

Soal

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

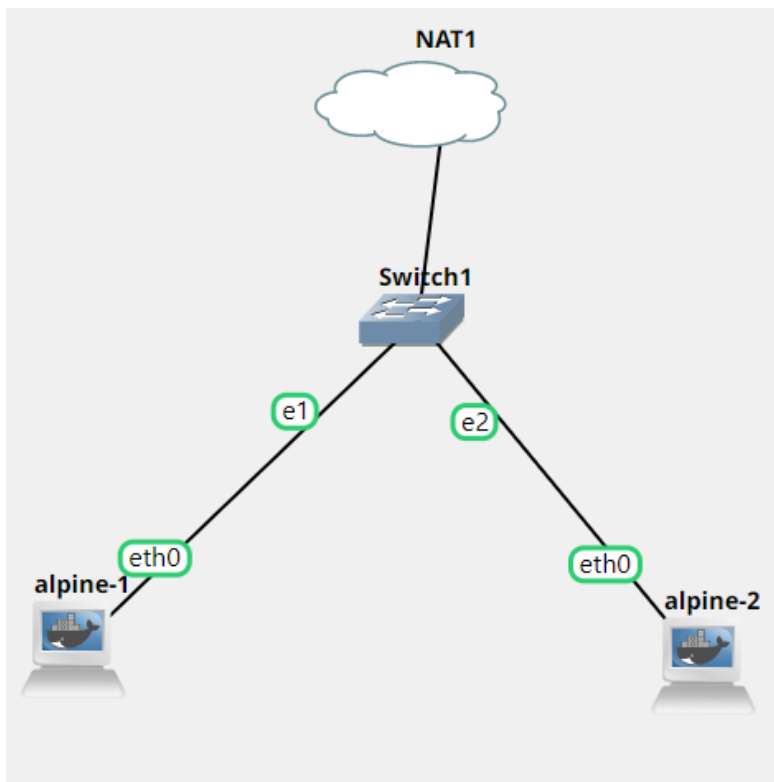
1. Deskripsi kasus yang dibuat
2. Gambar arsitektur jaringan (dalam simulator GNS3)
3. Program yang dibuat (1-4)
4. Hasil outputnya

Jawab

1. Kasus

Pada sebuah jaringan komputer terdapat beberapa server dan beberapa client. Client akan melakukan request file ke server berupa mengirim file gambar, lalu server dengan alamat broadcast yang sama akan menerima gambar yang di kirimkan oleh client.

2. Gambar Arsitektur



3. Program yang di buat

- Multi Process

```
def kirim_multi_process_sync(daftar=None):
    if (daftar is None):
        return False
    f = open(daftar, "rb")
    l = f.read(1024)
    while (l):
        if (sock.sendto(l, (TARGET_IP, TARGET_PORT))):
            l = f.read(1024)
    f.close()

def multi_process_sync():
    texec = dict()
    daftar = ['testing1.jpg', 'testing2.jpg']

    catat_awal = datetime.datetime.now()
    for k in range(len(daftar)):
        print(f"mengirim {daftar[k]}")
        texec[k] = Process(target=kirim_multi_process_sync, args=(daftar[k],))
        texec[k].start()
    for k in range(len(daftar)):
        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}")
```

- Multi Thread

```
def kirim_multi_thread_sync(daftar=None):
    if (daftar is None):
        return False
    f = open(daftar, "rb")
    l = f.read(1024)
    while (l):
        if (sock.sendto(l, (TARGET_IP, TARGET_PORT))):
            l = f.read(1024)
    f.close()

def multi_thread_sync():
    texec = dict()
    daftar = ['testing1.png', 'testing2.jpeg']

    catat_awal = datetime.datetime.now()
    for k in range(len(daftar)):
        print(f"mengirim {daftar[k]}")
        texec[k] = threading.Thread(target=kirim_multi_thread_sync, args=(daftar[k],))
        texec[k].start()
    for k in range(len(daftar)):
        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}")
```

- Multi Process Asynchronous

```
def kirim_multi_process_async(daftar=None):
    if (daftar is None):
        return False
    f = open(daftar, "rb")
    l = f.read(1024)
    while (l):
        if (sock.sendto(l, (TARGET_IP, TARGET_PORT))):
            l = f.read(1024)
    f.close()

def multi_process_async():
    texec = dict()
    daftar = ['testing1.png', 'testing2.jpeg']
    status_task = dict()
    task_pool = Pool(processes=20)
    catat_awal = datetime.datetime.now()
    for k in range(len(daftar)):
        print(f"mengirim {daftar[k]}")
        texec[k] = task_pool.apply_async(func=kirim_multi_process_async, args=(daftar[k],))
    for k in range(len(daftar)):
        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)
```

- Multi Thread Asynchronous

```
def kirim_multi_thread_async(daftar=None):
    if (daftar is None):
        return False
    f = open(daftar, "rb")
    l = f.read(1024)
    while (l):
        if (sock.sendto(l, (TARGET_IP, TARGET_PORT))):
            l = f.read(1024)
    f.close()

def multi_thread_async():
    texec = dict()
    daftar = ['testing1.png', 'testing2.jpeg']
    status_task = dict()
    task = concurrent.futures.ThreadPoolExecutor(max_workers=4)

    catat_awal = datetime.datetime.now()
    for k in range(len(daftar)):
        print(f"mendownload {daftar[k]}")
        waktu = time.time()
        texec[k] = task.submit(kirim_multi_thread_async, daftar[k])
    for k in range(len(daftar)):
        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)
```

- Pada Server

```

GNS3 console      alpine-1  X      alpine-2  X      -      X
d3\x0G\x0e\x06\x1dh\t^\x8b\xd7Uj9\x03\x1e:Vz\x8h\xd4j\x05\x9d\xbdE\x03\x08\x01\x08\xbb\x
a5\x05\x03\x02\x01\x05\x08\x0ad'\x0aI\t\x00\x03\x09\x0e\t\x0a\x03\x01\x08\xbb\x
b4\x08\x0e\x0a\x0e\x06\x0f\x07\x02\x09\x0a\x06\x03\x0e)\x05\x0e\x04\x06\x0d\x0e\t\x
10+j\x09'\x09M\x09\x0d\x07\x0a\x0a\x00'\x03\x0eR\x0a\x05\x0c\x04\x09dI\x08\x04\x0f\x0b\x0
4\n7'\x02:\x06\x0e1\x0e3kF\x07\x04P4T:\x09\x013\x01\x0a\x0bV\x07\x04T4T{\x0f\x00\x
a51\x0a~\x0e\x01\x0e5 {\x01\x0a\x019Q2\x03\x0d68f\x01\x04\x09\x0cau3\x0a<\x01:P
\x09\x0e\x04\x0ca\x089C\x08\x0b7\x01f\x0f'
dikirim oleh ('192.168.122.121', 60567)
('192.168.122.121', 60567)
diterima b'[\x0e\x01\x0faT\x09\x0e\x04\x0b\x0d\x03Y\x01f\x0f\x09b\x0e\x0118Sz\x04\x
ecY5\x0fb4\x07f\x08\x0a1\x092=\x08G^\tj\x0a\x0d\x09>m\x0a0\x0eT\x0d6H\x04'\x0f87\x0e \
\x0a\x07f\x0c2@\x09\x0nk\x0b2\x0e\x0f\x0c3\x0c0\x00Y\x0db\x0806\x086\x090>1J\x0d2{\x096F
\x0a\x0e1\x0e\x0b6\x0e#\x0e8\x0e1\x0cb1\x0b5\x0b3=}D\x0f9\x0f4\x0a4\x0ec\x0a1\x0d1\x017\x0ff\x0
0\x0d\x089\x0b\x0b4\x097\x0cd\x092F\x016\x0c7\x0ecZ\x0e\x04\x0e\x094{8\x0f4Eo\x015Y\x0e\x0b9
|\x0d8\x0e\x015o \x0f6-H\x0d8\x0e4L\x08f1S\x096\x0c5r\x0a9);\x0c9\x0b68\x0e5\x09bJ2\x0a6\x0d0C
R\x0900\x0bc\x08a6\x016\x0ec\x0f4\x0da7\x0f7\x011\x0fc"\x086U\x0d0~\x0d6\x0a7\x0f5?\x099\x0e1\x0b3
o\x0f6h\x0e\x011\x0faTI!\x05-\x0db:\x016\x0e8\x02\x02\x013\x01d Q\x05\x0ce\x083I\x0fb\x0a3
\x0dc(X\x099\x09fQ\x0168\x0f7G\x0b8Q#m\x0eez\x010\x07!\x0ee\x0a8\x03\x0da\x084=\x0a8AT \x0a\x
10\x0e5C\x0fa\x093P\x082\x09a\x084=\x09a\x084#\x0dd\x0d94\x0e0\x087\x01bB\x0c7\x0ef\x0a4+\x0e6*\x
10\x0af^\x0f0\x0e\x016\x0e1\x0efY1\x0cf\x0d5NM\x0c8\x09fR(\x0e6d\x03\x0bf\x0e8\x0e7\x00b\x081\x
fe\x0c8\x0812=e\x09d<%F=\x094s2\x05\x0f8s\x0851\x0ad\x082\b\x0dd\x0b6\x0d7\x011\x09c\tT\x01d\
xc6c*\x0f8\x0d01\x081\x0f7\x010(\x010\x0ff\x0d9'
dikirim oleh ('192.168.122.121', 60567)

```

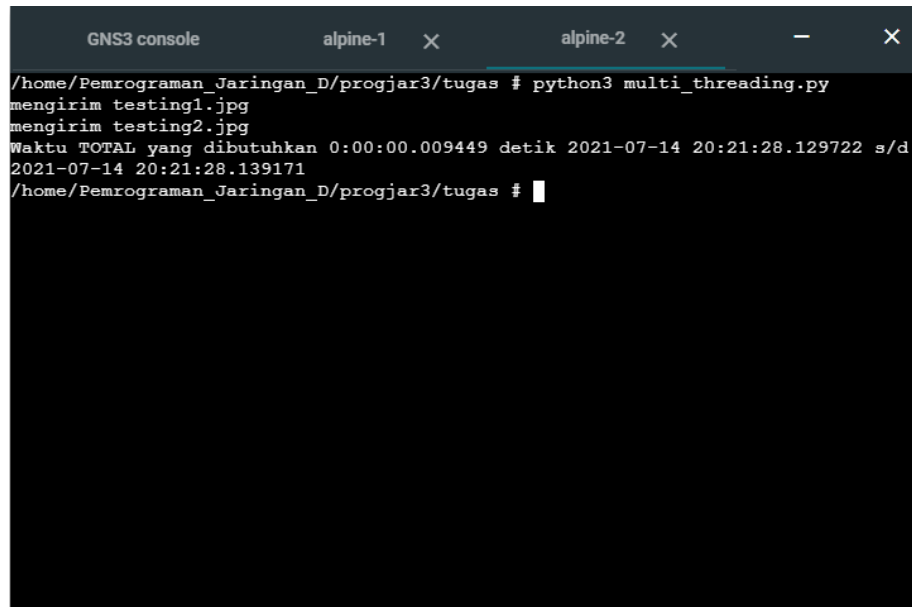
- Pada Client Multi Process

```

GNS3 console      alpine-1  X      alpine-2  X      -      X
/home/Pemrograman_Jaringan_D/progjar3/tugas # python3 multi_processing.py
mengirim testing1.jpg
mengirim testing2.jpg
Waktu TOTAL yang dibutuhkan 0:00:00.091085 detik 2021-07-14 20:19:15.198836 s/d
2021-07-14 20:19:15.289921
/home/Pemrograman_Jaringan_D/progjar3/tugas #

```

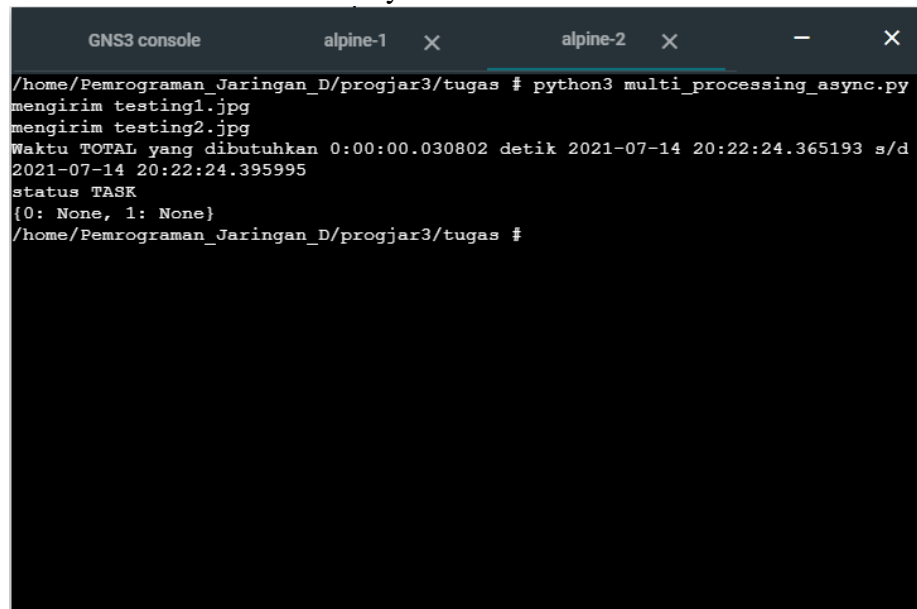
- Pada Client Multi Thread



The screenshot shows a GNS3 console window with three tabs: 'GNS3 console', 'alpine-1', and 'alpine-2'. The 'alpine-2' tab is active. The terminal output is as follows:

```
/home/Pemrograman_Jaringan_D/progjar3/tugas # python3 multi_threading.py
mengirim testing1.jpg
mengirim testing2.jpg
Waktu TOTAL yang dibutuhkan 0:00:00.009449 detik 2021-07-14 20:21:28.129722 s/d
2021-07-14 20:21:28.139171
/home/Pemrograman_Jaringan_D/progjar3/tugas #
```

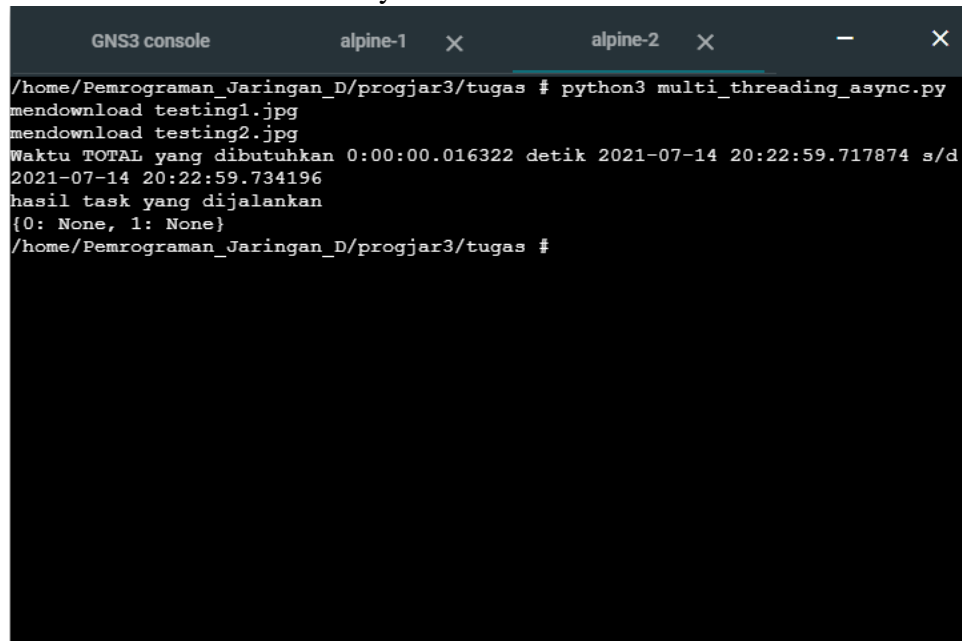
- Pada Client Multi Process Asynchronous



The screenshot shows a GNS3 console window with three tabs: 'GNS3 console', 'alpine-1', and 'alpine-2'. The 'alpine-2' tab is active. The terminal output is as follows:

```
/home/Pemrograman_Jaringan_D/progjar3/tugas # python3 multi_processing_async.py
mengirim testing1.jpg
mengirim testing2.jpg
Waktu TOTAL yang dibutuhkan 0:00:00.030802 detik 2021-07-14 20:22:24.365193 s/d
2021-07-14 20:22:24.395995
status TASK
{0: None, 1: None}
/home/Pemrograman_Jaringan_D/progjar3/tugas #
```

- Pada Client Multi Thread Asynchronous



```
GNS3 console      alpine-1  X      alpine-2  X      -      X
/home/Pemrograman_Jaringan_D/progjar3/tugas # python3 multi_threading_async.py
mendownload testing1.jpg
mendownload testing2.jpg
Waktu TOTAL yang dibutuhkan 0:00:00.016322 detik 2021-07-14 20:22:59.717874 s/d
2021-07-14 20:22:59.734196
hasil task yang dijalankan
{0: None, 1: None}
/home/Pemrograman_Jaringan_D/progjar3/tugas #
```