

BAB 3

METODOLOGI PENELITIAN

3.1 PERANGKAT YANG DIGUNAKAN

3.1.1 PERANGKAT KERAS (*HARDWARE*)

Perangkat keras yang digunakan pada penelitian ini yaitu menggunakan 1 buah laptop dengan spesifikasi sebagaimana terdapat pada tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras

Laptop	OS	Windows 10
	<i>Processor</i>	AMD A9
	RAM	8 GB
	<i>Hard Disk</i>	476 GB

3.1.2 PERANGKAT LUNAK (*SOFTWARE*)

3.1.2.1 PERANGKAT VIRTUAL

Perangkat virtual yang digunakan pada penelitian ini yaitu 2 perangkat virtual yang dibangun pada *Oracle VM Virtualbox* dengan sistem operasi *Debian 9*, satu perangkat sebagai *Client* dan satu perangkat diberikan instalasi 2 buah *web server* pada 1 *server*. Untuk spesifikasi perangkat virtual terdapat pada tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Lunak

<i>Client</i>	OS	<i>Debian 9</i>
	RAM	1 GB
	CPU	1 Core
	Alamat IP	2001:db8::8/64
<i>Server</i>	OS	<i>Debian 9</i>
	RAM	1 GB
	CPU	1 Core
	Alamat IP	2001:db8::5/64

3.1.2.2 SOFTWARE TOOL

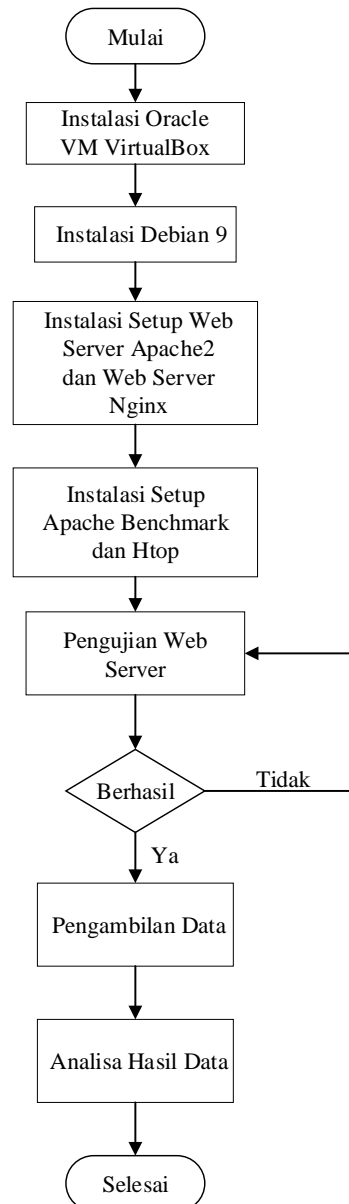
Tool dan aplikasi perangkat lunak yang digunakan pada penelitian ini tercantum pada tabel 3.3.

Tabel 3.3 Spesifikasi *Software Tool*

No	<i>Software</i>	Versi	Fungsi
1	<i>Apache</i>	2.4.25	<i>Web Server</i>
2	<i>Nginx</i>	1.10.3	<i>Web Server</i>
3	<i>Apache Benchmarking</i>	2.3	<i>Tool pengujian performansi web server</i>
4	<i>Htop</i>	3.0.5	<i>Tool monitoring penggunaan memori pada pengujian web server</i>
5	<i>Oracle VM Virtualbox</i>	6.1.26	Media Virtualisasi

3.2 ALUR PENELITIAN

Penelitian ini dilakukan dengan dengan beberapa proses tahapan seperti yang tertera pada diagram *flowchart* di bawah ini :



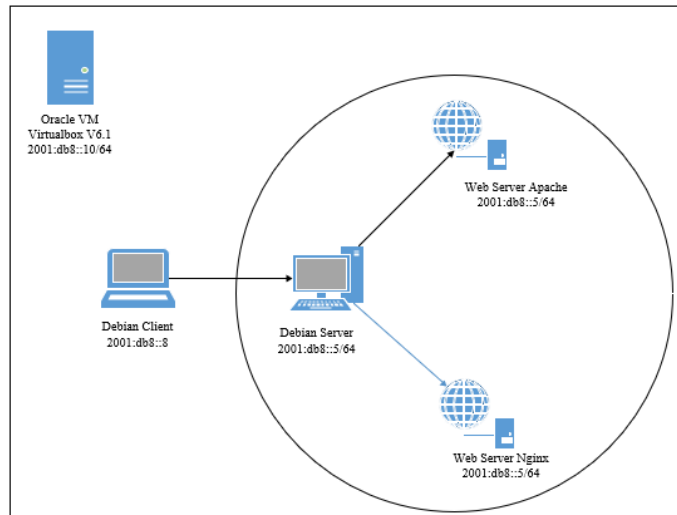
Gambar 3.1 Diagram Alur Penelitian

Tahapan proses penelitian yang ditunjukkan pada gambar 3.1 diawali dengan melakukan instalasi *Oracle VM Virtualbox* pada laptop yang digunakan sebagai media virtualisasi yang akan dibangun. Saat instalasi *Oracle VM Virtualbox* telah berhasil, selanjutnya yaitu melakukan instalasi sistem operasi yang akan digunakan dimana pada penelitian ini menggunakan sistem operasi *Debian 9*. Setelah instalasi

sistem operasi dilakukan, selanjutnya yaitu melakukan instalasi *setup web server Apache* dan *web server Nginx*, instalasi setup kedua *web server* tersebut dilakukan pada satu sistem operasi *Debian 9*, oleh karena itu dalam satu OS terdapat 2 *setup web server* yang terinstal. Untuk memeriksa *web server* telah terinstal dan siap digunakan yakni dengan melakukan pencarian pada *browser* dengan alamat IP yang digunakan pada *web server*, saat hasil pencarian menampilkan halaman utama *web server* maka *web server* telah siap digunakan. Saat setup *web server* telah terinstal tahap selanjutnya yaitu melakukan instalasi *Apache Benchmark* dan *htop* sebagai *software tool*. Saat *Apache Benchmark* telah berhasil terinstal maka tahapan selanjutnya yaitu melakukan pengujian terhadap *web server Apache* dan *web server Nginx* dengan menggunakan *tool Apache Bench* dan *Htop*, dimana parameter yang diuji yaitu *Time Taken for Test*, *Request per second*, *Transfer Rate (Kb/s)*, *Time per Request (ms)*, dan *Memory Usage*. Pengujian dilakukan secara bertahap antara *web server Apache* dan *web server Nginx*. Dikarenakan kedua *web server* tidak dapat berjalan bersamaan karena menggunakan *port* yang sama, maka pengujian dilakukan pada *web server Apache* terlebih dahulu kemudian selanjutnya pengujian pada *web server Nginx*. Kemudian setelah dilakukan pengujian, selanjutnya yaitu pengambilan data hasil dari pengujian kedua *web server* tersebut. Akan tetapi jika saat pengujian *web server* tidak berhasil maka harus kembali mengulang pengujian dengan memperhatikan parameter pada *Apache Bench*. Saat data hasil pengujian telah didapatkan maka tahap selanjutnya yaitu melakukan perbandingan dari kedua hasil data dan kemudian dapat dianalisa dan disimpulkan.

3.3 TOPOLOGI JARINGAN

Topologi jaringan yang digunakan pada penelitian ini terdiri dari 1 buah *device* laptop yang digunakan sebagai *client*, dimana *client* yang akan melakukan atau mengirimkan *request* pada *web server*. *Oracle VM Virtualbox* sebagai *software* virtualisasi. Pada *Oracle Virtualbox* dibangun *Debian 9* sebagai sistem operasi dan di berikan layanan 2 buah *web server* yaitu *web server Apache* dan *web server Nginx* seperti yang ditunjukkan pada gambar 3.2 dengan pengalamatan IPv6 yang digunakan seperti pada tabel 3.4.

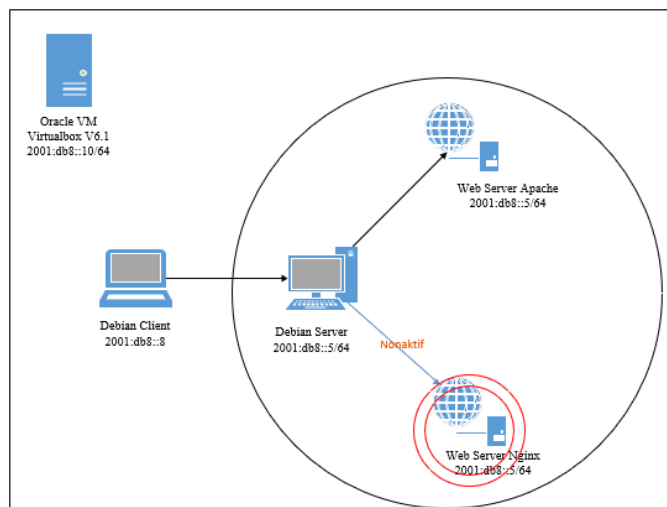


Gambar 3.2 Topologi Jaringan Sistem

Tabel 3.4 Pengalamatan IPv6

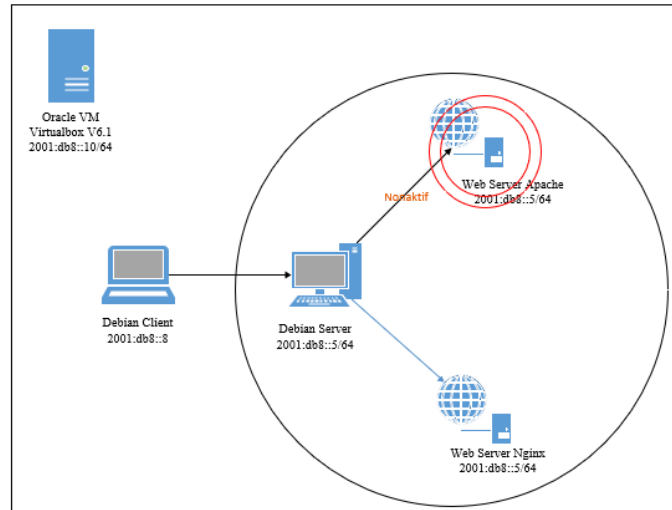
Alamat IP	Penggunaan
2001:db8::10/64	<i>Oracle VM Virtualbox</i>
2001:db8::8/64	<i>Client</i>
2001:db8::5/64	<i>Webserver Apache dan Webserver Nginx</i>

Untuk melakukan pengujian pada *web server Apache* pada sistem operasi *Debian 9*, maka *web server Nginx* dinonaktifkan terlebih dahulu karena kedua *web server* tidak dapat berjalan bersama-sama karena menggunakan *port* yang sama seperti yang ditunjukkan pada gambar 3.3.



Gambar 3.3 Topologi Pengujian pada *web server Apache*

Untuk melakukan pengujian pada *web server Nginx* pada operasi sistem *Debian 9*, maka *web server Apache* dinonaktifkan terlebih dahulu karena kedua *web server* tidak dapat berjalan bersama-sama karena menggunakan *port* yang sama seperti yang ditunjukkan pada gambar 3.4.



Gambar 3.4 Topologi Pengujian *web server Nginx*

3.4 SKENARIO PENGUJIAN

3.4.1 UJI COBA WEB SERVER

3.4.1.1 KONFIGURASI *WEB SERVER APACHE* DAN *WEB SERVER NGINX*

Pada penelitian ini menggunakan *web server Apache* dan *web server Nginx* dengan sistem operasi *Debian 9*. Pada penelitian ini dalam satu sistem operasi diberikan dua layanan *web server Apache* dan *web server Nginx*. Untuk pengalamatan pada *web server* digunakan pengalamatan IPv6 dengan satu alamat IP untuk kedua *web server*. Untuk menginstall instalasi *web server Apache* pada *debian* dilakukan dengan *command* pada *root apt install apache2*, seperti perintah yang ditunjukkan pada gambar 3.5.

```
apt install apache2
```

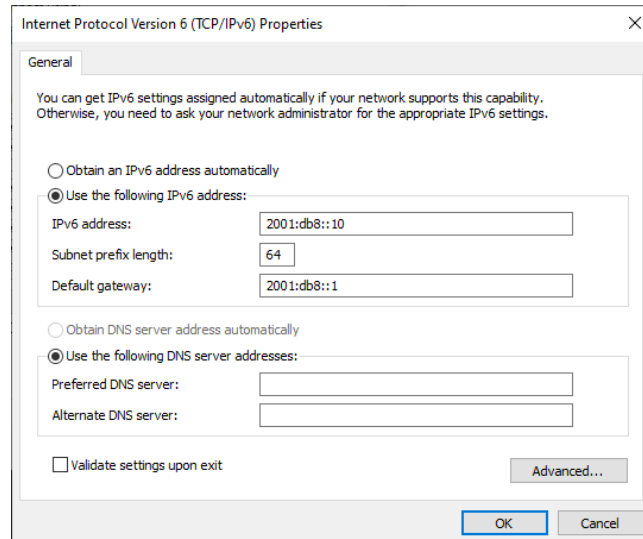
Gambar 3.5 *Command* install *web server Apache*

Untuk menginstall instalasi *web server Nginx* pada *debian* dilakukan dengan *command root apt install nginx*, seperti perintah yang ditunjukkan pada gambar 3.6.

```
apt install nginx
```

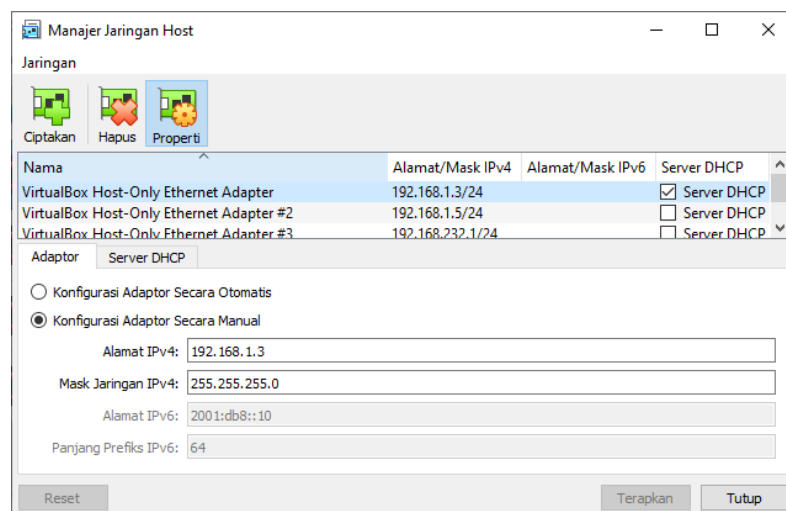
Gambar 3.6 *Command* install *web server Nginx*

Untuk menambahkan alamat IPv6 yang akan digunakan dengan menambahkan pada pengaturan *Network Connections* pilih *Adaptor* yang digunakan lalu pada IPv6 ditambahkan alamat IPv6 yang akan digunakan, penulis menggunakan alamat IPv6 2001:db8::10/64 dengan *gateway* 2001:db8::1



Gambar 3.7 Menambahkan alamat IPv6 yang akan digunakan pada *network adaptor*

Untuk memeriksa alamat IPv6 telah ditambahkan pada *Network adaptor* yang digunakan dapat dilihat pada *Manajer Jaringan Host* pada *server*, ditunjukkan pada gambar 3.7 bahwa alamat IPv6 telah berhasil ditambahkan.



Gambar 3.8 Alamat IPv6 pada *Host VirtualBox*

Untuk menambahkan alamat IPv6 yang akan digunakan pada *debian server* dilakukan dengan mengetikkan *command* `ip address add 2001:db8::5/64 dev`

enp0s8. Pada *command* yang ditunjukkan pada gambar 3.8, IP ditambahkan pada *devstack* enp0s8.

```
ip address add 2001:db8::5/64 dev enp0s8_
```

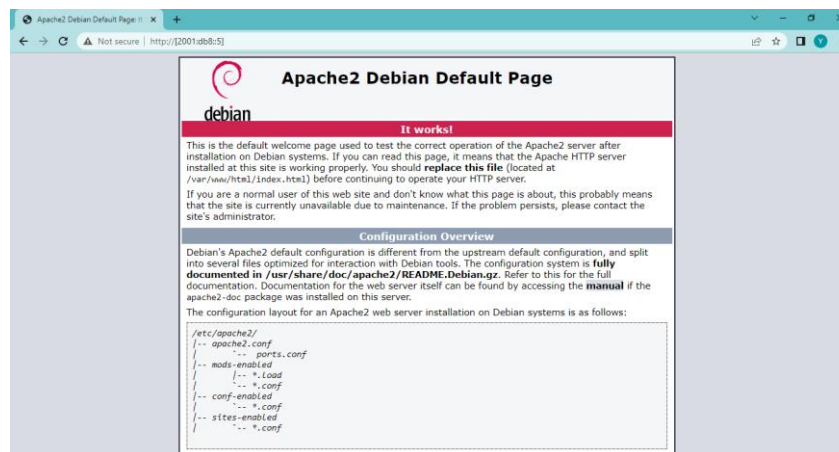
Gambar 3.9 Menambahkan alamat IPv6 pada *debian server*

Untuk memastikan IPv6 telah berhasil ditambahkan dengan menetikkan *command* pada *root* ip a, seperti yang ditunjukkan pada gambar 3.9 menunjukkan bahwa alamat IPv6 2001:db8::05/64 berhasil ditambahkan pada *devstack* enp0s8.

```
root@yulinanur:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:74:da:49 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe74:da49/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:d0:54:f5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 2001:db8::5/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed0:54f5/64 scope link
        valid_lft forever preferred_lft forever
```

Gambar 3.10 Pengalamatan IPv6 pada *debian server*

Untuk memeriksa dan memastikan *web server Apache* dan *web server Nginx* telah berhasil dikonfigurasi dan siap digunakan yaitu dengan melakukan pencarian pada *browser* dengan alamat IP yang digunakan pada *web server*, saat hasil pencarian menampilkan halaman utama *web server* maka *web server* telah siap digunakan. Seperti yang ditunjukkan pada gambar 3.10 menunjukkan bahwa saat *web server Apache* sedang aktif maka saat dilakukan pencarian pada laman *web browser* dengan alamat IP 2001:db8::05 maka akan menghasilkan tampilan *web server Apache2 Default Page* yang berisi informasi *web server Apache*.



Gambar 3.11 Tampilan laman utama *web server Apache*.

Sedangkan untuk mengaktifkan *web server Nginx* maka *web server Apache* pada server harus dinonaktifkan terlebih dahulu karena kedua *web server* tidak dapat berjalan secara bersama-sama karena menggunakan *port* yang sama. Saat *web server Nginx* telah aktif maka saat dilakukan pencarian pada *web browser* dengan alamat 2001:db8::05 maka akan ditampilkan laman utama *web server Nginx* seperti yang ditampilkan pada gambar 3.11.



Gambar 3.12 Tampilan laman utama *web server Nginx*.

3.4.1.2 KONFIGURASI DAN PENGUJIAN DENGAN *APACHE BENCHMARK* PADA *WEB SERVER APACHE*

Konfigurasi *Apache Bench* pada *web server Apache* telah terkonfigurasi dan terinstal secara otomatis saat menginstal *Apache* dalam sistem operasi. Pengujian menggunakan *Apache Bench* dilakukan untuk memberikan koneksi yang sesuai dengan permintaan *client* ke *web server Apache*. Pada penelitian ini diberikan koneksi yang ditujukaan pada server sebanyak 500, 1000, 5000, 8000, 10000, 15000, 20000, 25000, 30000, 35000, 45000, 50000, 65000, 75000, 85000, 100000, 150000, 200000, 250000, 350000 dan 500000 koneksi, dan jumlah permintaan yang dilakukan sekaligus atau bebarengan sebanyak 100 permintaan pada *web server Apache* seperti yang terdapat pada tabel 3.6. Untuk mengukur performansi *web server* dengan *Apache Bench* dapat digunakan perintah `ab -c 100 -n 5000 http://[Address]/`

Tabel 3.5 Parameter *Apache Benchmark*

Parameter	Fungsi
<i>ab</i>	<i>Command</i> untuk menggunakan <i>Apache Benchmark</i>
<i>c</i>	<i>Request concurrent</i> (bersamaan) dalam satu waktu
<i>n</i>	Koneksi yang dibuat ke <i>server</i> tujuan
<i>Address</i>	Alamat IP atau halaman yang ingin diproses <i>web server</i> pada <i>Apache Benchmark</i>

```
ab -c 100 -n 15000 http://[2001:db8::5]/
```

Gambar 3.13 *Command* pengujian *Apache Benchmark* pada *web server Apache* dengan koneksi 15000

```
ab -c 100 -n 30000 http://[2001:db8::5]/
```

Gambar 3.14 *Command* pengujian *Apache Benchmark* pada *web server Apache* dengan koneksi 30000

Gambar 3.13 menunjukkan perintah yang digunakan untuk melakukan pengujian dengan parameter *c* merupakan jumlah permintaan yang bersamaan yang dibuat dalam satu waktu, dengan contoh dalam perintah yaitu 100 permintaan bersamaan dalam satu waktu. Pada parameter *n* merupakan jumlah koneksi yang dibuat ke *server* tujuan, dengan contoh dalam perintah yaitu 15000 koneksi untuk *server* tujuan. Dan pada gambar 3.14 menunjukkan diberikan 100 permintaan bersamaan dalam satu waktu dan 500000 koneksi untuk *server* tujuan. Parameter *Address* dapat berupa alamat IP atau halaman yang diinginkan yang kemudian akan diproses oleh *web server*.

Tabel 3.6 Parameter Pengujian

Pengujian ke-	Jumlah permintaan per detik	Jumlah koneksi
1	100	500
2	100	1000
3	100	5000
4	100	8000
5	100	10000
6	100	15000
7	100	20000
8	100	25000
9	100	30000
10	100	35000
11	100	45000
12	100	50000
13	100	65000

Pengujian ke-	Jumlah permintaan per detik	Jumlah koneksi
14	100	75000
15	100	85000
16	100	100000
17	100	150000
18	100	200000
19	100	350000
20	100	500000

```

Server Software:      Apache/2.4.53
Server Hostname:     2001:db8::5
Server Port:         80

Document Path:       /
Document Length:     10701 bytes

Concurrency Level:    100
Time taken for tests: 26.172 seconds
Complete requests:   15000
Failed requests:     0
Total transferred:   164625000 bytes
HTML transferred:    160515000 bytes
Requests per second: 573.14 [#/sec] (mean)
Time per request:    174.479 [ms] (mean)
Time per request:    1.745 [ms] (mean, across all concurrent requests)
Transfer rate:       6142.74 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    2   4.4      1    66
Processing:  1  172 124.7    169  2335
Waiting:    1  131  78.8    139  1676
Total:      2  174 124.6    170  2343

Percentage of the requests served within a certain time (ms)
 50%    170
 66%    178

```

Gambar 3.15 Hasil pengujian dengan *Apache Benchmark* pada *web server Apache* dengan koneksi pada *server 15000*

```

Server Software:      Apache/2.4.53
Server Hostname:     2001:db8::5
Server Port:         80

Document Path:       /
Document Length:     10701 bytes

Concurrency Level:    100
Time taken for tests: 70.661 seconds
Complete requests:   30000
Failed requests:     0
Total transferred:   329250000 bytes
HTML transferred:    321030000 bytes
Requests per second: 424.56 [#/sec] (mean)
Time per request:    235.535 [ms] (mean)
Time per request:    2.355 [ms] (mean, across all concurrent requests)
Transfer rate:       4550.39 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    3  12.3      1    277
Processing:  1  232 381.8    139  22374
Waiting:    1  128 229.3     91  22363
Total:      2  235 382.4    141  22375

Percentage of the requests served within a certain time (ms)
 50%    141
 66%    230

```

Gambar 3.16 Hasil pengujian dengan *Apache Benchmark* pada *web server Apache* dengan koneksi pada *server 30000*

3.4.1.3 KONFIGURASI DAN PENGUJIAN DENGAN *APACHE BENCHMARK* PADA *WEB SERVER NGINX*

Konfigurasi *Apache Bench* pada *web server Nginx* dapat langsung otomatis di gunakan karena telah tersedia saat melakukan instalasi *web server Apache* dalam sistem operasi. Pengujian menggunakan *Apache Bench* dilakukan untuk memberikan koneksi yang sesuai dengan permintaan *client* ke *web server*. Pada penelitian ini diberikan koneksi yang ditujukan pada server sebanyak 500, 1000, 5000, 8000, 10000, 15000, 20000, 25000, 30000, 35000, 45000, 50000, 65000, 75000, 85000, 100000, 150000, 200000, 250000, 350000 dan 500000 koneksi, dan jumlah permintaan yang dilakukan sekaligus atau bebarengan sebanyak 100 permintaan pada *web server* seperti yang terdapat pada tabel 3.5. Untuk mengukur performansi *web server* dengan *Apache Bench* pada *web server Nginx* dapat digunakan perintah `ab -c 10 -n 5000 http://[Address]/`

```
ab -c 100 -n 500 http://[2001:db8::5]/
```

Gambar 3.17 *Command* pengujian *Apache Benchmark* pada *web server Apache* dengan koneksi 500

```
ab -c 100 -n 1000 http://[2001:db8::5]/
```

Gambar 3.18 *Command* pengujian *Apache Benchmark* pada *web server Apache* dengan koneksi 1000

Gambar 3.17 menunjukkan perintah yang digunakan untuk melakukan pengujian dengan parameter *c* merupakan jumlah permintaan yang bersamaan yang dibuat dalam satu waktu, dengan contoh dalam perintah yaitu 10 permintaan bersamaan dalam satu waktu. Pada parameter *n* merupakan jumlah koneksi yang dibuat ke *server* tujuan, dengan contoh dalam perintah yaitu 5000 koneksi untuk *server* tujuan. Dan pada gambar 3.18 menunjukkan diberikan 100 permintaan bersamaan dalam satu waktu dan 500000 koneksi untuk *server* tujuan. Parameter *Address* dapat berupa alamat IP atau halaman yang diinginkan yang kemudian akan diproses oleh *web server*.

```

Server Software:      nginx/1.10.3
Server Hostname:     2001:db8::5
Server Port:         80

Document Path:       /
Document Length:     10701 bytes

Concurrency Level:   100
Time taken for tests: 1.041 seconds
Complete requests:   500
Failed requests:     0
Total transferred:   5468500 bytes
HTML transferred:   5350500 bytes
Requests per second: 480.08 [#/sec] (mean)
Time per request:    208.297 [ms] (mean)
Time per request:    2.083 [ms] (mean, across all concurrent requests)
Transfer rate:       5127.62 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0   19  21.1    11   98
Processing: 31  160  65.3   167  281
Waiting:    1  153  68.6   159  273
Total:      97  179  53.8   175  314

Percentage of the requests served within a certain time (ms)
 50%    175
 66%    189
 75%    214

```

Gambar 3.19 Hasil pengujian dengan *Apache Benchmark* pada *web server Nginx* dengan koneksi pada *server 500*

```

Server Software:      nginx/1.10.3
Server Hostname:     2001:db8::5
Server Port:         80

Document Path:       /
Document Length:     10701 bytes

Concurrency Level:   100
Time taken for tests: 1.737 seconds
Complete requests:   1000
Failed requests:     0
Total transferred:   10937000 bytes
HTML transferred:   10701000 bytes
Requests per second: 575.75 [#/sec] (mean)
Time per request:    173.686 [ms] (mean)
Time per request:    1.737 [ms] (mean, across all concurrent requests)
Transfer rate:       6149.42 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0   47  26.5    48  103
Processing: 20  116  33.0   122  179
Waiting:    1   70  30.9    64  149
Total:      89  163  26.7   165  219

Percentage of the requests served within a certain time (ms)
 50%    165
 66%    174
 75%    180

```

Gambar 3.20 Hasil pengujian dengan *Apache Benchmark* pada *web server Nginx* dengan koneksi pada *server 1000*