

BAB III

METODE PENELITIAN

Pada penelitian ini penulis meneliti mengenai Analisis Performansi *Reverse Proxy Nginx* dalam menangani *Client Request* menggunakan Ubuntu 20.4 dengan metode Eksperimen. Di mana metode ini bertujuan untuk menguji pengaruh suatu *variable* terhadap *variable* lain, atau menguji hubungan sebab akibat antara *variable* yang satu dengan yang lainnya.

3.1 PERANGKAT YANG DIGUNAKAN

Pada penelitian ini penulis menggunakan beberapa alat yaitu perangkat keras dan perangkat lunak sebagai berikut

3.1.1 PERANGKAT KERAS (*HARDWARE*)

Perangkat keras yang akan digunakan pada penelitian ini yaitu menggunakan 1 laptop sebagai client sekaligus sebagai perangkat yang digunakan penulis untuk melakukan penelitian ini, dengan spesifikasi seperti yang terdapat pada tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras

OS	Windows 10 Home Single 20H2
<i>Processor</i>	Intel@ Celeron@ N4120 CPU 1.10GHz
RAM	4 GB
SSD	256 GB

3.1.2 PERANGKAT KERAS LUNAK (*SOFTWARE*)

Pada penelitian ini terdapat 2 perangkat virtual yang dibangun pada VirtualBox yaitu 2 *web server* dengan *reverse proxy nginx* dan tanpa *reverse proxy nginx*. Spesifikasi perangkat virtual tercantum pada Tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Virtual

Virtual Machine	Spesifikasi	
<i>WEB SERVER 1</i>	OS	Ubuntu Server 20.04
	RAM	1 GB
	Harddisk	15 GB
	Alamat IP	192.168.56.108
	Nama VM	Ubuntu Server Apache Nginx
<i>WEB SERVER 2</i>	OS	Ubuntu Server 20.04
	RAM	1 GB
	Harddisk	15 GB
	Alamat IP	192.168.56.109
	Nama VM	Ubuntu Server Apache Varnish
<i>WEB SERVER 3</i>	OS	Ubuntu Server 20.04
	RAM	1 GB
	Harddisk	15 GB
	Alamat IP	192.168.56.105
	Nama VM	Ubuntu Client

Perangkat lunak sebagai *Tool* dan aplikasi yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.3.

Tabel 3.3 Tool dan Aplikasi

No	Software	Versi	Fungsi
1	VirtualBox	6.1	Virtual Enviroment
2	Apache	2.4	Web server
3	Apache Benchmark	2.3	Tool Pengujian Web server
4	Browser Firefox	96.0.3	Menampilkan Halaman Web server
5	Reverse Proxy Caching Nginx	1.18	Web Caching

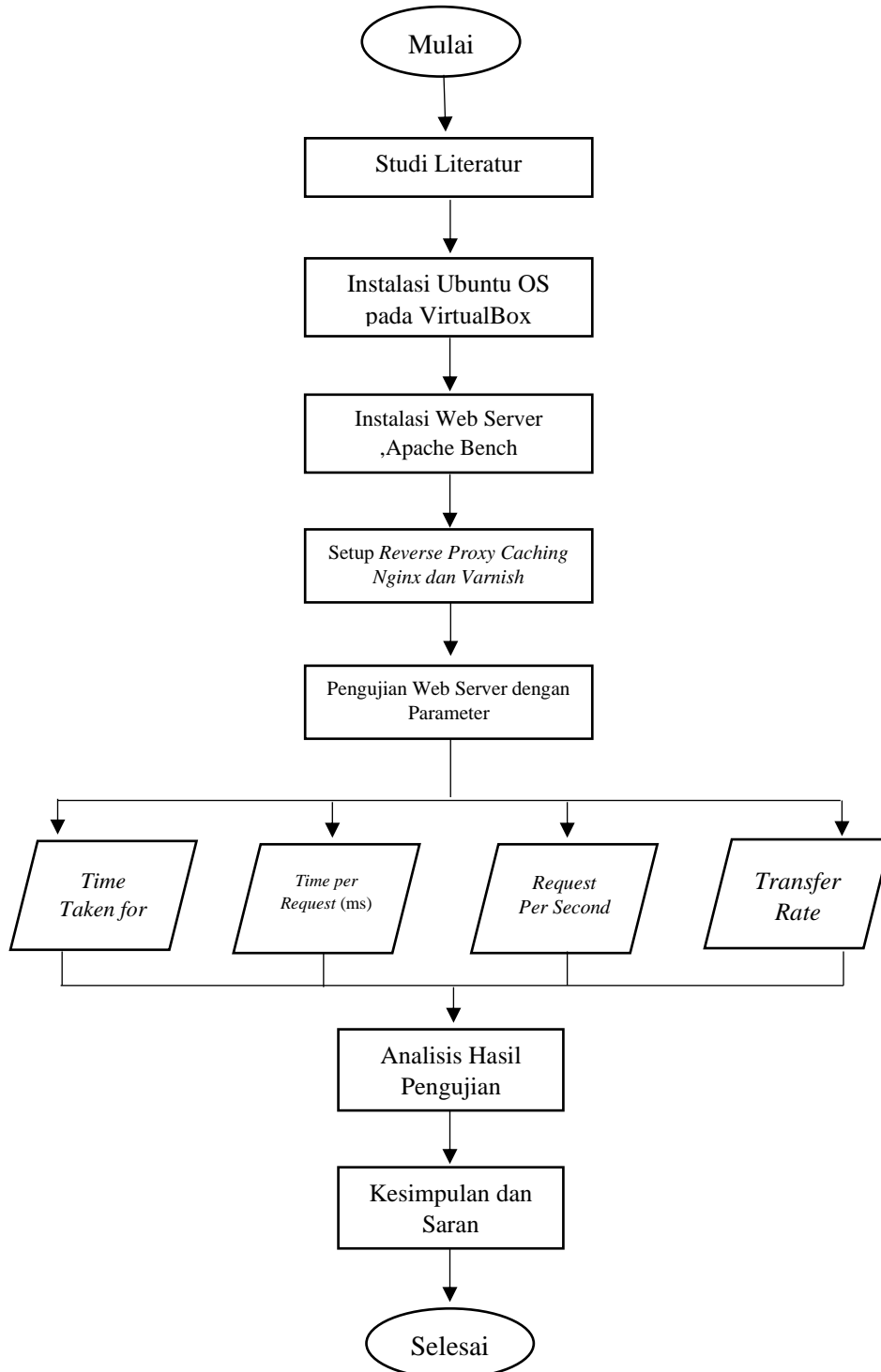
6	Reverse Proxy Caching Varnish	6.0.6	<i>Web Caching</i>
---	-------------------------------------	-------	--------------------

3.2 ALUR PENELITIAN

Pada saat melakukan penelitian terdapat beberapa tahapan yang dilakukan untuk mendapatkan hasil yang maksimal. Jalannya penelitian dapat dilihat pada Gambar 3.1. Berikut merupakan penjelasan dari setiap tahap alur penelitian pada Gambar 3.1.

1. Tahap pertama, Studi Literatur. Mencari dan memahami literatur mengenai kinerja web server, terutama mengenai pendekatan-pendekatan dan teknik-teknik dalam menilai kinerja web server.
2. Tahap Kedua, Instalasi Ubuntu OS pada VirtualBox, Melakukan tahap instalasi seperti mengatur seberapa besar Ram dan Penyimpanan yang akan dibagikan dari perangkat peneliti kepada Ubuntu OS yang akan di install pada VirtualBox.
3. Tahap Ketiga, Instalasi Web Server dan Apache Benchmark, Melakukan instalasi *web server* Apache dan melakukan *test* pada URL default sampai muncul laman Apache *web server* dilanjutkan dengan instalasi Apache Benchmark sebagai *tool* pengujian kinerja *web server*.
4. Tahap Keempat, Setup *reverse proxy caching* Nginx dan Varnish, Melakukan konfigurasi *reverse proxy caching* pada *web server* Apache baik itu Nginx dan Varnish lalu *restart web server* hingga berhasil.
5. Tahap Kelima, Pengujian Web Server dengan parameter *Request per Second, Transfer Rate, Time taken for test, Time per Request*, Melakukan pengujian *web server* dengan Apache Benchmark *Tool* dengan mengamati parameter *Request per Second, Transfer Rate, Time taken for test*, dan *Time per Request*.
6. Tahap Keenam, Analisis Hasil Pengujian, Mengamati masing masing parameter dan membandingkan performa mana yang lebih baik antara *web server* Apache yang sudah dikonfigurasi *reverse proxy caching* Nginx dan Varnish.

7. Tahap Ketujuh, Kesimpulan dan Saran, Menyimpulkan dari kedua *reverse proxy caching* mana yang lebih baik dengan menyebutkan parameter mana yang meningkat, dan memberikan saran kepada peneliti selanjutnya apa saja yang harus ditambahkan Ketika akan melakukan penelitian yang sama.



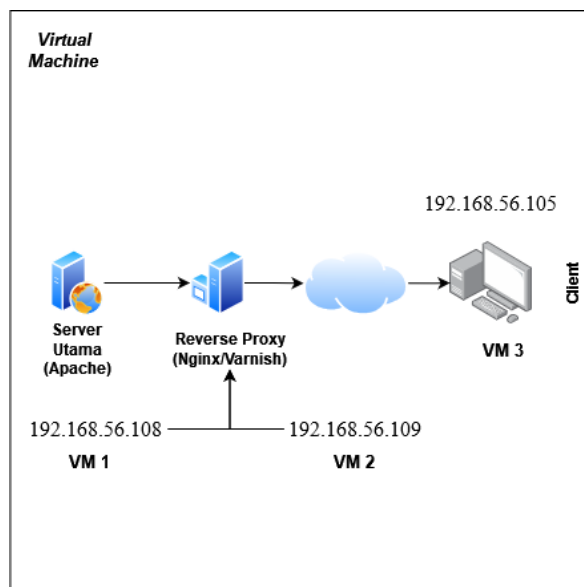
Gambar 3.1 *Flowchart* alur penelitian

Pada saat penyusunan tugas akhir ini memerlukan rencana agar proses pengerjaan penelitian tugas akhir ini dapat berjalan dengan maksimal dan sesuai dengan rencana yang sudah ditentukan sebelumnya. Pada Gambar 3.1 menunjukkan diagram alur penelitian pada tugas akhir ini. Pertama untuk pengerjaan tugas akhir ini penulis melakukan studi literatur, di mana meneliti pada berbagai literatur yang berhubungan dengan penelitian studi terhadap sistem sejenis yang sudah dibangun agar mendapatkan gambaran secara menyeluruh terhadap sistem yang akan dibangun. Selanjutnya melakukan instalasi Ubuntu OS pada VirtualBox sebagai *virtual environment* pada laptop. Jika sudah berhasil maka dilanjutkan instalasi *web server*, Apache benchmark sebagai *Tool* pengujian pada penelitian ini.

kemudian melakukan pengujian pada *web server* dengan mengamati parameter *Time Taken for Tests*, *Time per Request (ms)*, *Transfer Rate* dan *Request per Second* dengan menggunakan Apache Benchmark *Tool*, lalu setiap parameter yang telah dikumpulkan selanjutnya dilakukan analisis pada hasil pengujian kemudian diakhiri dengan kesimpulan dan saran penulis pada penelitian untuk tugas akhir ini.

3.3 TOPOLOGI PENELITIAN

Pada penelitian ini, *reverse proxy caching nginx* dan *varnish* berperan sebagai perantara antara web server utama dan client seperti Gambar 3.2



Gambar 3.2 Topologi Client dan Web Server

Pada gambar topologi diatas menunjukkan bahwa server proxy akan tampak seperti server utama untuk client penghubung manapun, artinya server proxy yang akan meneruskan permintaan dari client ke server utama dan meneruskan konten atau informasi dari server utama kembali ke client lagi. Istilah “server utama” digunakan untuk menggambarkan server tempat informasi atau konten berasal. Pada Tabel 3.4 menunjukkan table IP pada topologi yang dibuat

Tabel 3.4 Tabel IP Topologi Client dan Web Server

Web Server		Client
<i>Reverse proxy caching</i> <i>Nginx</i>	<i>Reverse proxy caching</i> <i>Varnish</i>	
192.168.56.108	192.168.56.109	192.168.56.105

3.4 INSTALASI DAN KONFIGURASI SERVER

Pada penelitian ini membangun sebuah *web server* Apache yang akan di *install* dengan *reverse proxy caching* Nginx dan Varnish. Adapun Server yang digunakan adalah virtual Server dengan menggunakan aplikasi virtual box. Kemudian *Web server* akan dikonfigurasi dengan menggunakan *reverse proxy caching nginx* sebagai *web accelerator* dari Server virtual. Pada penelitian ini juga melakukan pengukuran perfomansi Server dengan parameter *time taken for test*, *request per second*, *time per request (ms)*, dan *transfer rate*. Nantinya akan dilakukan perbandingan performasi *web server* antara *web server* apache dengan konfigurasi *reverse proxy* Nginx dan tanpa konfigurasi *reverse proxy* Nginx. Berikut merupakan simulasi dalam melakukan instalasi yang diperlukan dalam membuat *web server*:

A. INSTALASI WEB SERVER APACHE

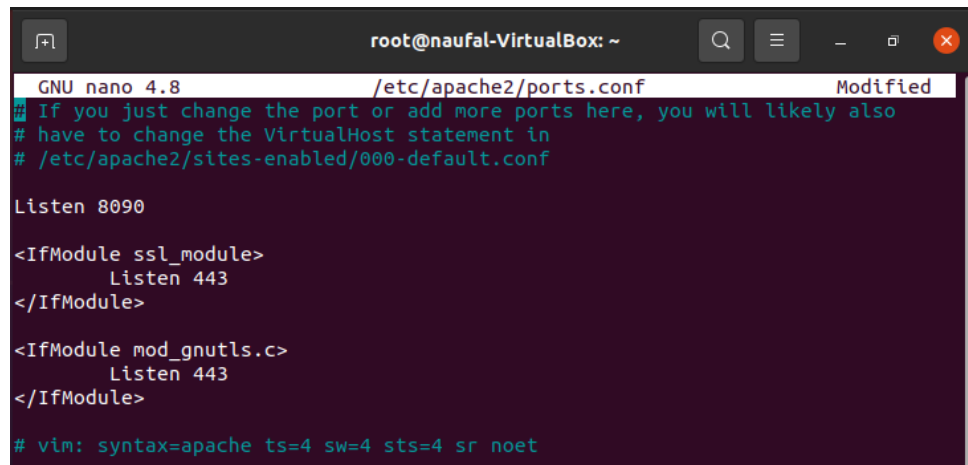
Pengujian dari server apache2 dilakukan untuk melihat perbandingan performa dari kedua perangkat lunak yang nantinya dijadikan sebagai bahan pertimbangan untuk diimplementasikan pada *web server* Apache. Sebelum dilakukan pengujian, proses pengerjaan server memerlukan software yang berfungsi sebagai media pelayanan *web server*, Langkah awal yang akan dilakukan adalah menginstall apache2 dengan melakukan perintah

pada terminal linux untuk mendapatkan mod-php,php-mysql, dan mod-rpaf,sebelum melakukan konfigurasi dilakukan perintah untuk menginstall paket apache2 dengan mengetikkan

```
apt install apache2 php libapache2-mod-php php-mysql libapache2-  
mod-rpaf
```

B. KONFIGURASI APACHE

Setelah instalasi selesai maka akan dilakukan konfigurasi pada port apache di dalam /etc/apache2/ports.conf. Lalu rubah port 80 menjadi 8090 seperti pada gambar 3.3 agar tidak terjadi crash saat nanti akan menginstall *reverse proxy nginx*.



```
GNU nano 4.8 /etc/apache2/ports.conf Modified  
# If you just change the port or add more ports here, you will likely also  
# have to change the VirtualHost statement in  
# /etc/apache2/sites-enabled/000-default.conf  
  
Listen 8090  
  
<IfModule ssl_module>  
    Listen 443  
</IfModule>  
  
<IfModule mod_gnutls.c>  
    Listen 443  
</IfModule>  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Gambar 3.3 port apache2

Kemudian lakukan konfigurasi pada /etc/apache2/sites-available/000-default.conf untuk mengarahkan document root pada /var/www/html/192.168.56.105 setelah itu buat *directory* dengan mengetikkan perintah

```
Mkdir /var/www/html/192.168.56.105
```

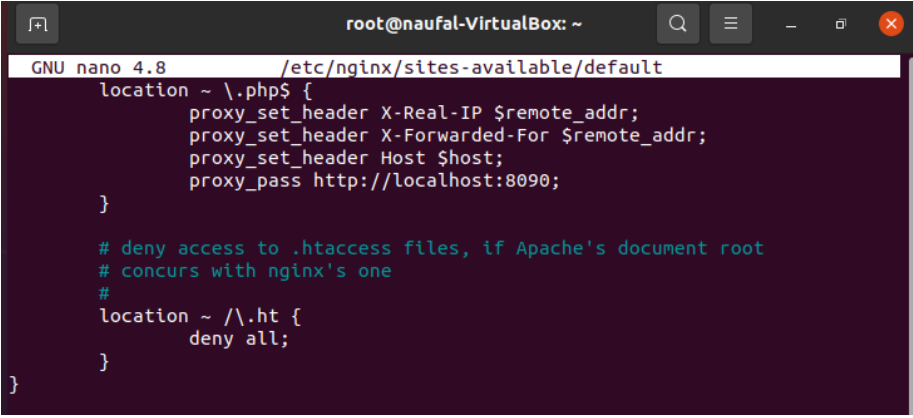
Setelah membuat *directory root*, maka lakukan restart pada apache2, Jika tidak ada masalah saat melakukan restart maka sudah terinstall dengan baik, untuk mastikan apakah *web server* apache2 jalan di port yang sudah dirubah, maka lakukan test dengan `curl -I localhost:8090` maka akan menampilkan seperti pada gambar 3.4.

```
root@naufal-VirtualBox:~# curl -I localhost:8090
HTTP/1.1 200 OK
Date: Wed, 19 Jan 2022 15:39:53 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Wed, 19 Jan 2022 08:44:41 GMT
ETag: "2aa6-5d5eb6593a8ae"
Accept-Ranges: bytes
Content-Length: 10918
Vary: Accept-Encoding
Content-Type: text/html
```

Gambar 3.4 apache2 status

C. INSTALASI MODUL *REVERSE PROXY*

Kemudian setelah selesai menginstall apache2 maka dilanjutkan menginstall nginx, jika sudah selesai menginstall nginx maka lakukan konfigurasi pada `/etc/nginx/sites-available/default` untuk menambahkan *directory root* `/var/www/html/192.168.56.105` seperti yang dilakukan pada langkah sebelumnya, selanjutnya masukkan modul pendukung konfigurasi reverse proxy dan karena apache2 berada di server yang sama masukkan `localhost:8090` untuk `proxy_pass` pada baris `location php` seperti pada gambar 3.5.



```
root@naufal-VirtualBox: ~
GNU nano 4.8 /etc/nginx/sites-available/default
location ~ /\.php$ {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header Host $host;
    proxy_pass http://localhost:8090;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
location ~ /\.ht {
    deny all;
}
}
```

Gambar 3.5 Modul *reverse proxy*

Selanjutnya *restart service* nginx, jika tidak ada masalah berarti sudah terkonfigurasi dengan benar dan server apache2 sudah bisa dikatakan terinstall konfigurasi *reverse proxy*.

D. INSTALASI APACHE BENCHMARK

lalu langkah terakhir lakukan instalasi untuk *benchmark Tools*, disini penulis menggunakan apache *benchmark Tools*, maka ketikkan perintah.

apt-get install apache2-utils

Selanjutnya jika sudah terinstall maka dapat di cek dengan menggunakan perintah `ab -V` untuk Apache benchmark seperti pada gambar 3.6.

```
root@naufal-VirtualBox:~# ab -V
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
```

Gambar 3.6 Apache benchmark

3.5 SKENARIO PENGUJIAN

Pengujian dilakukan dengan menggunakan aplikasi apache benchmarking. Aplikasi tersebut memberikan koneksi sesuai dengan yang di inginkan ke *web server*. Koneksi ini dapat dianalogikan jumlah pengunjung yang mencoba untuk mengakses *web server*.

Tabel 3.5 Skenario Pengujian

Variabel Pengujian	Web server	
	Reverse Proxy Caching Nginx	Reverse Proxy Caching Varnish
<i>Concurrency</i>	200	200
Koneksi	200, 2.000, 20.000	200, 2.000, 20.000

Pada Tabel 3.5 penelitian ini menggunakan variable pengujian berupa *concurrency*, dan koneksi. Variabel *Concurrency* hanya sebanyak 200, berbeda pada masing masing *web server* disebabkan karena keterbatasan dari perangkat peneliti, oleh karena itu angka yang ada disesuaikan dengan kemampuan dari perangkat/pc peneliti, pada koneksi yang dikirimkan secara bersamaan menggunakan Apache Benchmark *tools*, angka yang digunakan cukup besar agar dapat menghasilkan perbedaan pada hasil pengetesan. Dalam pengujian juga diperlukannya koneksi internet guna untuk melakukan test pada apache benchmarking *tools*. Pengujian dilakukan sebanyak 10 kali percobaan dan akan diambil rata rata dari total percobaan yang dilakukan, agar dapat mendapatkan nilai

yang maksimal. Berdasarkan Tabel 3.5, berikut deskripsi skenario pengujian yang dilakukan.

- a) Jumlah *request* yang dikirimkan dan Perintah pada saat melakukan Benchmark menggunakan Apache Benchmark *Tools*
 - 1) `ab -n 200 -c 200 http://[Address]/`
 - 2) `ab -n 2.000 -c 200 http://[Address]/`
 - 3) `ab -n 20.000 -c 200 http://[Address]/`
- b) Berikut URL atau halaman web yang akan digunakan untuk melakukan pengujian pada *web server*.
 - 1) `http://192.168.56.108`
 - 2) `http://192.168.56.109`

Pada pengujian *web server*, *virtual user* akan mengakses beberapa URL. Pada URL yang pertama dengan alamat `http://192.168.56.108` merupakan URL dengan *web server* yang dikonfigurasi dengan *reverse proxy caching* Nginx sedangkan `http://192.168.56.109` merupakan URL dengan *web server* yang dikonfigurasi dengan *reverse proxy caching* Varnish, masing masing URL akan diuji dengan Apache Benchmark *Tool* dengan *request* yang sudah ditentukan, Setelah itu akan dipantau dan diukur performansi Server dengan melihat parameter:

- a) *Time Taken for Tests*
- b) *Time per Request (ms)*
- c) *Tansfer Rate*
- d) *Request Per Second*

Berdasarkan scenario pengujian yang dibuat, penulis akan melakukan pengujian dengan menggunakan Apache Benchmark *tools*. Untuk mengukur performansi *web server* menggunakan Apache *Benchmarking Tool* dari sisi client dapat menggunakan perintah :

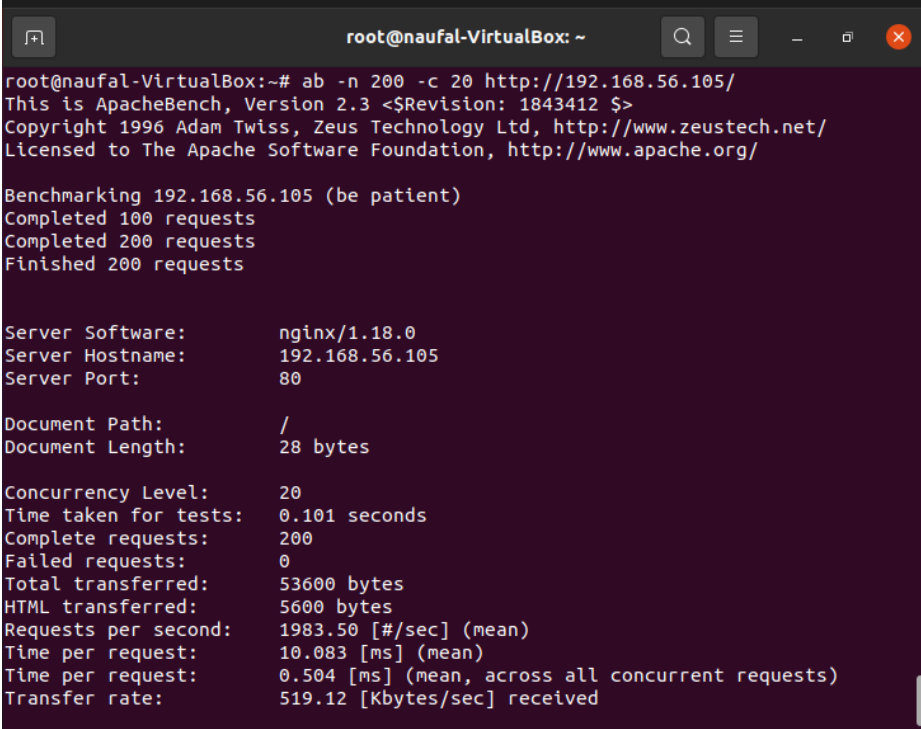
```
ab -n 200 -c 20 http://[Address]/
```

Berikut merupakan parameter yang digunakan oleh Apache Benchmarking *tools*:

- a) Parameter *n* adalah jumlah koneksi yang dibuat ke Server tujuan, dengan contoh diatas berarti koneksi yang dibuat adalah 200 koneksi.

- b) Parameter c adalah jumlah *request concurrent* (bersama) yang dibuat, dengan contoh diatas berarti jumlah request yang dibuat adalah 20 request dalam satu waktu.
- c) Parameter terakhir adalah Address. Address dapat berupa alamat IP atau halaman yang akan diproses oleh *web server* di benchmark.

Gambar 3.7 adalah contoh dari pengujian menggunakan Apache *Benchmarking Tool* dari sisi client.



```
root@naufal-VirtualBox: ~  
root@naufal-VirtualBox:~# ab -n 200 -c 20 http://192.168.56.105/  
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>  
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/  
Licensed to The Apache Software Foundation, http://www.apache.org/  
  
Benchmarking 192.168.56.105 (be patient)  
Completed 100 requests  
Completed 200 requests  
Finished 200 requests  
  
Server Software:      nginx/1.18.0  
Server Hostname:     192.168.56.105  
Server Port:         80  
  
Document Path:       /  
Document Length:     28 bytes  
  
Concurrency Level:   20  
Time taken for tests: 0.101 seconds  
Complete requests:   200  
Failed requests:     0  
Total transferred:   53600 bytes  
HTML transferred:    5600 bytes  
Requests per second: 1983.50 [#/sec] (mean)  
Time per request:    10.083 [ms] (mean)  
Time per request:    0.504 [ms] (mean, across all concurrent requests)  
Transfer rate:       519.12 [Kbytes/sec] received
```

Gambar 3.7 Contoh hasil pengujian Apache *Benchmarking Tool* dari sisi client