

UJI PERFORMA KONTROLER SOFTWARE-DEFINE NETWORK FLOODLIGHT vs ONOS

Sizka L. Hanifa¹⁾ dan Rikie Kartadie²⁾

^{1, 2)} Pendidikan Teknologi Informasi, STKIP PGRI Tulungagung

Jln. Mayor Sujadi Timur No, 7 Tulungagung Kode Pos 66221

e-mail: sizkalailatu96@gmail.com¹⁾, rikie.kartadie@stkipgritulungagung.ac.id²⁾

ABSTRAK

Komponen yang penting dari Software-Defined Network adalah kontroler yang secara langsung mengontrol data path dari perangkat. Software-Defined Network atau yang sering kita sebut dengan istilah SDN itu sendiri merupakan suatu konsep atau paradigma baru dalam mendesain, mengelola dan mengimplementasikan jaringan dengan memisahkan sistem kontrol (control plane) dan sistem forwarding (forwarding plane) kemudian mengisolasi komponen atau sub sistem dengan mendefinisikan interface yang standard. Sekarang ini, perlu adanya analisis untuk melihat seberapa baiknya performa dari sebuah kontroler tersebut mengingat pentingnya fungsi dari kontroler itu sendiri. Pada uji latency dan throughput ini, akan diperoleh informasi mengenai kemampuan dari kontroler tersebut. Hasil dari uji latency dan throughput pada kontroler Floodlight dan ONOS telah menunjukkan hasil bahwasanya kontroler memberikan nilai latency yang baik pada jumlah switch dibawah 60 switch. Nilai tertinggi respon yang diberikan oleh Floodlight adalah 2780.04 respon/detik pada jumlah 20 switch sedangkan ONOS memberikan nilai latency tertinggi yaitu 5931 respon/detik pada jumlah 30 switch. Untuk nilai throughput Floodlight sendiri memberikan nilai sebesar 1500,38 flow/detik pada jumlah host 450, sedangkan ONOS sendiri memberikan nilai yang lebih rendah yaitu 354,05 flow/detik pada jumlah host 450.

Kata Kunci: floodlight, kontroler, ONOS, Software-Defined Network

ABSTRACT

An important component of Software-Defined Network is a controller that directly controls the path data of the device. Software-Defined Network or what we often call the term SDN itself is a new concept or paradigm in designing, managing and implementing the network by separating the control plane and forwarding plane and then isolating the components or sub systems by defining the standard interface. Nowadays, it needs analysis to see how well the performance of a controller is given the importance of the function of the controller itself. In this latency and throughput test, information about the ability of the controller will be obtained. The results of the latency and throughput tests on the Floodlight and ONOS controllers have shown results that the controller provides a good latency value on the number of switches below 60 switches. The highest value of response given by Floodlight is 2780.04 response / sec on the number of 20 switches while ONOS gives the highest value of latency that is 5931 response / second on 30 switches. For Floodlight throughput value alone gives a value of 1500.38 flow / second on the host number 450, while ONOS itself gives a lower value of 354.05 flow / second on the host number 450.

Keywords: controller, floodlight, ONOS, Software-Defined Network

I. PENDAHULUAN

Jaringan internet adalah salah satu dari teknologi yang saat ini sangat dibutuhkan oleh masyarakat luas. Kebutuhan akan jaringan internet ini membuat teknologi yang satu ini terus berkembang dan mengalami perubahan yang sangat pesat. Jaringan internet ini biasanya dibangun dari berbagai perangkat jaringan seperti router, switch dan perangkat jaringan lainnya.

Software-Defined Network atau yang sering kita sebut dengan istilah SDN merupakan suatu konsep atau paradigma baru dalam mendesain, mengelola dan mengimplementasikan jaringan dengan memisahkan sistem kontrol (control plane) dan sistem forwarding (forwarding plane) kemudian mengisolasi komponen atau sub sistem dengan mendefinisikan interface yang standard [1]. Komponen utama dari *Software-Defined Network* adalah kontroler yang secara langsung melakukan kontrol terhadap data path dari perangkat. Disini, kontroler pada dasarnya memusatkan kecerdasan jaringan, sedangkan jaringan mempertahankan data plane forwarding yang didistribusikan melalui Switch OpenFlow. Untuk alasan ini kontroler menyediakan antarmuka untuk mengelola, mengendalikan dan Administrasi tabel flow yang Switch ini [2].

Kontroler memiliki dua perilaku yaitu reaktif dan proaktif. Dalam pendekatan reaktif, paket pertama dari aliran yang diterima oleh switch memicu kontroler untuk menyisipkan flow di setiap jaringan switch OpenFlow. Pendekatan ini relatif paling efisien penggunaan memori tabel flow yang ada, tapi setiap flow baru menyebabkan waktu setup tambahan kecil. Dalam pendekatan Proaktif, kontroler melakukan prapopulasi tabel flow di setiap switch. Pendekatan ini tidak memiliki tambahan waktu setup aliran karena aturan forwarding didefinisikan terlebih dahulu.

Jika switch kehilangan koneksi dengan kontroler, tidak mengganggu lalu lintas. Namun, operasi jaringan memerlukan manajemen rumit, misalnya, membutuhkan agregat (wildcard) aturan untuk menutup semua rute [3].

Menurut Metzler, A, kriteria kita dalam pemilihan kontroler ada sepuluh hal yang menjadi perhatian (1) Support kontroler terhadap protokol OpenFlow, (2) Virtualisasi jaringan yang dapat di support, (3) Fungsi dari sistem jaringan, (4) Scalability, (5) Performa dari kontroler, (6) Kemampuan untuk memprogram jaringan, (7) Keandalan, (8) Keamanan dari jaringan, (9) Pemantauan dan Visualisasi Terpusat, dan (10) Vendor yang membuat kontroler [4].

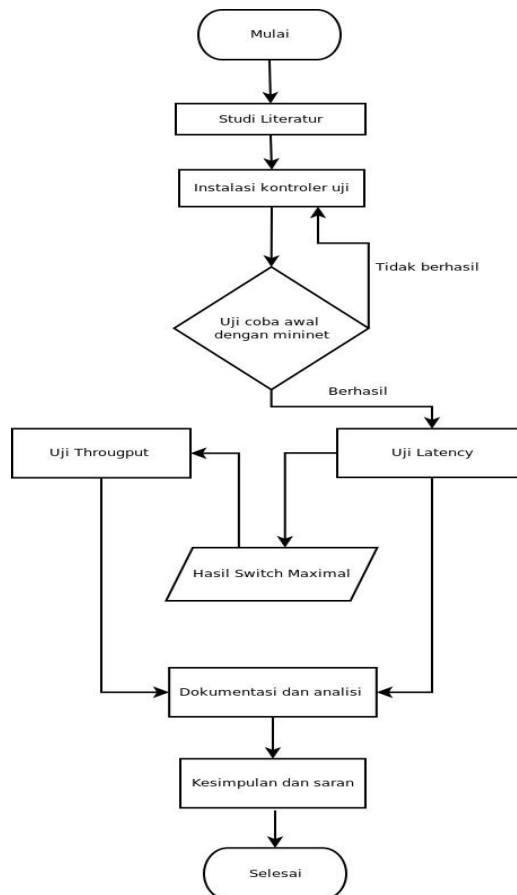
Salah satu dari hal penting dalam pemilihan sebuah kontroler adalah performa dari kontroler itu sendiri. Sebuah kontroler haruslah bisa membuah sebuah *flow* tabel sampai pada tingkat yang harus memiliki kemampuan untuk memproses dan mampu memastikan bahwasannya kontroler bukanlah sebuah hambatan. Disini hal yang dapat mempengaruhi sebuah performa tersebut merupakan waktu dan juga jumlah *flow* per detik.

Kontroler yang telah berkembang diantaranya *OpenDayLight* (ODL), POX, NOX, Beacon, Ryu, ONOS, Floodlight, Maestro dan masih ada beberapa lainnya yang terus dikembangkan. Floodlight disini merupakan salah satu kontroler enterprise yang penerapannya menggunakan bahasa pemrograman Java, terbuka berlisensi Apache dan dikembangkan oleh komunitas pengembang di Big Switch Network [5]. Sedangkan ONOS (*Open Network Operating System*) merupakan sistem operasi jaringan SDN yang terbuka (*open source*), ditujukan terutama untuk jaringan operator [1].

Mengingat pentingnya fungsi dari sebuah kontroler dalam arsitektur jaringan. Perlu adanya analisis untuk melihat seberapa baiknya dari performa kontroler tersebut. Dalam artikel ini, uji yang dilakukan adalah untuk mengetahui tingkat *throughput* dan *latency* yang ada pada kontroler terhadap penanganan sejumlah switch sehingga diperoleh informasi mengenai kemampuan dari kontroler yang akan digunakan.

II. PANDUAN UNTUK MEMPERSIAPKAN ARTIKEL

Metode yang digunakan dalam pengujian performa kontroller *Software-Define Network* dengan *floodlight* dan ONOS yang dikerjakan dalam penelitian ini dapat dilihat pada alur penelitian pada gambar 1 dengan penjelasan sebagai berikut :



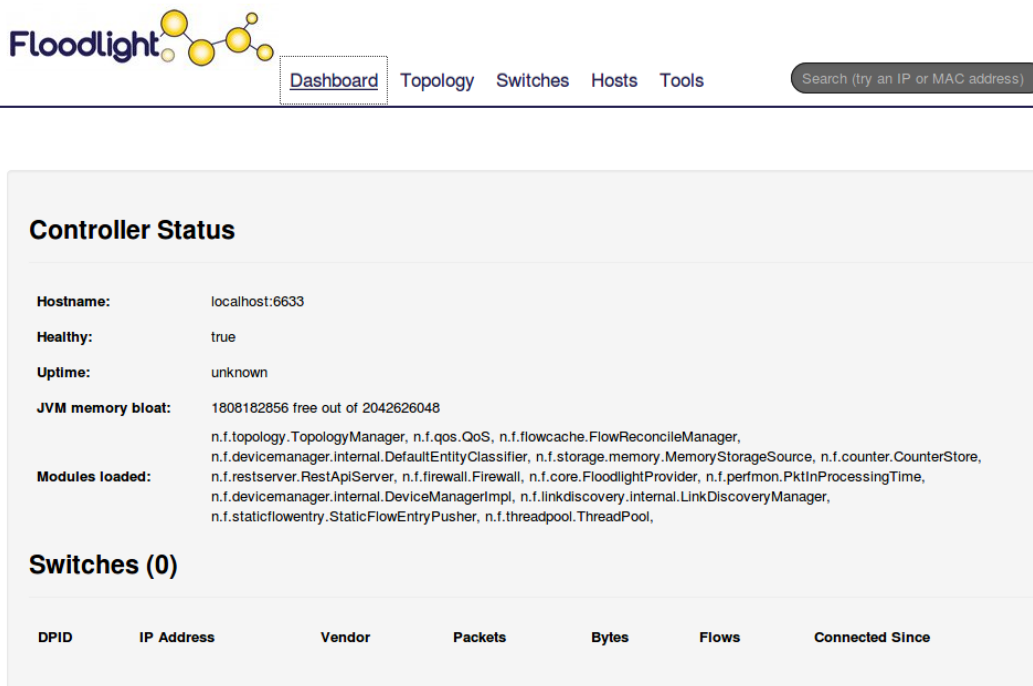
Gambar. 1. Alur penelitian

A. Instalasi kontroler yang akan diujikan.

1) Instalasi Kontroler Floodlight

Gambaran langkah-langkah instalasi dari kontroler Floodlight adalah sebagai berikut :

- a. Instalasi dependensi yang dibutuhkan oleh Floodlight dengan perintah :
~\$sudo apt-get install build-essential default-jdk ant python-dev eclipse
- b. Melakukan cloning file Floodlight dari web github dengan perintah :
~\$git clone git://github.com/floodlight/floodlight.git
- c. Setelah semua terinstal, Floodlight dapat dijalankan pada folder Floodlight dengan perintah :
~\$cd floodlight/target
~/floodlight/target\$ sudo java -jar floodlight.jar
- d. Setelah semua dijalankan, maka kontroler Floodlight GUI dapat diakses pada link <http://localhost:8080/ui/index.html> dengan tampilan seperti pada gambar 2.

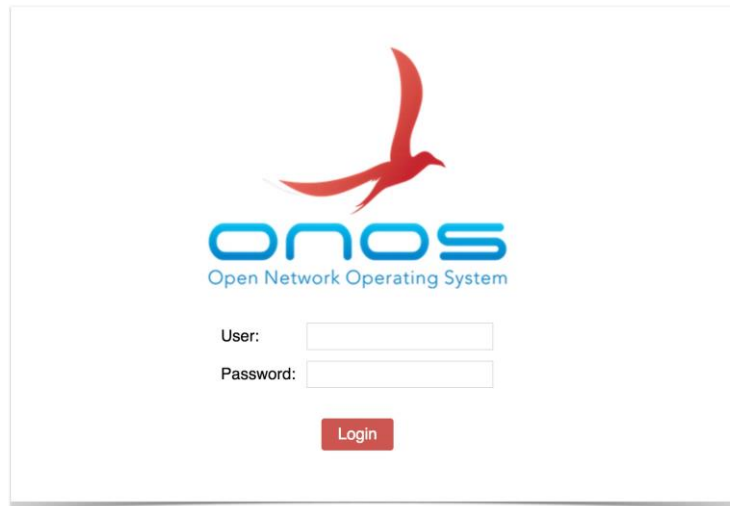


Gambar. 2. GUI Floodlight

2) Instalasi Kontroler ONOS

Gambaran langkah-langkah instalasi dari controller ONOS adalah sebagai berikut :

- a. Buat direktori opt dengan perintah :
~\$ sudo mkdir /opt
~\$ cd /opt
- b. Melakukan cloning file ONOS dari web github dengan perintah :
~\$ cd wget -c http://repo1.maven.org/maven2/org/onosproject/onos-releases/1.13.1/onos-1.13.1.tar.gz
- c. Kemudian ekstrak file ke folder opt
~\$ sudo tar xzf onos-\$ONOS_VERSION.tar.gz
- e. Kemudian compile dengan perintah
~\$ /opt/onos/bin/onos-service start
- f. Setelah itu, maka kontroler ONOS GUI dapat diakses pada link <http://ip-onos:8181/onos/ui/index.html> dengan username & password defaultnya (karaf:karaf) atatu (onos : rocks) seperti pada gambar 3.



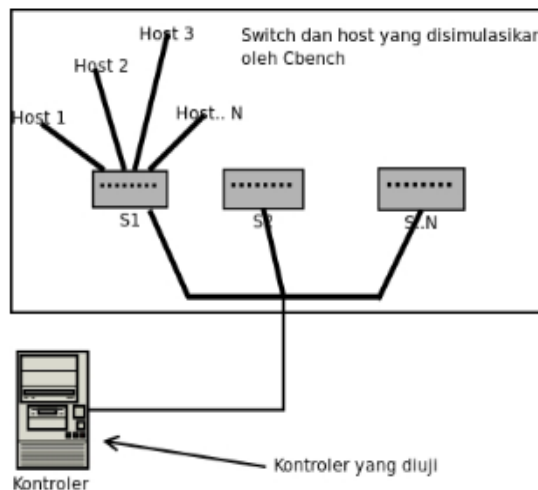
Gambar. 3. GUI ONOS

B. Menguji Kontroler yang Telah Terinstall

Uji kesiapan kontroler *Floodlight* dan ONOS, menguji keberhasilan langkah pertama dengan emulator mininet yang diinstall dan dijalankan pada VirtualBox, dengan skenario topologi single, 5 host. Bila kontroler tidak dapat mengenal switch pada mininet, maka proses instalasi akan di ulang.

Pengujian dilakukan dengan skenario meletakkan kontroler dan cbench diletakkan pada virtualbox dalam 1 host/PC dan dijalankan pada sumber daya yang sama. Spesifikasi PC/Laptop yang digunakan dalam uji ini adalah sebagai berikut :

- 1) CPU : Intel® Core™ i3-4005U CPU @ 1.70GHz × 4
- 2) RAM : SODIMM DDR3 4Gb 1333MHz
- 3) Operation System : Linux UBUNTU 17.04 LTS kernel 4.15.0-24-generic.



Gambar. 4. Skenario Uji

Pada pengujian digunakan seluruh threads CPU yang ada, tidak dilakukan perubahan/variasi pada threads.

C. Uji Latency

Pengujian latency dilakukan dengan perintah sebagai berikut : `~/oflops/cbench$ cbench -c [ip-kontroler] -p 6633 -l 5 -m 5000 -D 5 -M 5 -s [jumlah switch]`

D. Uji Throughput

Pengujian *throughput* dilakukan dengan perintah sebagai berikut : `~/oflops/cbench$ cbench -c [ip-kontroler] -p 6633 -l 5 -m 5000 -D 5 -M [jumlah MAC] -s [jumlah switch] -t`

Pada uji *throughput* dan *latency* diberikan beban dengan panjang pengujian 5000 ms dan diberikan jeda mulai pengujian setelah *features_reply* diterima selama 5 ms.

III. HASIL UJI LATENCY DAN THROUGHPUT

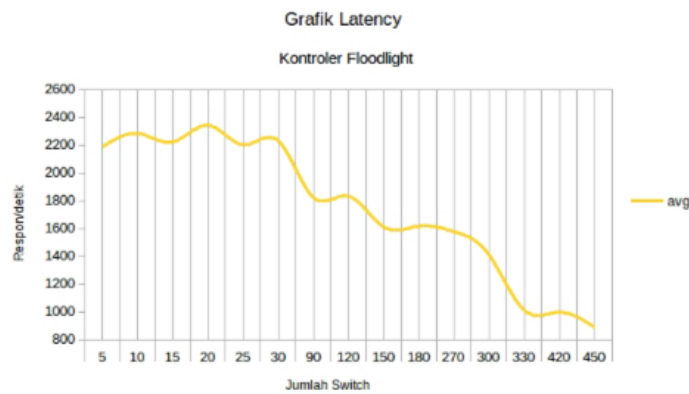
A. Pada Kontroler Floodlight

Pada penelitian ini, sampel data yang digunakan untuk kontroler *floodlight* dapat dilihat pada tabel 1 dimana pengujian dilakukan pada jumlah switch 5 hingga 450 switch.

TABEL I
SAMPEL DATA UJI *LATENCY* PADA KONTROLER *FLOODLIGHT*

n Switch	20	90	120	330	450
Min	2065.22	1615.83	1661.75	246.97	195.25
Max	2780.04	2028.07	2003.42	1505.84	1239.06
Avg	2342.25	1821.06	1831.35	1010.41	887.08

Dari data yang diperoleh, grafik hasil pengujian *latency* pada kontroler *floodlight* terlihat seperti gambar 5 dibawah ini :



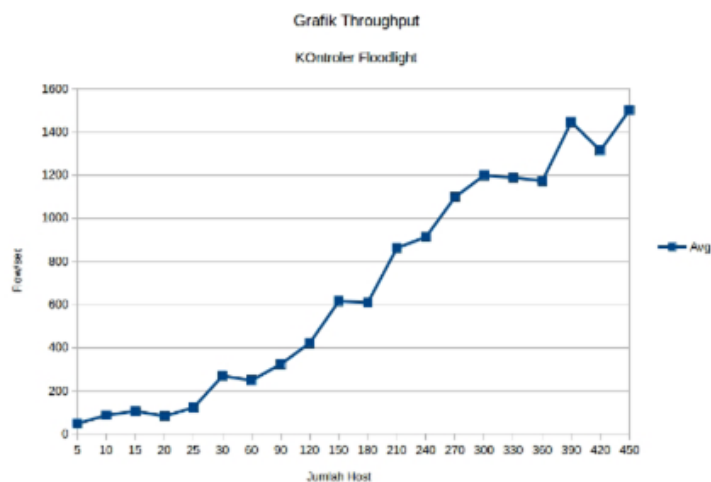
Gambar. 5. Hasil Pengujian *Latency Floodlight*

Dari data yang terlihat pada tabel 1 dan gambar 5 diatas, jika dilihat dari data rata-rata (avg) respon/detik vs switch, dapat disimpulkan bahwasannya semakin banyak switch yang terlihat, maka semakin rendah pula respon yang diberikan oleh kontroler dalam menangani *flow*. Lambatnya kontroler *floodlight* dimulai pada switch berjumlah lebih dari 60, sehingga jumlah dari switch optimum untuk kontroler *floodlight* adalah antara 1 sampai 60 switch [6]. Sedangkan pada pengujian *throughput* digunakan switch sebanyak 20 switch.

TABEL II
SAMPEL DATA UJI *THROUGHPUT* PADA KONTROLER *FLOODLIGHT*

n Host	20	90	120	330	450
Min	79.06	322.66	419.88	1185.06	1498.26
Max	84.55	326.11	433.04	1190.84	1503.06
Avg	80.62	323.11	421.57	1187.79	1500.38

Dari data yang diperoleh, grafik hasil pengujian *throughput* pada kontroler *floodlight* terlihat seperti gambar 6 dibawah ini :



Gambar. 6. Hasil Pengujian *Latency Floodlight*

Dari data yang terlihat pada tabel 2 dan gambar 6 diatas, apabila kita melihat dari data rata-rata (avg) *flow*/detik vs jumlah dari host yang ditampilkan pada tabel 2, kontroler *floodlight* dapat memberikan kemampuan yang baik dikarenakan mengingat jumlah host yang ada, maka semakin tinggi jumlah *flow*/detik yang dapat dilayani oleh kontroler [6].

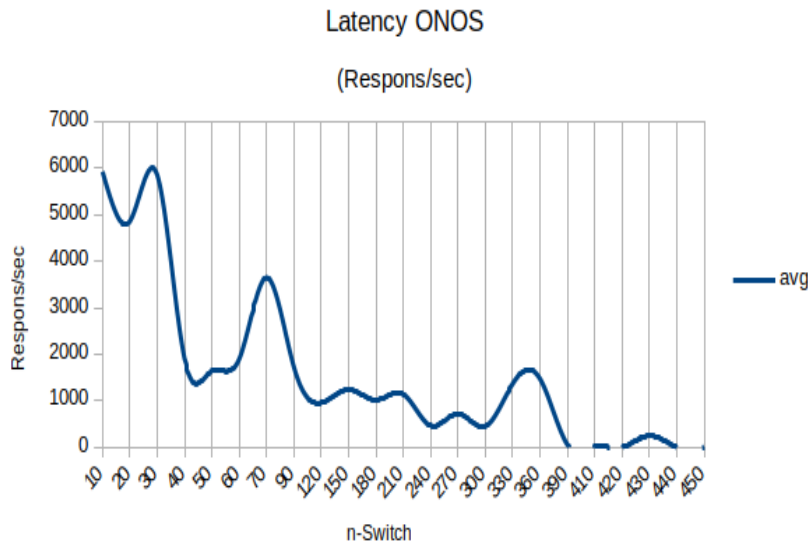
B. Pada Kontroler ONOS

Pada kontroler *ONOS* dilakukan pengujian yang sama dengan kontroler *floodlight*. Pengujian melibatkan switch dan host dengan jumlah yang sama. Sampel data pada uji *latency* kontroler *ONOS* dapat dilihat pada tabel 3 dibawah ini.

TABEL III
SAMPel DATA Uji LATENCY PADA KONTROLER ONOS

n Switch	30	70	210	360	450
Min	294.00	1493.97	0.00	0.00	0
Max	12543.25	5465.99	4543.79	4381.88	30
Avg	5859.89	3649.59	1135.95	1468.59	7.5

Dari data yang diperoleh, grafik hasil pengujian *latency* pada kontroler *ONOS* terlihat seperti gambar 7 dibawah ini :



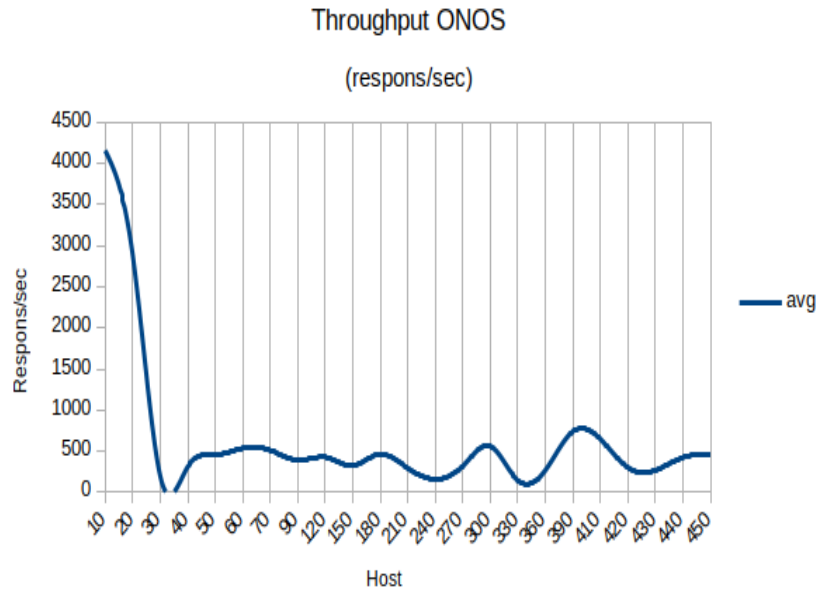
Gambar. 7. Hasil Pengujian Latency ONOS

Dilihat dari tabel 3 dan gambar 7 diatas, rata-rata (avg) respon/detik terlihat bahwa semakin banyak switch yang terlibat, maka akan semakin rendah pula respon yang akan diberikan oleh kontroler dalam menghadapi *flow*. Kontroler *ONOS* mulai mengalami keterlambatan pada switch dengan jumlah diatas 70, sehingga jumlah switch optimum untuk kontroler *ONOS* adalah antara 1 switch sampai 70 switch. Jumlah switch yang digunakan pada pengujian *throughput* sebanyak 30 switch karena memiliki respon sedikit.

TABEL IV
SAMPel DATA Uji THROUGHPUT PADA KONTROLER ONOS

n Host	30	70	210	360	450
Min	153.85	323.72	249.91	244.12	195.45
Max	246.94	708.97	380.96	323.94	851.59
Avg	200.16	511.42	293.41	267.32	466.09

Dari data yang diperoleh, grafik hasil pengujian *throughput* pada kontroler *ONOS* terlihat seperti gambar 8 dibawah ini :



Gambar. 8. Hasil Pengujian Latency ONOS

Dilihat dari tabel 4 dan gambar 8 diatas, rata-rata (avg) respon/detik vs jumlah host yang ditampilkan, kontroler ONOS memberikan kemampuan respon yang baik, sejalan dengan meningkatnya jumlah host yang ada. Respon yang dihasilkan relative stabil pada jumlah host 40/switch.

IV. KESIMPULAN

Dari hasil pengujian kontroler *floodlight* dan *ONOS* ini, dapat ditarik kesimpulan bahwasannya kedua kontroler memberikan nilai *latency* yang baik pada jumlah switch dibawah 60 switch. Nilai tertinggi respon yang diberikan oleh kontroler *floodlight* adalah 2780.04 respon/detik pada jumlah switch sebanyak 20 switch. Sedangkan pada *ONOS* memberikan nilai *latency* tertinggi yaitu 5931 respon/detik pada jumlah 30 switch. Untuk nilai *throughput*, kontroler *floodlight* memberikan nilai sebesar 1500.38 *flow*/detik pada jumlah host 450 dan kontroler *ONOS* memberikan nilai yang rendah yaitu 354.05 *flow*/detik pada jumlah host 450. Dari paparan tersebut dapat disimpulkan bahwasannya performa dari kontroler *floodlight* lebih baik dibandingkan dengan kontroler *ONOS* untuk jumlah switch yang besar. *Floodlight* sendiri mampu memberikan *throughput* yang lebih besar dibandingkan dengan kontroler *ONOS*, sedang performa *latency* yang disajikan dalam jumlah respon/detik, kontroler *floodlight* memberikan respon yang lebih baik dibandingkan dengan kontroler *ONOS*.

DAFTAR PUSTAKA

- [1] Mulyana, Eueng (2015). Buku Komunitas SDN-RG. Bandung : Gitbook
- [2] G. R. D. T. Muntaner (Oktober 2012). Evaluation of OpenFlow Controllers. hal. 90
- [3] Marcial P. Fernandez (2013). Evaluating OpenFlow Controller Paradigms. ICN 2013 : The Twelfth International Conference on Networks. hal.151-157
- [4] A. Metzler dan J. Metzler (2015). Ten Things to Look for in an SDN Controller.
- [5] Anggara, S., M., (2015). Pengujian Performa Kontroler SDN: POX dan Floodlight. Institut Teknologi Bandung : Bandung
- [6] Rikie Kartadie dan Barka Satya (Februari 2015). Uji Performa Kontroler Floodlight Dan Opendaylight Sebagai Komponen Utama Arsitektur Software-Defined Network. STIMIK AMIKOM Yogyakarta : Yogyakarta