Tugas Akhir Mata Kuliah Pemrograman Jaringan
Pemrograman Jaringan D

Aulia Ihza Hendradi	**05111840000089**	**PROGJAR D**
Kinasihurrabb Moralluhung	**05111840000161**	**PROGJAR D**

INFORMATICS ENGINEERING DEPARTMENT
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA

2021

A. Pendahuluan

Reverse Proxy adalah server yang berada di depan server web dan meneruskan permintaan klien (misalnya *web browser*) ke server web tersebut. *Reverse Proxy* biasanya diterapkan untuk membantu meningkatkan keamanan, kinerja, dan keandalan. Sedangkan, *Load Balancing Reverse Proxy* mendistribusikan permintaan klien yang masuk di antara sekelompok server, dalam setiap kasus mengembalikan respons dari server yang dipilih ke klien yang sesuai. Pada dokumen ini akan dijelaskan penerapan studi kasus *Reverse Proxy* dan *Load Balancing Reverse Proxy* menggunakan Bahasa Pemrograman Python.

B. Anggota dan Jobdesk Kelompok

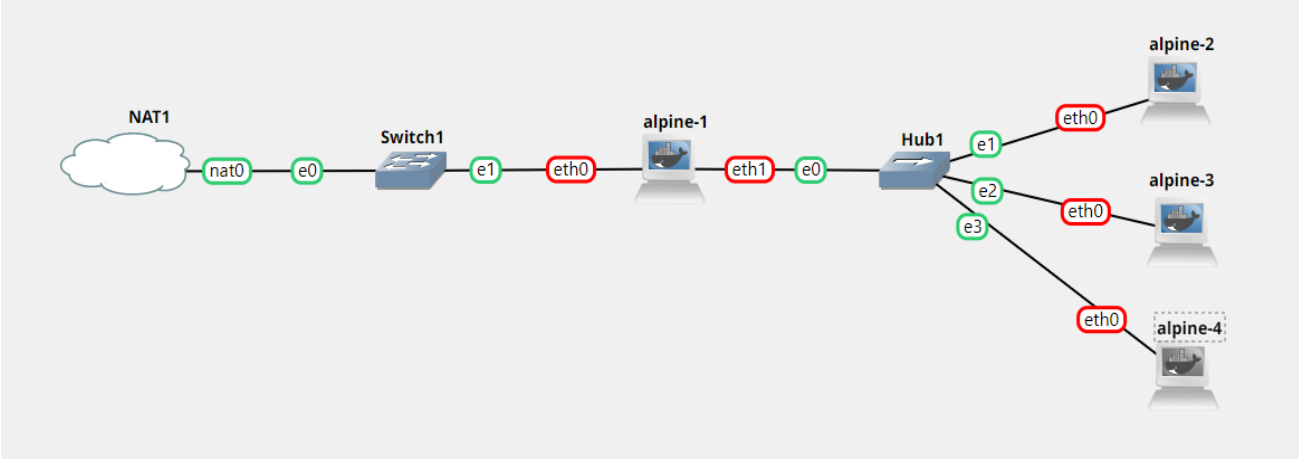
1. Aulia Ihza Hendradi – Melakukan Pengujian Apache Benchmark untuk *Load Balancing Reverse Proxy*, Memperbaiki glob dan pengecek 404 http_get pada http.py, Pemecahan HTTP untuk setiap Server Backend, Memperbaiki Respons dari *Threaded Load Balancing Reverse Proxy*, Mengerjakan Laporan.

2. Kinasihurrabb Moralluhung – Desain arsitektur untuk studi kasus *Reverse Proxy* dan *Load Balancing Reverse Proxy*, Implementasi awal untuk *Reverse Proxy*, Implementasi awal untuk *Threaded Load Balancing Reverse Proxy*, Mengerjakan laporan.

C. Arsitektur dan Konfigurasi

1. Reverse Proxy

- **Gambar Arsitektur**

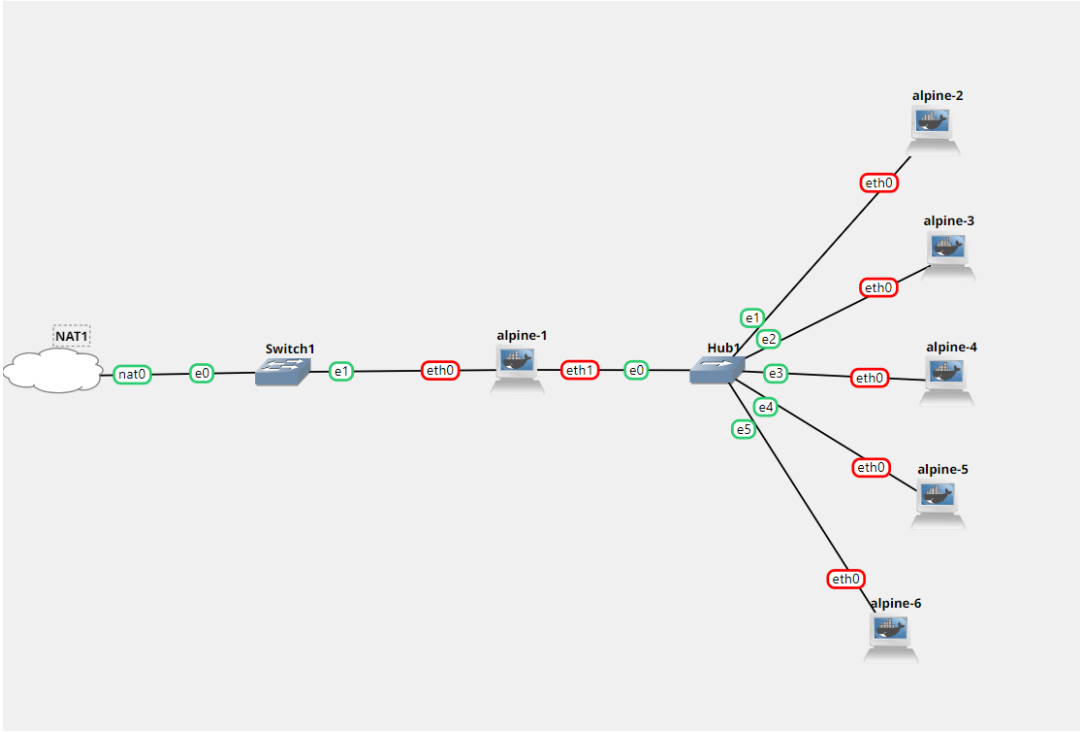


- **Konfigurasi**

Nama Node	Peran	IP	Port
Alpine - 1	Reverse Proxy Server	Localhost (127.0.0.1)	18000
Alpine - 2	Default Server	Localhost (127.0.0.1)	8888
Alpine - 3	Image Web Server	Localhost (127.0.0.1)	8889
Alpine - 4	PDF Web Server	Localhost (127.0.0.1)	8890

2. Load Balancing Reverse Proxy

- **Gambar Arsitektur**



- **Konfigurasi**

Nama Node	Peran	IP	Port
Alpine – 1a	Reverse Proxy Server - Threaded	Localhost (127.0.0.1)	666
Alpine – 1b	Reverse Proxy Server - Asynchronous	Localhost (127.0.0.1)	44444
Alpine - 2	Web Server	Localhost (127.0.0.1)	9001
Alpine - 3	Web Server	Localhost (127.0.0.1)	9002
Alpine - 4	Web Server	Localhost (127.0.0.1)	9003
Alpine - 5	Web Server	Localhost (127.0.0.1)	9004
Alpine - 6	Web Server	Localhost (127.0.0.1)	9005

D. Pengujian

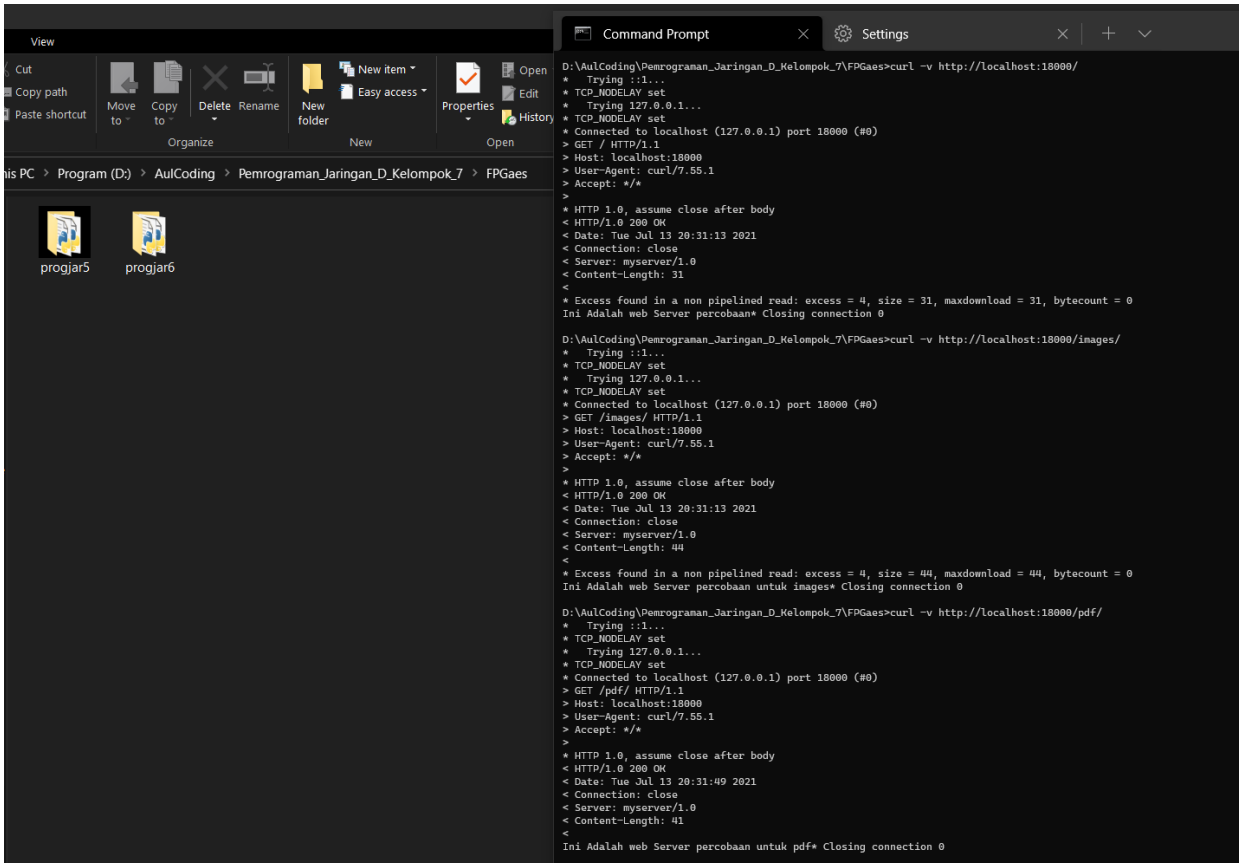
1. **Reverse Proxy**
   Pengujian pada *reverse proxy* akan menggunakan **dua** *HTTP Client*, yakni **curl** dan **web browser** berupa **Google Chrome**. Untuk metode **curl** digunakan parameter **-v** (**verbose)** untuk menampilkan detil **request** dan **–output** untuk menyimpan objek khusus untuk *Image* dan *PDF Web Server*.

2. **Load Balancing Reverse Proxy**
   Pengujian untuk *Load Balancing Reverse Proxy* akan menggunakan **Apache Benchmark** untuk menyajikan *request* dalam tingkatan *concurrency* yang berbeda – beda untuk menguji kemampuan arsitektur dalam menangani *traffic* yang diberikan.

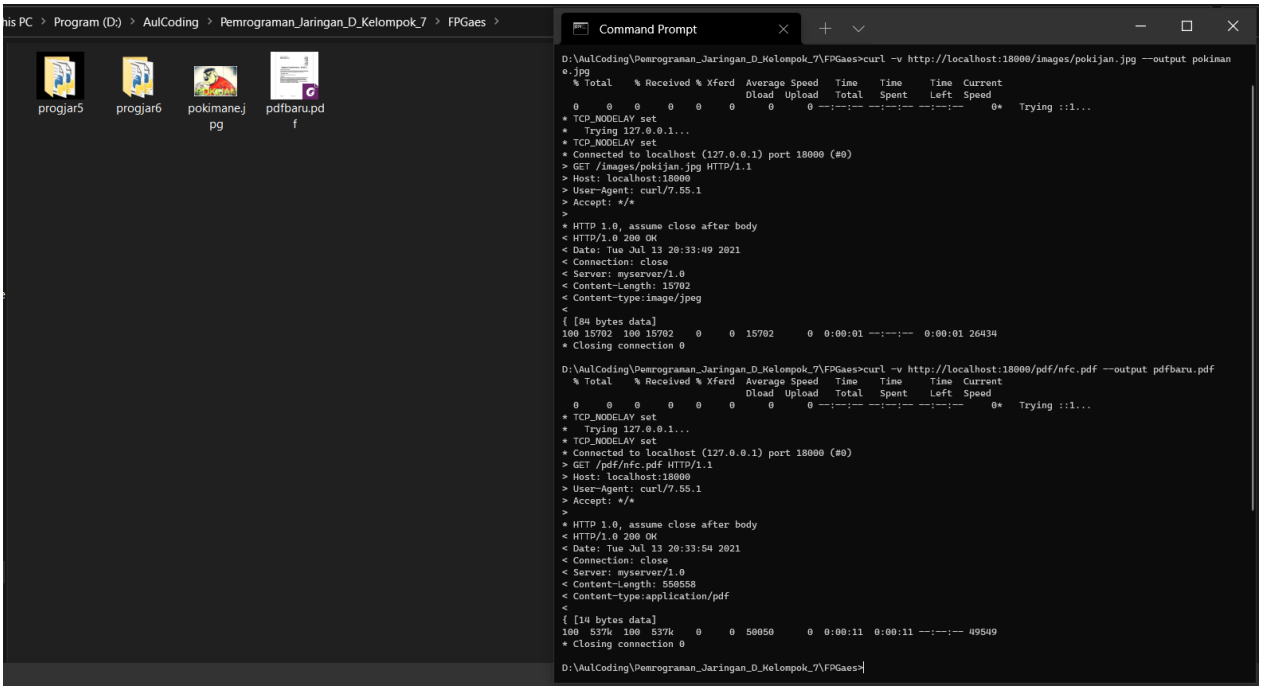
E. Screenshot Hasil

1. Reverse Proxy

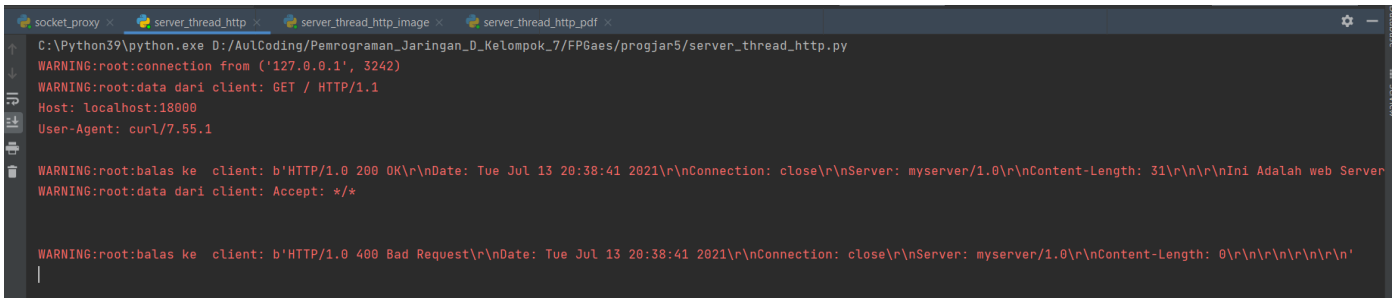
- **Pengujian curl -v untuk ketiga laman utama server**



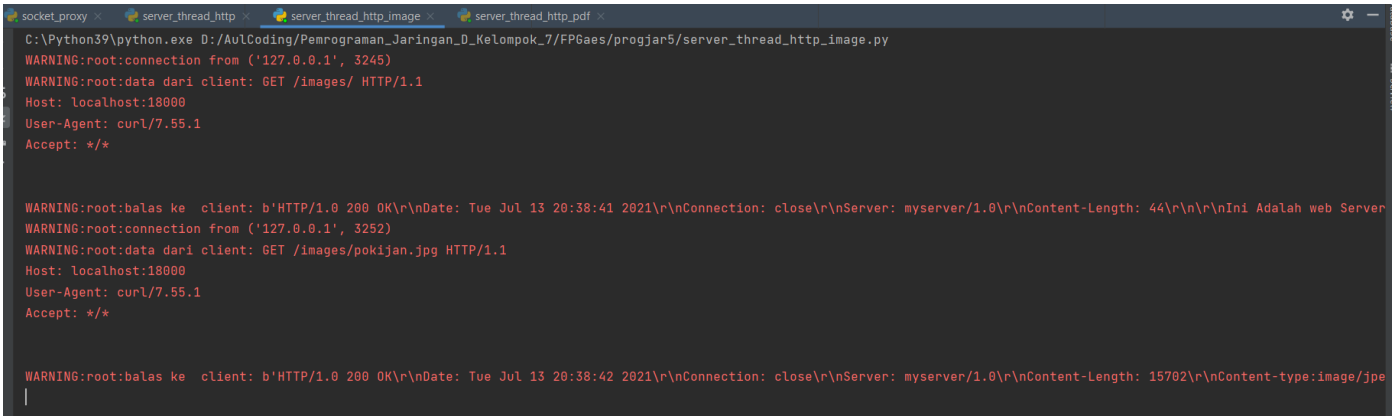
- **Pengujian curl -v dengan –output untuk pokijan.jpg pada Image Web Server dan nfc.pdf pada PDF Web Server**



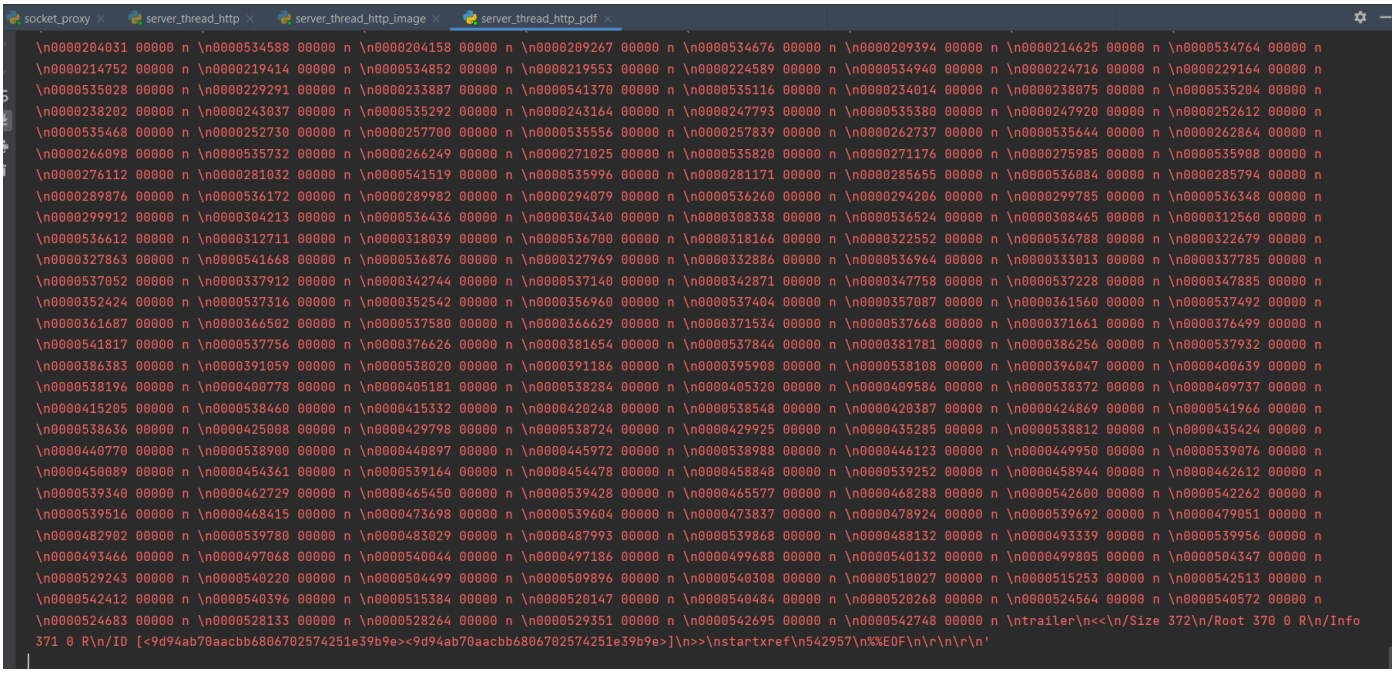
- **Log di Reverse Proxy**



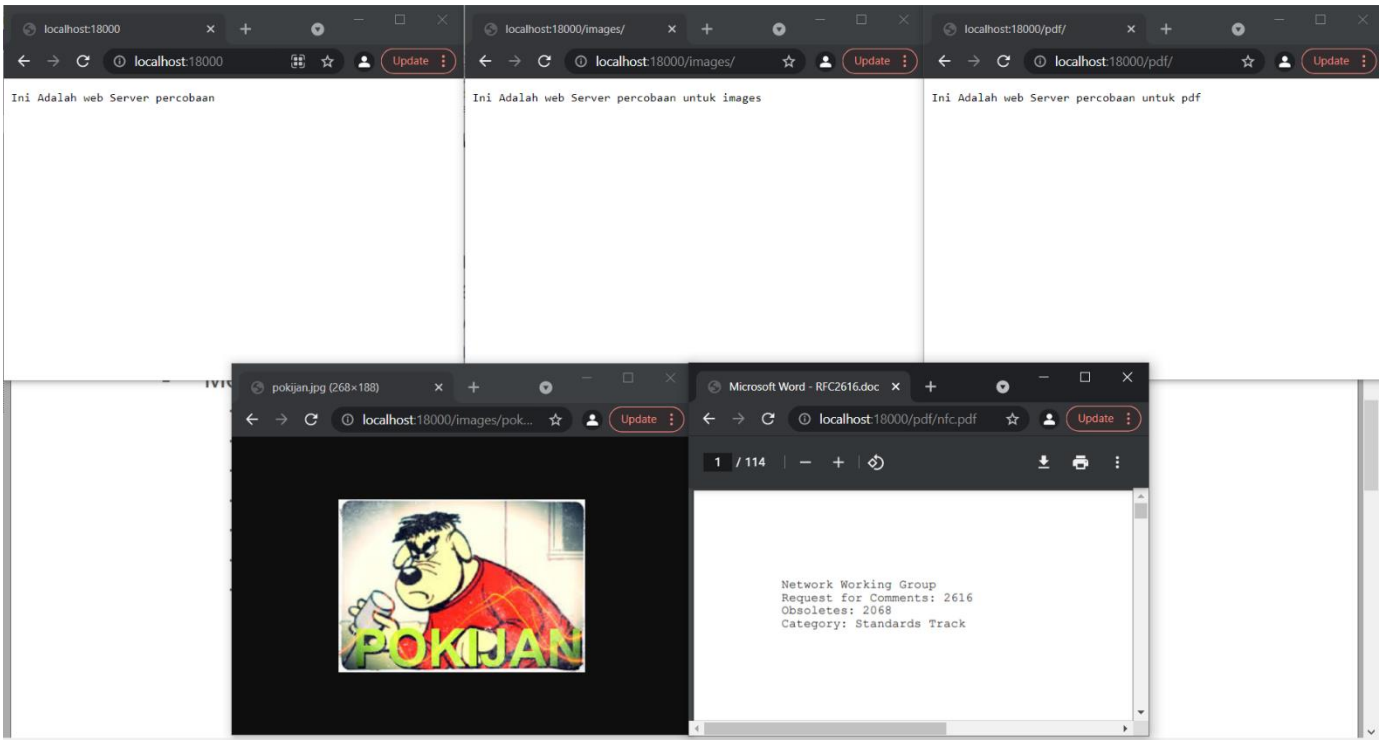
- **Log di Image Web Server**



- **Log di PDF Web Server**

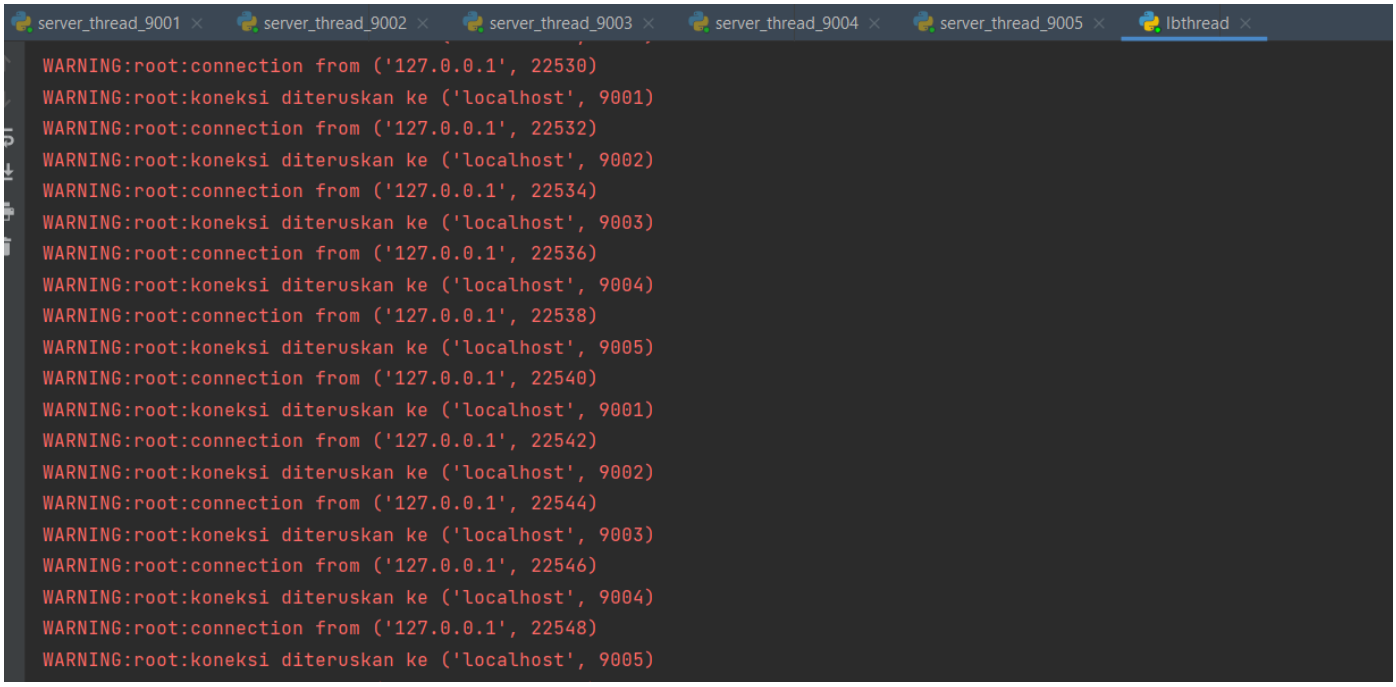


- **Pengujian di Browser Google Chrome**

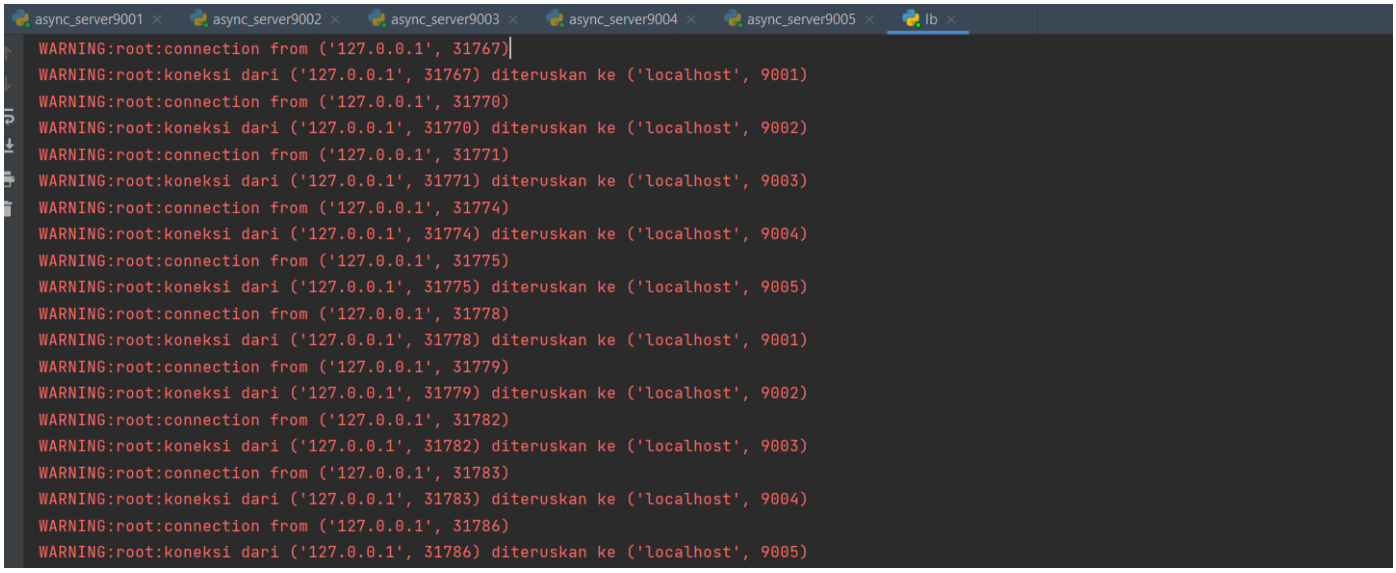


2. Load Balancing Reverse Proxy

- **Log di Threaded Reverse Proxy Load Balancer**



- **Log di Asynchronous Reverse Proxy Load Balancer**



F. Tabel Hasil Apache Benchmark untuk Kasus 2

1. Threaded Reverse Proxy Load Balancer

Jumlah HTTP Server	Concurrency	Jumlah Complete Request	Non-2xx response	Jumlah request per second	Time Per request (mean across)
1	2	10000	0	378.12 #/sec	2.645 ms
	5	10000	0	4.64 #/sec	215.722 ms
	10	10000	0	4.89 #/sec	204.429 ms
2	2	10000	0	443.85 #/sec	2.253 ms
	5	10000	0	157.43 #/sec	6.352 ms
	10	10000	0	11.14 #/sec	89.779 ms
3	2	10000	0	456.06 #/sec	2.193 ms
	5	10000	0	759.17 #/sec	1.317 ms
	10	10000	0	57.57 #/sec	17.371 ms
4	2	10000	0	435.01 #/sec	2.299 ms
	5	10000	0	753.43 #/sec	1.327 ms
	10	10000	0	120.10 #/sec	8.326 ms
5	2	10000	0	353.06 #/sec	2.832 ms
	5	10000	0	748.18 #/sec	1.337 ms
	10	10000	0	155.60 #/sec	6.427 ms

2. Asynchronous Reverse Proxy Load Balancer

Jumlah HTTP Server	Concurrency	Jumlah Complete Request	Non-2xx response	Jumlah request per second	Time Per request (mean across)
1	2	10000	0	525.19 #/ssec	1.904 ms
	5	10000	0	3.92 #/sec	254.829 ms
	10	10000	0	3.92 #/sec	254.961 ms
2	2	10000	0	539.83 #/sec	1.852 ms
	5	10000	0	10.36 #/sec	96.524 ms
	10	10000	0	5.91 #/sec	169.075 ms
3	2	10000	0	527.32 #/sec	1.896 ms
	5	10000	0	656.13 #/sec	1.524 ms
	10	10000	0	9.05 #/sec	110.452 ms
4	2	10000	0	496.25 #/sec	2.015 ms
	5	10000	0	676.01 #/sec	1.479 ms
	10	10000	0	11.73 #/sec	85.277 ms
5	2	10000	0	420.57 #/sec	2.378 ms
	5	10000	0	445.18 #/sec	2.246 ms

	10	10000	0	14.38 #/sec	69.539 ms

G. Kesimpulan dan Penutup

Untuk *Reverse Proxy*, dapat dilihat ketika *Reverse Proxy Server* menerima *request* dalam *path URL* tertentu (PDF dan Image) maka *request* akan diteruskan ke *backend server* sesuai dengan request yang dispesifikasikan. Hal ini tentunya bermanfaat pada saat kita membuat sebuah *website* yang melakukan *hosting* suatu konten. Agar semua *traffic* tidak menuju pada *server* utama, kita dapat mengimplementasikan *Reverse Proxy* agar kita dapat membagi *traffic* yang datang pada masing – masing *webserver* yang sesuai. Pada *Reverse Proxy Server*, proses *traffic forwarding* dapat dibuktikan dengan log yang tertulis pada *Reverse Proxy Server*. Sedangkan, dari sudut pandang *client* yang dapat dilihat hanyalah *request* yang dikirimkan ke *Reverse Proxy Server* serta respons yang dikirimkan dari *backend server* yang dihantarkan melalui *Reverse Proxy Server*.

Untuk *Load Balancing Reverse Proxy*, dapat dilihat dari perbandingan hasil dua *apache benchmark* diatas, *Threaded Reverse Proxy Load Balancer* memiliki performa yang lebih baik jika dibandingkan dengan *Asynchronous Reverse Proxy Load Balancer* dalam aspek jumlah *request* yang berhasil dijalankan setiap detik untuk level konkurensi 5 dan 10. Akan tetapi, *Asynchronous Reverse Proxy Load Balancer* juga memiliki kelebihan pada uji konkurensi tingkat 2, ia unggul pada pengujian tersebut. Sehingga, dapat disimpulkan bahwa masing – masing model memiliki kelebihan dan kekurangan masing – masing untuk tingkatan konkurensi dalam tes. Perlu diingat, bahwa level konkurensi menentukan seberapa banyak request yang dapat diproses dalam waktu yang bersamaan.

Dari dokumen tugas akhir mata kuliah pemrograman jaringan yang telah ditulis, diharapkan pembaca dapat memperoleh informasi terkait dengan arsitektur, konfigurasi, pengujian, serta hasil dan perbandingan mengenai arsitektur *Reverse Proxy* dan *Load Balancing Reverse Proxy* yang telah dibuat dan diuji oleh penulis.