```

Alpine-1 menerima request dari client (alpine-2) untuk mengirim file pokijan.jpg.

Alpine-2:



```
mendownload 96
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:sending message
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:sending message
mendownload 97
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:data received from server:
WARNING:root:sending message
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:sending message
mendownload 98
mendownload 99
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:sending message
WARNING:root:data received from server:
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:sending message
WARNING:root:data received from server:
WARNING:root:sending message
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:sending message
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:connecting to ('192.168.122.38', 80)
WARNING:root:sending message
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:data received from server:
WARNING:root:data received from server:
Waktu TOTAL yang dibutuhkan 0:00:02.840883 detik 2021-06-03 08:19:53.994686 s/d 2021-06-03 08:19:56.835569
/work/Progjar-D/progjar4a/tugas4 #
```

Alpine 2 mengirim request file pokijan.jpg pada server