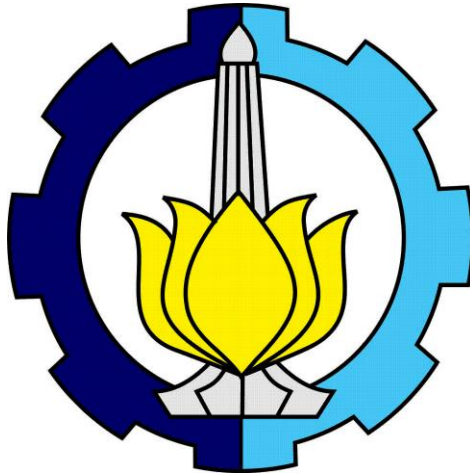


Dokumentasi
Studi Kasus Reverse Proxy dan
Load Balancing Reverse Proxy



Pemrograman Jaringan
Kelas E

Kelompok E9:

Khofifah Nurlaela	05111840000025
Muhammad Afif Fadhlurrahman	05111840000093
Muhammad Rivaldhi Purnomo	05111840000128
Brananda Denta W.P.	05111840000143

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

S1 TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2021

1. Pendahuluan

Pada tugas kelompok mata kuliah Pemrograman Jaringan kali ini terdapat dua studi kasus yang harus diimplementasikan, yaitu *reserve proxy* dan *load balancing reserve*.

Studi kasus yang pertama adalah *reserve proxy*. Pada kasus ini terdapat *client*, *proxy*, dan sebuah *backend server*. *Reverse proxy* berfungsi untuk melakukan *port forwarding* suatu *request*, dari *public request* menuju ke dalam sistem. Pada kasus ini, *reverse proxy* akan menerjemahkan *path* sebuah URL untuk diteruskan ke *backend server* yang sesuai. Jika *reverse proxy* menerima permintaan dalam *path /images*, maka objek akan di *retrieve* dari *backend server* yang melayani */images*. Begitu pula jika *reverse proxy* menerima permintaan dalam *path /pdf*, maka objek akan di *retrieve* dari *backend server* yang melayani */pdf*. Untuk *HTTP client* yang digunakan dalam uji coba studi kasus pertama ini berupa web browser dan curl command.

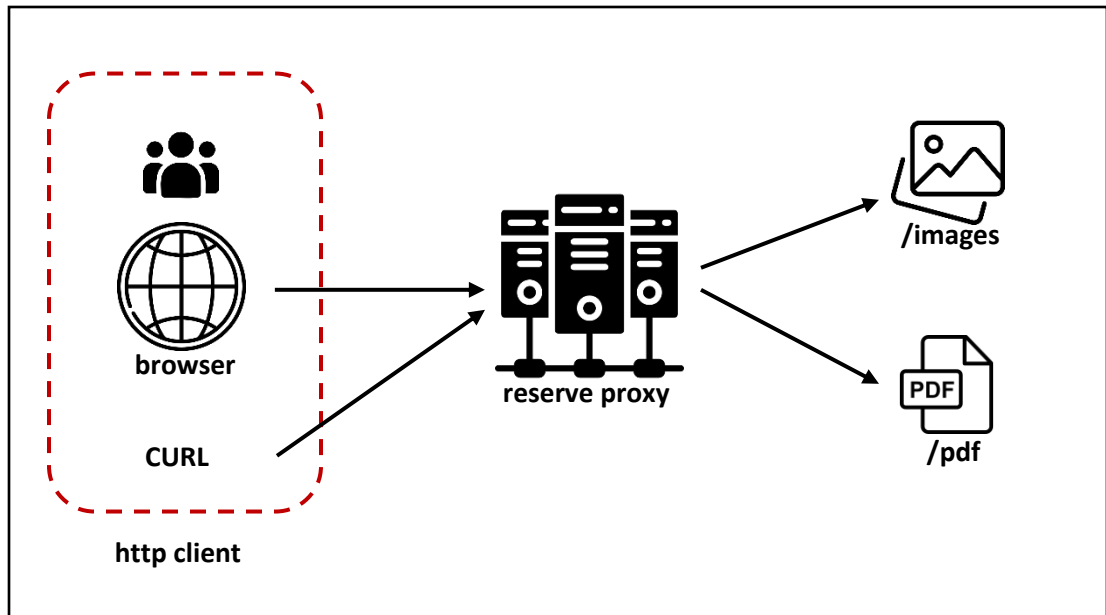
Studi kasus kedua adalah *load balancing reverse proxy*. Penggunaan *load balancing* pada web server bertujuan untuk meringankan beban yang ditanggung masing-masing server. *Load balancing* sebuah teknik mendistribusikan beban *traffic* pada dua jalur atau lebih, sehingga didapatkan sambungan yang seimbang, *traffic* yang lebih optimal, *throughput* data maksimal, *delay* minimal, serta tidak terjadi *overload*. Pada studi kasus ini terdapat dua model *load balancing reverse proxy* yang akan diimplementasikan, yaitu model *threaded* dan model *asynchronous*. Permintaan yang diterima oleh *reverse proxy*, akan diteruskan ke *cluster backend* dengan menggunakan algoritma *Round Robin*. Algoritma *Round Robin* merupakan algoritma yang paling sederhana dan paling banyak digunakan oleh perangkat *load balancing*. Algoritma ini bekerja dengan cara membagi beban secara bergiliran dan berurutan dari satu server ke server lainnya. Setiap model akan diuji testing performance dengan menggunakan *Apache Benchmark* dengan target server *reverse proxy*. Uji testing performance akan dilakukan masing-masing 15 kali percobaan pada *threaded* dan *asynchronous* dengan menggunakan 10000 *request*. Jumlah backend server untuk setiap uji percobaan dimulai dari 1 server sampai 5 server serta akan menggunakan *concurrency level* sebesar: 2, 5, dan 10.

2. Tugas Anggota Kelompok

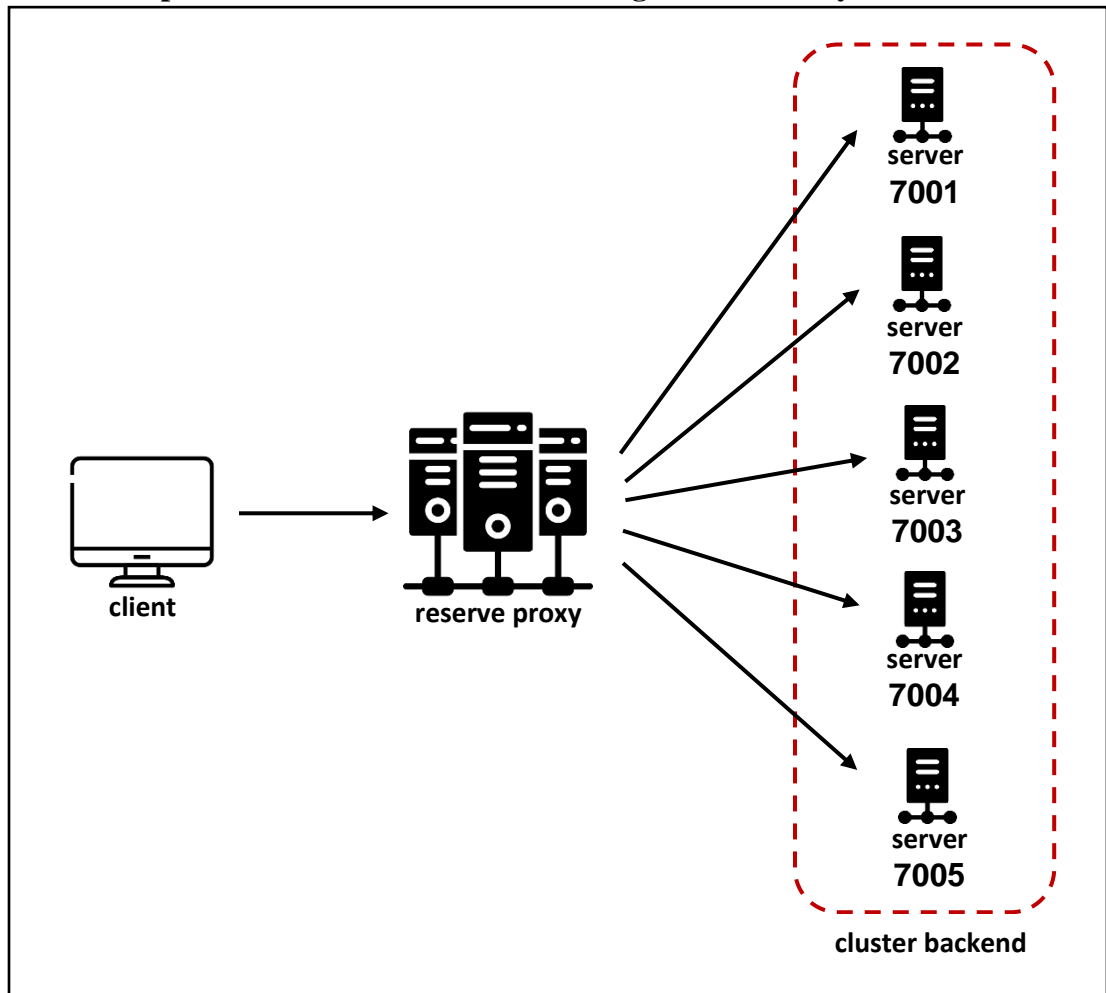
- Muhammad Afif Fadhlurrahman : Load balancing threaded
- Muhammad Rivaldhi Purnomo : Reserve Proxy
- Brananda Denta W.P. : Load balancing asynchronous
- Khofifah Nurlaela : Reserve Proxy

3. Arsitektur dan Konfigurasi

1) Arsitektur pada studi kasus 1: Reserve Proxy



2) Arsitektur pada studi kasus 2: Load Balancing Reserve Proxy



4. Pengujian

1) Studi Kasus 1: Reverse Proxy

Berikut merupakan cara melakukan pengujian reverse proxy pada studi kasus 1:

- a) Menjalankan perintah `python .\server_thread_http_images.py` sebagai backed server untuk dapat /images.
- b) Menjalankan perintah `python .\server_thread_http_pdf.py` sebagai backed server untuk dapat /pdf.
- c) Untuk menjalankan reverse proxy, maka menjalankan `python3 .\socket_poxy.py`. Sehingga reverse proxy dapat diakses pada `http://localhost:5000`.
- d) Pada browser, mengakses <http://localhost:5000/images> untuk mendapatkan file jpg, sedangkan <http://localhost:5000/pdf> untuk file pdf.

2) Studi Kasus 2: Load Balancing Reverse Proxy

Berikut merupakan cara melakukan pengujian load balancing reverse proxy pada studi kasus 2:

- a) Pada model Threaded:
 - (i) Membuka beberapa terminal sesuai dengan kebutuhan *cluster backend server*. Jika dibutuhkan 1 *backend server*, maka hanya membuka 1 terminal. Jika dibutuhkan 2 backend server, maka menggunakan 2 terminal, dan seterusnya.
 - (ii) Pada setiap terminal menjalankan `python3 server_thread_http.py <port>`. Port menyesuaikan jumlah backend server yang dimulai dari port 7001. Maka jika jumlah backend server ada 5, port yang akan dibuka antara lain: port 7001, port 7002, port 7003, port 7004, dan port 7005.
 - (iii) Untuk reverse proxy, membuat satu terminal dan menjalankan `python3 socket_proxy_thread.py`. Maka reverse proxy dapat diakses pada `http://localhost:18000/`.
 - (iv) Untuk penerapan load balancing secara *Round Robin*, maka menggunakan *library* `lb.py` dengan membuat array yang berisi alamat masing-masing backend server berdasarkan poin (ii).
 - (v) Untuk menjalankan performance test dengan *Apache Benchmark*, maka membuat satu terminal baru, kemudian menjalankan `ab -n 10000 -c <jumlah concurrency> http://localhost:18000/`. Kemudian menunggu hasil setelah proses *Apache Benchmark* selesai.
- b) Pada model Asynchronous:
 - (i) Membuka beberapa terminal sesuai dengan kebutuhan *cluster backend server*. Jika dibutuhkan 1 *backend server*, maka hanya membuka 1 terminal. Jika dibutuhkan 2 backend server, maka menggunakan 2 terminal, dan seterusnya.

- (ii) Pada setiap terminal menjalankan `python3 async_server.py <port>`. Port menyesuaikan jumlah backend server yang dimulai dari port 7001. Maka jika jumlah backend server ada 5, port yang akan dibuka antara lain: port 7001, port 7002, port 7003, port 7004, dan port 7005.
- (iii) Untuk reverse proxy, membuat satu terminal dan menjalankan `python3 socket_proxy_async.py`. Maka reverse proxy dapat diakses pada `http://localhost:8890/`.
- (iv) Untuk penerapan load balancing secara *Round Robin*, maka menggunakan *library* `lb.py` dengan membuat array yang berisi alamat masing-masing backend server berdasarkan poin (ii).
- (v) Untuk menjalankan performance test dengan *Apache Benchmark*, maka membuat satu terminal baru, kemudian menjalankan `ab -n 10000 -c <jumlah concurrency> http://localhost:8890/`. Kemudian menunggu hasil setelah proses *Apache Benchmark* selesai.

5. Screenshoot Hasil

Studi kasus 1 Reverse proxy:



Studi kasus 2 Load balancing reverse proxy:

Asynchronous:

1 server, concurrency 2, 5, 10:

```
C:\xampp\apache\bin>ab -n 10000 -c 2 http://localhost:8080/
This is ApacheBench, Version 2.3 <Revision: 1879490> $
Copyright 1996 Adam Teiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:      mysqserver/1.0
Server Hostname:      localhost
Server Port:          68001

Document Path:        /
Document Length:       19 bytes

Concurrency Level:    2
Time taken for tests:  312.583 seconds
Complete requests:    10000
Failed requests:       0
  (Connect: 0, Receive: 0, Length: 1, Exceptions: 0)
Total transferred:    190000 bytes
HTML transferred:     179000 bytes
Requests per second:  31.99 [#/sec] (mean)
Time per request:     62.517 [ms] (mean)
Time per request:     31.258 [ms] (mean, across all concurrent requests)
Transfer rate:        4.09 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  0  0  0.8  1  30
Processing: 8 61 202.0  34  7888
Waiting:  0 59 201.7  32  7887
Total:    9 62 202.1  35  7889

Percentage of the requests served within a certain time (ms)
 50%  35
 66%  46
 75%  56
 80%  64
 90%  97
 95%  145
 98%  270
 99%  445
100% 7889 (longest request)
```

```
C:\xampp\apache\bin>ab -n 10000 -c 5 http://localhost:68001/
This is ApacheBench, Version 2.3 <Revision: 1879490> $
Copyright 1996 Adam Teiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:      mysqserver/1.0
Server Hostname:      localhost
Server Port:          68001

Document Path:        /
Document Length:       19 bytes

Concurrency Level:    5
Time taken for tests:  162.135 seconds
Complete requests:    10000
Failed requests:       0
  (Connect: 0, Receive: 0, Length: 872, Exceptions: 0)
Total transferred:    173570 bytes
HTML transferred:     173432 bytes
Requests per second:  61.667 [#/sec] (mean)
Time per request:     21.067 [ms] (mean)
Time per request:     10.213 [ms] (mean, across all concurrent requests)
Transfer rate:        11.43 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  0  0  0.7  1  6
Processing: 1 50 125.1  45  5660
Waiting:  0 46 124.9  42  5456
Total:    1 51 125.1  46  5660

Percentage of the requests served within a certain time (ms)
 50%  45
 66%  51
 75%  55
 80%  58
 90%  67
 95%  89
 98%  143
 99%  144
100% 5660 (longest request)
```

```
C:\xampp\apache\bin>ab -n 10000 -c 10 http://localhost:68001/
This is ApacheBench, Version 2.3 <Revision: 1879490> $
Copyright 1996 Adam Teiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:      mysqserver/1.0
Server Hostname:      localhost
Server Port:          68001

Document Path:        /
Document Length:       19 bytes

Concurrency Level:    10
Time taken for tests:  842.093 seconds
Complete requests:    10000
Failed requests:       0
  (Connect: 0, Receive: 0, Length: 529, Exceptions: 0)
Total transferred:    1246703 bytes
HTML transferred:     179000 bytes
Requests per second:  11.88 [#/sec] (mean)
Time per request:     842.093 [ms] (mean)
Time per request:     84.209 [ms] (mean, across all concurrent requests)
Transfer rate:        1.44 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  0  0 109.1  0  519
Processing: 10 758 285.4  517 1037
Waiting:  0 485 115.3  512  520
Total:    12 842 245.0 1021 1535

Percentage of the requests served within a certain time (ms)
 50% 1021
 66% 1023
 75% 1025
 80% 1025
 90% 1027
 95% 1028
 98% 1030
 99% 1030
100% 1535 (longest request)
```

Untuk hasil screenshoot yang lain dapat dilihat pada :

<https://drive.google.com/drive/folders/1phBIGrUubluXdKqhsyAkkQqUTyVUppcj?usp=sharing>

Threaded:

1 server, concurrency 2, 5, 10:

```
C:\xampp\apache\bin>ab -n 10000 -c 2 http://localhost:18000/
This is ApacheBench, Version 2.3 <Revision: 1879490> $
Copyright 1996 Adam Teiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:      mysqserver/1.0
Server Hostname:      localhost
Server Port:          18000

Document Path:        /
Document Length:       19 bytes

Concurrency Level:    2
Time taken for tests:  45.477 seconds
Complete requests:    10000
Failed requests:       0
Total transferred:    1310000 bytes
HTML transferred:     100000 bytes
Requests per second:  212.89 [#/sec] (mean)
Time per request:     9.695 [ms] (mean)
Time per request:     4.548 [ms] (mean, across all concurrent requests)
Transfer rate:        28.13 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  0  0  0.5  0  2
Processing: 3  9 36.4  6 1183
Waiting:  3  7 12.8  5  691
Total:    4  9 36.4  6 1183

Percentage of the requests served within a certain time (ms)
 50%  6
 66%  7
 75%  8
 80%  8
 90%  10
 95%  11
 98%  14
 99%  17
100% 1183 (longest request)
```

```
C:\xampp\apache\bin>ab -n 10000 -c 5 http://localhost:18000/
This is ApacheBench, Version 2.3 <Revision: 1879490> $
Copyright 1996 Adam Teiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:      mysqserver/1.0
Server Hostname:      localhost
Server Port:          18000

Document Path:        /
Document Length:       19 bytes

Concurrency Level:    5
Time taken for tests:  38.553 seconds
Complete requests:    10000
Failed requests:       0
Total transferred:    1310000 bytes
HTML transferred:     100000 bytes
Requests per second:  259.38 [#/sec] (mean)
Time per request:     19.277 [ms] (mean)
Time per request:     3.855 [ms] (mean, across all concurrent requests)
Transfer rate:        33.18 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  0  0  0.5  0  3
Processing: 4 19 93.0  8 1635
Waiting:  3 14 68.9  7 1501
Total:    4 19 93.0  8 1635

Percentage of the requests served within a certain time (ms)
 50%  8
 66%  9
 75%  10
 80%  10
 90%  12
 95%  14
 98%  25
 99%  510
100% 1635 (longest request)
```

```
C:\xampp\apache\bin>ab -n 10000 -c 10 http://localhost:18000/
This is ApacheBench, Version 2.3 <Revision: 1879490> $
Copyright 1996 Adam Teiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:      mysqserver/1.0
Server Hostname:      localhost
Server Port:          18000

Document Path:        /
Document Length:       19 bytes

Concurrency Level:    10
Time taken for tests:  60.791 seconds
Complete requests:    10000
Failed requests:       0
Total transferred:    1310000 bytes
HTML transferred:     100000 bytes
Requests per second:  164.50 [#/sec] (mean)
Time per request:     60.791 [ms] (mean)
Time per request:     6.079 [ms] (mean, across all concurrent requests)
Transfer rate:        21.84 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  0  2 26.4  0  516
Processing: 4  50 250.4  9 2069
Waiting:  3 36 177.2  7 1075
Total:    4 60 253.3  9 2069

Percentage of the requests served within a certain time (ms)
 50%  9
 66% 10
 75% 11
 80% 12
 85% 15
 95% 512
 98% 1023
 99% 1537
100% 2069 (longest request)
```

Untuk hasil screenshoot yang lain dapat dilihat pada :

https://drive.google.com/drive/folders/1zdR6WvNXaK_qGkjdJ_45Nj37qpbLiFF1?usp=sharing

6. Tabel Hasil Apache Benchmark untuk Kasus 2

Tabel Apache Benchmark Asynchronous:

No	Jumlah http server	Conccurency	Jumlah complete request	Non-2xx response	Jumlah request per second	Time per request (mean across)
1	1 backend server	2	10000	1	31.99	62.517
2	1 backend server	5	10000	872	97.91	51.067
3	1 backend server	10	10000	529	11.88	842.093
4	2 backend server	2	10000	10	274.7	7.281
5	2 backend server	5	10000	1117	273.33	18.293
6	2 backend server	10	10000	1	11.81	846.678
7	3 backend server	2	10000	4	170.2	11.751
8	3 backend server	5	10000	128	246.73	20.265
9	3 backend server	10	10000	0	11.69	855.151
10	4 backend server	2	10000	0	184.68	10.829
11	4 backend server	5	10000	171	208.56	23.973
12	4 backend server	10	10000	0	11.71	854.027
13	5 backend server	2	10000	8	162.66	12.296
14	5 backend server	5	10000	104	179.66	27.83
15	5 backend server	10	10000	0	11.74	851.816

Tabel Apache Benchmark Threaded:

No	Jumlah Http Server	Concurrency	Jumlah complete request	Non 2x response	Jumlah request per second	Time per request (mean across)
1	1 backend server	2	10000	0	219.89	9.095
2	1 backend server	5	10000	0	259.38	19.277
3	1 backend server	10	10000	0	164.5	60.791
4	2 backend server	2	10000	0	226.81	8.818
5	2 backend server	5	10000	0	274.66	18.244
6	2 backend server	10	10000	0	13.73	728.452
7	3 backend server	2	10000	0	160.14	12.489
8	3 backend server	5	10000	0	195.67	25.553
9	3 backend server	10	10000	0	13.51	740.371

10	4 backend server	2	10000	0	165.47	18.964
11	4 backend server	5	10000	0	135.24	36.972
12	4 backend server	10	10000	0	13.51	739.975
13	5 backend server	2	10000	0	95.89	20.857
14	5 backend server	5	10000	0	105.56	47.366
15	5 backend server	10	10000	0	13.61	734.75

7. Kesimpulan

Berdasarkan hasil percobaan yang dilakukan pada studi kasus *reverse proxy*, *reverse proxy* dapat berjalan dengan sesuai dengan fungsinya, *reverse proxy* dapat menghubungkan *client* dengan *backend server*, dan dapat melakukan download file, baik berupa gambar maupun pdf.

Kemudian untuk uji testing pada studi kasus *load balancing reverse proxy* baik itu pada model *asynchronous* maupun *threaded*, jika jumlah *backend server* yang digunakan semakin banyak, maka waktu yang dibutuhkan akan relatif semakin lama. Kemudian pada semua uji coba, jika *concurrency level* yang digunakan semakin besar, maka waktu yang dibutuhkan untuk menyelesaikan pemrosesan request akan semakin cepat. Berdasarkan hasil percobaan dapat disimpulkan bahwa pada model *asynchronous* waktu yang diperlukan untuk dapat menyelesaikan requests lebih cepat dibandingkan pada model *threaded*. Hal ini dikarenakan pada model *asynchronous*, eksekusi proses dijalankan ketika instruksi diberikan, sedangkan pada proses *threaded* yang bukan *asynchronous thread* akan berjalan normal sesuai *flow control*.