

BAB III METODE PENELITIAN

Penelitian ini merupakan penelitian eksperimental yang dilakukan dengan menguji *QoS load balancing* dengan algoritma *round robin*. Percobaan dilakukan dengan mengirim *request rate* 1.000, 3.000, dan 5.000 pada *hybrid SDN* dan *pure SDN* masing – masing sebanyak 30 kali percobaan di setiap *request rate*. Penelitian ini menggunakan pendekatan kuantitatif dengan jenis data numerik dan statistik. Analisis data dilakukan dengan cara mencari nilai rata-rata di setiap parameter yaitu *throughput*, *delay*, *jitter*, dan *packet loss*. Kesimpulan didapat berdasarkan hasil analisis *QoS*.

3.1 PERANGKAT YANG DIGUNAKAN

3.1.1 HARDWARE

Pada penelitian ini menggunakan 1 perangkat keras yaitu laptop dengan sistem operasi dan spesifikasi *hardware*

Tabel 3. 1 Spesifikasi Laptop

Sistem Operasi	Linux Backbox 7
<i>Processor</i>	AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx (8 CPUs), ~2.3GHz
<i>RAM</i>	16 GB
<i>SSD</i>	512 GB

3.1.2 PERANGKAT LUNAK (SOFTWARE)

Perancangan sistem untuk penelitian ini, diperlukan sebuah perangkat lunak untuk dapat membantu merealisasikan sistem yang akan dibangun. Berikut spesifikasi perangkat keras yang digunakan :

3.1.2.1 PERANGKAT VIRTUAL

Pada penelitian ini terdapat 7 jenis perangkat *virtual* yang dibangun pada GNS3 VM yaitu : 1 *load balancer*, 2 *web server*, dan 2 perangkat *switch Software Defined Network*, 2 *Router konvensional*, 1 *controller Software Defined Network*, 1 *client*. Spesifikasi perangkat *virtual* tercantum pada tabel 3.2.

Tabel 3. 2 Spesifikasi Perangkat di *GNS3*

<i>LOAD BALANCER</i>	<i>OS</i>	Linux Backbox 7
	<i>RAM</i>	1024 MB
	<i>Storage</i>	4096 MB
	<i>vCPU</i>	1 Core
<i>WEB SERVER</i>	<i>OS</i>	Linux Backbox 7
	<i>RAM</i>	1024 MB
	<i>Storage</i>	4096 MB
	<i>vCPU</i>	1 Core
<i>FRR (Router)</i>	<i>OS</i>	Linux Backbox 7
	<i>RAM</i>	1024 MB
	<i>Storage</i>	4096 MB
	<i>CPU</i>	1 Core
<i>Open vSwitch</i>	<i>OS</i>	Linux Backbox 7
	<i>RAM</i>	1024 MB
	<i>Storage</i>	4096 MB
	<i>vCPU</i>	1 Core
<i>ONOS Controller</i>	<i>OS</i>	Docker
	<i>RAM</i>	2048 MB
	<i>Storage</i>	2048 MB
	<i>CPU</i>	2 Core
<i>Client</i>	<i>OS</i>	Linux Backbox 7
	<i>RAM</i>	1024 MB
	<i>Storage</i>	4096 MB
	<i>vCPU</i>	1 Core

Beberapa *software* yang digunakan dalam penelitian ini adalah sebagai berikut :

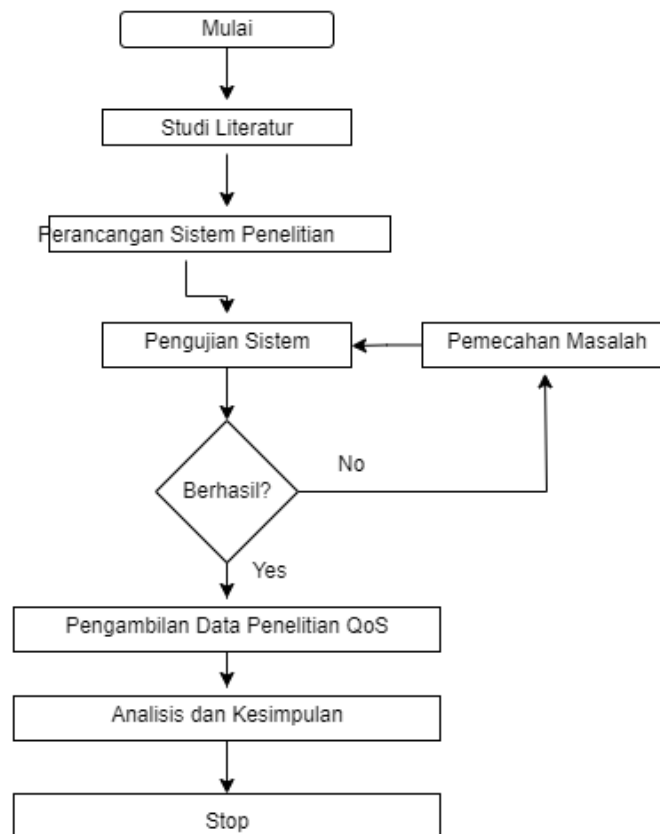
Tabel 3. 3 Daftar *Software*

No.	Nama Software	Versi	Fungsi
1.	<i>GNS 3</i>	2.2.24	Melakukan pendefinisian Topologi
2.	<i>Wireshark</i>	3.6.7	Analisis Paket Data pada jaringan
3.	<i>Httpperf</i>	0.9.0	Melakukan pengujian performansi pada <i>Web Server</i>
4.	<i>ONOS Controller</i>	2.0.0	Mengatur forwarding data pada SDN

No.	Nama Software	Versi	Fungsi
5.	<i>HAProxy</i>	1.8.8	<i>Load Balancing</i>

3.2 ALUR PENELITIAN

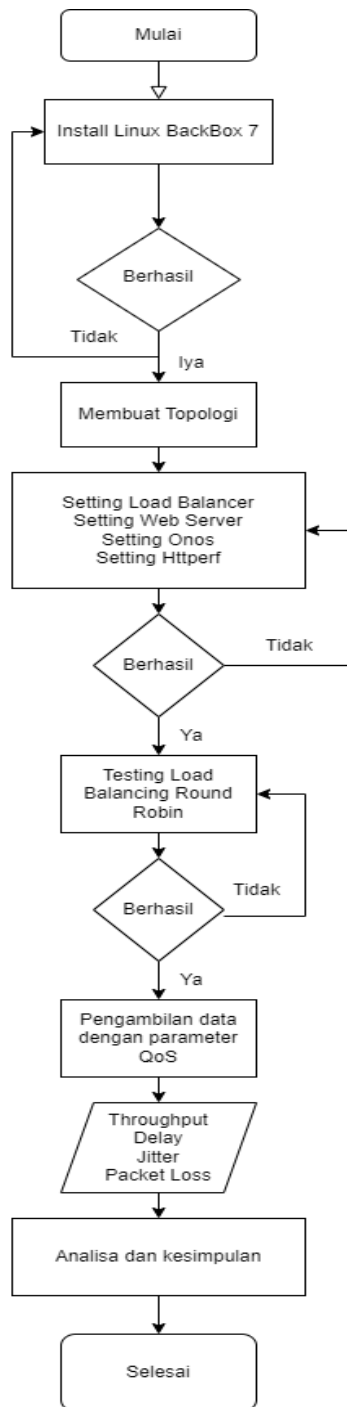
Penelitian dilakukan dalam beberapa tahap yaitu tahap studi literatur, tahap perancangan sistem, pembuatan simulasi, pengujian simulasi, tahap analisis dari hasil pengujian simulasi, dan yang terakhir analisis data dan kesimpulan Gambar 3.1



Gambar 3. 1 Flowchart Alur Penelitian

3.3 ALUR SIMULASI

Diagram alur simulasi menjelaskan mengenai gambaran prosedur yang dilakukan dalam proses pembuatan sistem dan dilakukan pengujian *load balancing* berdasarkan parameter *QoS*. Proses simulasi akan dijelaskan secara teknis melalui diagram *flowchart* yang ditujukan pada gambar 3.2 *flowchart* simulasi.

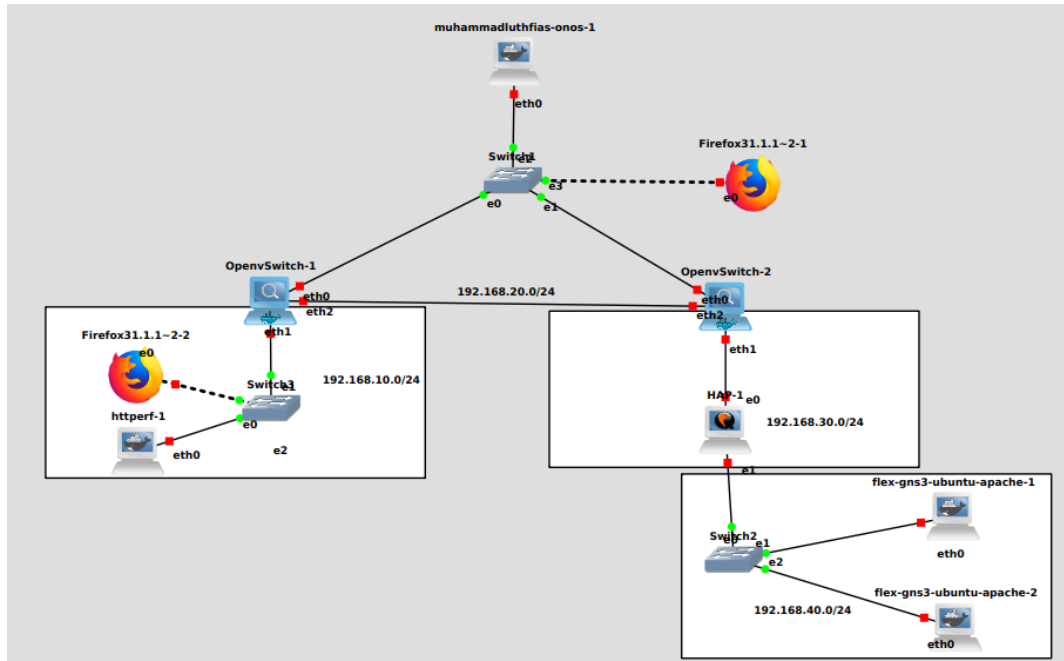


Gambar 3. 2 *Flowchart* Simulasi

Pada gambar 3.2 diperlihatkan *flowchart* simulasi sesuai urutan yang benar. Diawali dengan melakukan instalasi Sistem Operasi *Linux Backbox 7*. Jika berhasil, langkah selanjutnya adalah menentukan topologi yang akan digunakan pada percobaan analisis performansi *load balancing* dengan menggunakan algoritma *round robin* pada arsitektur *hybrid SDN* maupun *pure*

SDN. Langkah selanjutnya adalah instalasi *GNS3*, *web server*, *client*, *load balancer*, dan *controller* yang dipakai dalam pengujian simulasi. Jika berhasil, ujicoba *QoS load balancing* dilakukan dengan algoritma *round robin*. Selanjutnya, dilakukan pengujian kedua arsitektur melalui tahapan 4 parameter QoS yaitu, *throughput*, *delay*, *jitter* dan *packet loss*. Langkah terakhir yang dilakukan adalah menganalisis data dan membuat kesimpulan.

3.4 TOPOLOGI JARINGAN



Gambar 3. 3 Topologi Jaringan *Pure SDN*

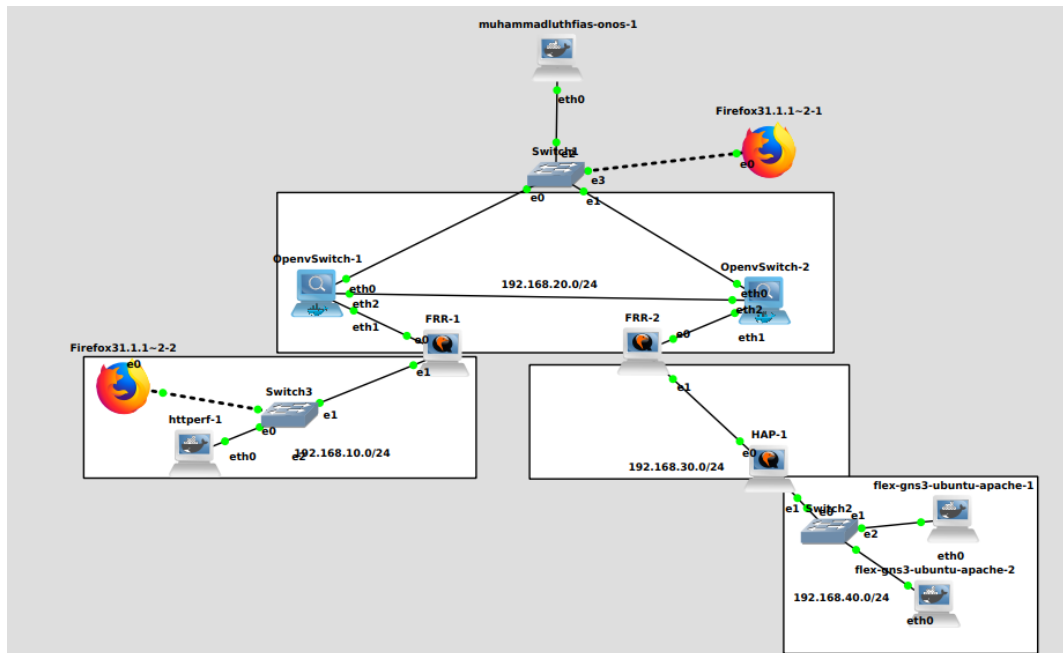
Berikut ini merupakan daftar IP dari topologi *Pure SDN* yang terdiri dari 1 *client*, 2 *OpennvSwitch*, 1 *load balancer*, 1 *controller*, dan 2 *web server*.

Tabel 3. 4 Daftar IP *Pure SDN*

Httpperf	Eth0	192.168.10.4
OpennvSwitch 1	Eth0	192.168.1.1
OpennvSwitch 2	Eth0	192.168.1.2
Load Balancer	Eth0	192.168.30.2
	Eth1	192.168.40.1
ONOS	Eth0	192.168.1.254
Web Server 1	Eth0	192.168.40.10

Web Server 2	Eth0	192.168.40.20
--------------	------	---------------

Topologi jaringan *hybrid SDN* pada penelitian ini ditunjukkan gambar 3.4 yang terdiri dari 1 *client*, 2 *switch*, 2 *router* konvensional, 2 *switch OpenFlow*, 2 *web server* dan 1 *controller* yang digunakan untuk melakukan simulasi pengujian.



Gambar 3. 4 Topologi Jaringan *Hybrid SDN*

Berikut ini merupakan data *IP* dari Topologi diatas yang terdiri dari 2 *FRR*Router, 2 *OpenvSwitch*, 1 *HAP*Proxy, 1 *Controller*, 2 *Web Server*

Tabel 3. 5 Daftar *IP Hybrid SDN*

Httpperf	Eth0	192.168.10.4
FRRouting 1	Eth0	192.168.20.1
	Eth1	192.168.10.1
FRRouting 2	Eth0	192.168.20.2
	Eth1	192.168.30.1
OpenvSwitch 1	Eth0	192.168.1.1
OpenvSwitch 2	Eth0	192.168.1.2
Load Balancer	Eth0	192.168.30.2
	Eth1	192.168.40.1
ONOS	Eth0	192.168.1.254
Web Server 1	Eth0	192.168.40.10
Web Server 2	Eth0	192.168.40.20

3.5 SKENARIO PENGUJIAN

Pada skenario pengujian akan di lakukan di jaringan berskala kecil khususnya untuk suatu gedung maupun instansi pemerintahan dengan menggunakan 1 algoritma dan 2 topologi jaringan yang berbeda, topologi *pure SDN* yang menggunakan algoritma *round robin* dengan topologi *hybrid SDN* yang menggunakan algoritma *round robin* dengan setiap perincian algoritma ditambahkan dengan ketentuan berikut.

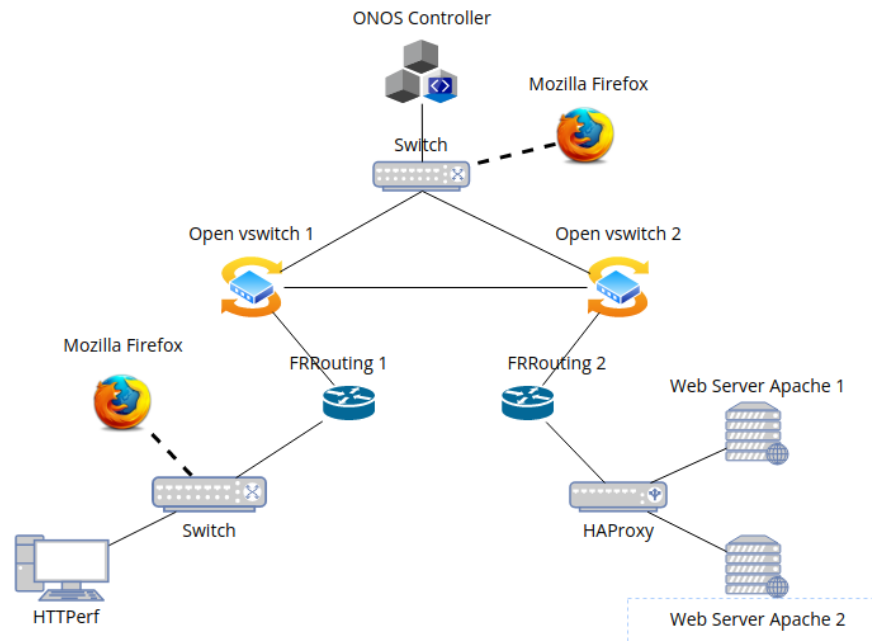
Tabel 3.3 Perincian Skenario Pengujian Topologi *Pure SDN*

Arsitektur	Algoritma	Jumlah Switch	Jumlah Host	Banyak Percobaan
Hybrid SDN	Round Robin	2	1	30
Pure SDN	Round Robin	2	1	30

Terdapat 4 parameter yang digunakan pada tugas akhir ini , yaitu *throughput*, *delay*, *jitter* dan *packet loss*. Selanjutnya, 4 parameter tersebut menjadi dasar untuk analisis sistem algoritma yang dilakukan agar dapat melakukan perbandingan nilai performansi *QoS* berdasarkan TIPHON. Proses pengukuran ini dilakukan dengan mengalirkan trafik pada topologi yang telah dibuat.

Trafik tersebut diukur untuk setiap pengujian *QoS* pada *web server*. Pengujian ini menggunakan *timeout* 5 ms. Masing- masing di uji dalam beberapa koneksi yaitu 1.000, 3.000 dan 5.000 *request*.

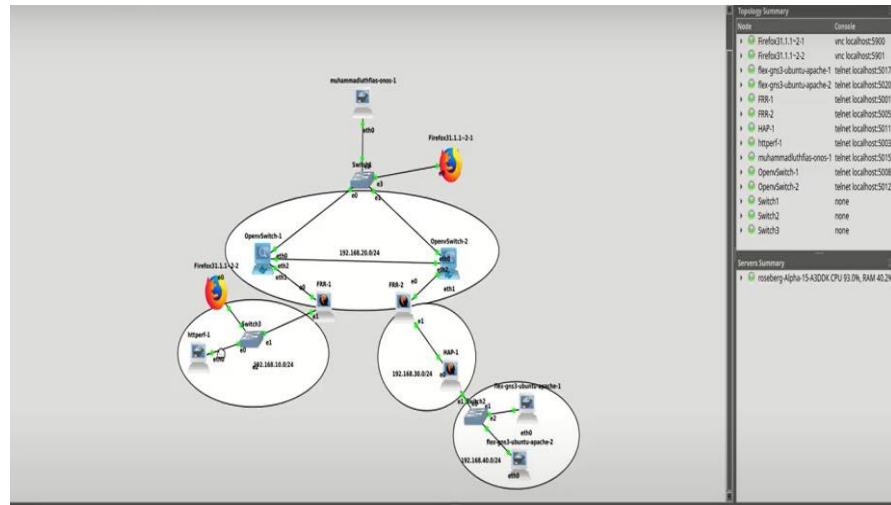
3.6 SKENARIO JARINGAN



Gambar 3. 5 Skenario jaringan *hybrid SDN*

Topologi jaringan *hybrid SDN* pada penelitian ini ditunjukkan gambar 3.4 yang terdiri dari 1 *client*, 2 *switch*, 2 *router* konvensional, 2 *switch OpenFlow*, 2 *web server Apache* dan 1 *controller* yang digunakan untuk melakukan simulasi pengujian. Semua keperluan tersebut sudah di-*install* di aplikasi *GNS3*. Pengujian dilakukan dengan memberikan beban *request* yang bervariasi ke *web server* yang sudah terdapat *load balancer*. *Request traffic* akan dibagi dan diteruskan ke *web server* lalu akan dilakukan penerapan *load balancer round robin*. *Load balancer* di-*install* pada aplikasi *haproxy* yang berfungsi sebagai *frontend* untuk membagi beban *request traffic* ke *web server* yang sudah di-*install Apache2* yang berfungsi sebagai *backend* yang akan memberikan *service* berupa layanan penerima. Kemudian, 1 komputer yang berjalan pada sistem operasi *Linux Backbox 7* yang sudah terinstall *Httpperf* bertindak sebagai *client* dan mengirimkan *request traffic packet* kepada *server*. Kemudian, aplikasi *Wireshark* akan meng-*capture* data yang diterima dan dikirim oleh *client* yang sudah ter-*install Httpperf*.

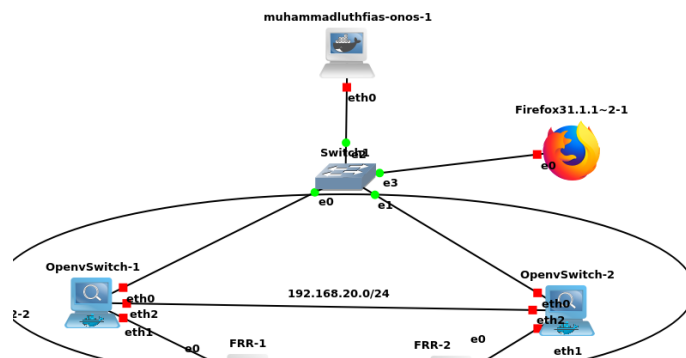
3.6.1 GNS3



Gambar 3. 6 Tampilan GNS3

Pada gambar 3.6 merupakan tampilan jendela awal *GNS3* yang berfungsi sebagai media *emulator* dalam melakukan pengujian pada penelitian ini yang mendekati pengalaman pemakaian seperti *real device*. Agar mendapatkan kemampuan seperti *real device*, perlu dilakukan *setting* pada setiap penambahan *device*. Pastikan membaca minimum spesifikasi perangkat agar pengujian dapat berjalan dengan lancar.

3.6.2 INTEGRASI OVS DENGAN ONOS CONTROLLER



Gambar 3. 7 Integrasi OVS dengan ONOS

Pada gambar 3.7, dapat dilihat integrasi *OVS* dengan *ONOS Controller*. Topologi yang digunakan adalah *mesh* yang terdiri dari 2 *OpenvSwitch*, 1 *switch*, dan *ONOS Controller*. Device diatas terinstall pada sistem operasi *Linux Backbox 7*. Untuk spesifikasi *hardware* pada

tiap *device* telah di-*setting* dan disamakan dengan tiap *OpenvSwitch* menggunakan 1 *core* dan 1024 MB *RAM*. *OpenvSwitch* digunakan sebagai *forwarder* dan bentuk implementasi dari *SDN*.

3.6.3 UJI COBA WEB SERVER

Pada gambar 3.8 merupakan perintah dalam melakukan instalasi *software* pendukung *web server*, yaitu *Apache2*. *Web server* ini diinstall pada 2 buah *virtual machine* yang berjalan pada sistem operasi *Linux Backbox 7*. Kedua *web server* ini memiliki spesifikasi yang sama yaitu, yaitu CPU 1 *core*, 1024 *RAM*, dan 4096 MB pada *storage*. Konfigurasi *web server* dilakukan pada setiap mesin virtual yang telah dibuat menggunakan beberapa perintah.

```
root@server1:~# apt install apache2_
```

Gambar 3. 8 *Install Apache2*

Setelah tahap instalasi selesai, maka perlu dipastikan bahwa *web server Apache2* sudah berhasil berjalan pada *virtual machine* dengan cara mengetikkan *IP Address web server* pada *browser*.

3.6.4 KONFIGURASI LOAD BALANCER

Konfigurasi *load balancer* dilakukan dengan cara melakukan instalasi aplikasi *Haproxy* terlebih dahulu. *Haproxy* sendiri berjalan pada sistem operasi *Linux Backbox 7* dan menggunakan algoritma *round robin* pada penelitian ini. Diharapkan dapat menyeimbangkan beban *traffic* pada kedua arsitektur yaitu *hybrid SDN* dan *pure SDN*.

```
sudo apt-get install haproxy
```

Gambar 3. 9 *Install HAProxy*

Selanjutnya, tambahkan konfigurasi *load balancing round robin* di file *haproxy.cfg* yang berada pada direktori */etc/haproxy/*. Lalu tambahkan kedua alamat *IP web server* yang sebelumnya sudah kita *setting*. Gunakan perintah “*roundrobin*” pada konfigurasi *haproxy* untuk menggunakan algoritma *round robin*.

Uji coba dilakukan dengan mengakses alamat *IP load balancer* melalui *browser* pada *client*. Secara otomatis kita akan diarahkan ke laman *default web server* sebagai target pada *server load balancer*. Selanjutnya, dilakukan uji coba sistem *load balancing* untuk mengetahui pembagian beban *traffic* ke masing – masing *web server* dengan algoritma *round robin* pada arsitektur *hybrid SDN* dan *pure SDN*.