E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121

FLOODLIGHT VS ONOS DALAM UNJUK KERJA

Rikie Kartadie[1)], Vertika Panggayuh[2)]
[1, 2)]Pendidikan Teknologi Informasi
STKIP PGRI Tulungagung, Jawa Timur, Indonesia
e-mail: rikie@stkippgritulungagung.ac.id[1)], vertika@stkippgritulungagung.ac.id[2)]

ABSTRAK

Kontroler yang secara langsung melakukan kontrol terhadap data path dari perangkat adalah komponen utama dari Software-Defined Network (SDN). Kontroler bertanggung jawab untuk menentukan bagaimana menangani paket dan mengelola flow table (tabel flow) dengan menambahkan dan menghapus isi flow table melalui secure channel. Kontroler pada dasarnya memusatkan dan menjadi pusat kecerdasan jaringan, sedangkan jaringan mempertahankan data plane forwarding yang didistribusikan melalui Switch OpenFlow. Untuk alasan ini kontroler menyediakan antarmuka untuk mengelola, mengendalikan dan administrasi flow table Switch ini. Pengujian dilakukan untuk mengetahui tingkat throughput dan latency dari kontroler dan diuji menggunakan cbance tool yang akan menguji pada protokol TCP dan UDP. Pengujian dilaksanakan dengan memaksa kontroler pada titik maksimal kemampuannya tanpa melakukan setingan tambahan (Setingan standar), sehingga diperoleh informasi yang tepat kemampuan dari kontroler yang akan digunakan. Karena kebutuhan ini, perlu di uji unjuk kerja dari kontroler yang digunakan. pada penelitian ini pengujian dilakukan pada kontroler floodlight dan ONOS. Dari hasil pengujian dapat dilihat bahwa kontroler floodlight lebih stabil dalam mengatasi beban switch dan host lebih baik dibandingkan dengan kontroler ONOS.

***Kata Kunci**: Floodlight, Kontroler, ONOS, SDN, Unjukkerja*

ABSTRACT

The controller that directly controls the data path of the device is a major component of the Software-Defined Network (SDN). The controller is responsible for determining how to handle packages and manage flow tables by adding and removing flow table contents through secure channels. The controller focuses and becomes the center of network intelligence, while the network maintains the plane forwarding data that is distributed through the OpenFlow Switch. For this reason, the controller provides an interface for managing, controlling and administering this Switch flow table. Testing is done to determine the level of throughput and latency of the controller and tested using the cbance tool that will test the TCP and UDP protocols. The test is carried out by forcing the controller to the maximum point of its ability without making additional settings (standard settings) so that the correct information about the capabilities of the controller will be used. Because of this need, it is necessary to test the performance of the controller used. in this study, the testing was done on the floodlight controller and ONOS. From the test results, it can be seen that the floodlight controller is more stable in dealing with switch and host loads better than the ONOS controller.

***Keywords**: Floodlight, Kontroler, ONOS, SDN, Performance*

I. PENDAHULUAN

Jaringan saat ini adalah hasil dari protokol dan keputusan desain jaringan yang dibuat pada 1970-an. Setelah didirikan, topologi jaringan tidak diharapkan untuk berubah banyak, jika mungkin tidak berubah sama sekali. Namun kenyataannya, kebutuhan akan jaringan terus berkembang dan perancangan jaringan terus mengalami perubahan yang draktis. Pertumbuhan vendor pengembang perangkat jaringan pun terus bertambah.

Jaringan biasanya dibangun dari sejumlah besar perangkat jaringan seperti router, switch dan perangkat lainnya. Setiap perangkat menjalankan manipulasi penerusan paket data, dengan protokol yang komplek yang tertanam didalam perangkat tersebut. Operator jaringan bertanggungjawab secara langsung untuk melakukan konfigurasi, aturan, dan tidak jarang hingga ke aplikasi yang digunakan didalam jaringan. Operator biasanya melakukan konfigurasi secara manual pada setiap perangkat yang terhubung, hal ini memberikan celah pada kesalahan

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121

konfigurasi karna kesalahan manusia (*human error*) terutama bila harus menangani jumlah perangkat yang banyak.

Kontroler yang secara langsung melakukan kontrol terhadap *datapath* dari perangkat adalah komponen utama dari Software-Defined Network (SDN). Kontroler bertanggung jawab untuk menentukan bagaimana menangani paket dan mengelola tabel flow dengan menambahkan dan menghapus isi tabel flow melalui *secure channel*. Kontroler pada dasarnya memusatkan kecerdasan jaringan, sedangkan jaringan mempertahankan *dataplane forwarding* yang didistribusikan melalui switch OpenFlow. Untuk alasan ini kontroler menyediakan antarmuka untuk mengelola, mengendalikan dan Administrasi tabel flow pada Switch [1].

Unjuk kerja kontroler adalah dalah satu dari sepuluh kreteria pemilihan kontroler [2], kontroler SDN harus dapat membuat flow tabel sebelumnya sampai tingkat yang mungkin dan harus memiliki kemampuan pemprosesan dan I/O yang mampu memastikan bahwa kontroler bukanlah hambatan dalam pembentukan flow. Dengan demikian, dua dari kunci metrik kinerja yang terkait dengan kontoler SDN adalah waktu setup flow dan jumlah flow per detik. Metrik kinerja ini sangat mempengaruhi ketika pengontrol SDN tambahan harus dikerahkan. Misalnya, jika switch dalam arsitektur SDN memulai memberikan flow yang lebih besar daripada yang dapat didukung oleh kontroler SDN yang ada, sehingga lebih dari satu kontroler harus diimplementasikan. Waktu setup flow dan jumlah flow per detik berkaitan erat dengan *latency* dan *througput* yang dihasilkan oleh kontroler.

Kontroler yang berkembang beberapa diantaranya adalah OpenDayLight (ODL), POX, NOX, Beacon, Ryu, ONOS, Floodlight, Maestro dan masih ada beberapa yang lain yang terus muncul dan dikembangkan. Pengembang kontroler mengembangkannya dengan menggunakan platform yang beragam, dengan penggunaan *platform* yang beragam ini menjadikan tiap kontroler memiliki unjukkerja yang berbeda satu dengan yang lainnya.

Tujuan Penelitian

Floodlight dan ONOS merupakan kontroler yang banyak digunakan, sehingga dibutuhkan informasi yang tepat terhadap kemampuan dalam hal ini unjuk kerja dari kontroler tersebut terutama bila kontorler akan digunakan untuk implementasi maupun untuk pengujian lainnya terlebih pada penggunaan switch dan besar flow yang akan terlibat dan sebagai dasar pemilihan kontroler.

Tujuan khusus

Fungsi dari kontroler dalam arsitektur jaringan sangatlah vital sehingga unjukkerja dari kontroler dirasa perlu untuk diuji seberapa baik unjukkerja dari kontroler. Pengujian dilakukan untuk mengetahui tingkat throughput dan latency dari kontroler terutama kontroler floodlight dan ONOS yang memiliki Platform yang sama yaitu java dan dari pengembang yang sama yaitu ONF.

II. TINJAUAN PUSTAKA

Muntaner, GR. De Tejdana, melakukan pengujian kontroler NOX, Maestro, Beacon, dan Trema. Keempat kontoler tersebut merupakan kontroler yang tidak menggunakan GUI dalam melakukan *input table flow*, sedangkan penelitian ini menguji kontroler yang mengunakan GUI dalam melakukan *input table flow*[1]. Turull,D. Dkk, memfokuskan pada aplikasi virtualisasi jaringan, dan mengukur delay yang terjadi pada pesan ICMP, waktu transfer dari koneksi TCP, dan kerugian paket dalam lalu lintas UDP, berbagai penundaan antara switch dan kontroler[3]. Penelitian ini memfokuskan pada throughput dan latency dengan mensimulasi jumlah flow per detiknya dan mensimulasi jumlah switch yang terlibat pada kontroler floodlight dan ONOS.

Rowshanrad, S., et al., melakukan pengujian performa dari floodlight dan Opendayligh dengan mengukur performa kontroler bila dijalankan pada 3 topologi standar mininet yaitu single, linear, dan tree dengan jumlah switch dan *host* yang terlibat maksimal 8 (delapan*) switch* [3]. Sedangkan yang akan dilakukan pada penelitian ini adalah melibatkan sebanyak-banyaknya switch dan *host* dan mengukur kemampuan kontroler dalam menjalankan

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121

atau melakukan kontrol ke sejumlah perangkat tersebut dan yang diuji adalah kontroler floodlight dan ONOS.

Duque, et al., pun menguji performa dari kontroler floodlight dam opendaylight, penelitian Duque menekankan pada performa kedua kontroler tersebut dalam menangani load balancing dengan tidak memperhatikan banyaknya perangkat yang terlibat [4]. Penelitian ini tidak memfokuskan pada load balancing namun memfokuskan pada nilai trhoughput dan latency yang ditampilkan oleh kedua kontroler floodlight dan ONOS dan banyaknya perangkat yang terlibat.

Penelitian dengan pendekatan hampir sama adalah penelitian yang dilakukan oleh Saleh Asadollahi dan Bhargavi Goswami [5], dimana melakukan pengujian throughput dan latency, namun hanya menguji satu kontroler saja yaitu floodlight, sedangkan peneliti melakukan pengujian pada kontroler floodlight dan ONOS dan membandingkan performa dari kontroler tersebut mana yang lebih baik.

Sejalan dengan penelitian yang dilakukan Zhu [6], throughput dan latency adalah kompunen yang digunakan dalam mengukur kinerja dari kontroler. Tools yang dapat digunakan dalam mengukur throughput dan latency menurut Zhu diantaranya adalah cbench, PktBlaster, dan OFnet yang tentu saja memiliki keunggulan masing-masing. Pengujian performa sebenarnya dapat diujikan dengan menggunakan model antrian seperti yang dilakukan oleh Bing Xiong, et.al., [7] penelitian yang akan dilakukan mengadopsi dari penelitian ini namun langsung membandingkan dua kontroler.

Penelitan lain tentang pengujian kontroler diantaranya adalah pengujian performa kontroler floodlight dengan POX [8] dengan pengujian menggunakan topologi yang ada pada mininet. Pada penelitan ini tidak menggunakan topologi tertentu namun langsung memberikan beban jumlah perangkat kepada kontroler. Topologi tidak menjadi acuan utama pada penelitian ini.

Dari beberapa penelitian yang telah dilakukan, sebagian besar memfokuskan kepada pengujian performa atau unjuk kerja dari kontroler floodlight dan dibandingkan dengan kontroler lain dari vendor yang berbeda, bahkan ada sebagian yang membandingkan dengan kontroler yang memiliki flatform yang berbeda. Dalam penelitian ini akan dilakukan pengujian unjuk kerja kontroler floodlight dan ONOS yang merupakan kontroler dari vendor yang sama, dengan flatform yang sama namun dengan fitur, GUI dan Ux yang berbeda, sehingga pengguna kontroler dapat memilih kontroler dari vendor ONF ini yang meneurut pengguna paling tepat dan sesuai dengan kebutuhannya. Dilain sisi penelitian ini pun menguji kemampuan dari kontroler dalam menangani jumlah switch yang terlibat dan kemampuan kontroler dalam menangni flow.

Uji yang akan dilakukan adalah dengan menguji throughput dan latency dengan menggunakan tool uji *cbench* sesuai dengan rekomendasi yang diberikan oleh peneliti sebelumnya. Komparasi fitur antara floodlight dan ONOS yang diperoleh dari Mamushiane[9] dapat dilihat pada tabel I .

A. Metode Penelitian

Dalam penelitian ini, metode penelitian yang akan dilakukan adalah sebagai berikut: (1)Studi literatur, mengumpulkan bahan atau materi penelitian yang berupa literatur/referensi yang berhubungan dengan penelitian yang akan dilaksanakan, (2)Instalasi kontroler yang akan diujikan, (3)Uji kesiapan kontroler Floodlight dan ONOS, untuk menguji menguji keberhasilan langkah instalasi dengan emulator mininet yang diinstall dan dijalankan pada VirtualBox, dengan skenario topologi 2 switch, 5 host. Bila kontroler tidak dapat mengenal switch pada mininet, maka proses instalasi akan di ulang, (4)Melakukan pengujian *latency* Floodlight dan ONOS dengan jumlah switch mulai 2 hingga batas switch tidak dapat di kelola lagi oleh kontroler dan jumlah *host* setiap switch adalah 5 host, (5)Melakukan pengujian throughput Floodlight dan ONOS dengan jumlah switch tertinggi yang masih dapat dikelola oleh kontroler dari pengujian *latency*, (6)Dokumentasi dan melakukan analisis dari hasil dan menarik kesimpulan. Diagram Alir Penelitian dapat dilihat pada gambar 1 dibawah ini:

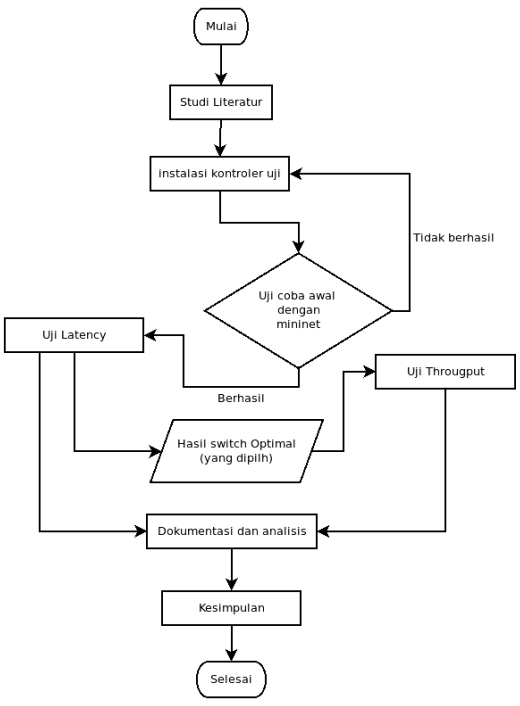
E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121

TABEL I

PERBANDINGAN FITUR FLOODLIGHT VS ONOS (SUMBER DIMODIFIKASI: MAUSHINE [9])

	Floodlight	ONOS
Antarmuka Southbound	OF 1.0, 1.1, 1.2, 1.3, 1.4, 1.5	OF 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, NETCONF
REST API	Ya	Ya
GUI	Web / Java-Based	Web- Based
Modularity	Menengah	Tinggi
Dukungan Orchestrator	Ya	Ya
OS yang medukung	Linux, Widows dan Mac	Linux, Widows dan Mac
Perusahaan partner	Big Switch Network	ON.LAB, SK Telkom, Cisco, Ericsson, Fujitsu, Huawei, Intel, AT&T, NEC, Ciena, Nsft, Ntt Communication
Dokumentasi	Baik	Baik
Bahasa Pemprograman	Java	Java
Dukungan Multi-threading	Ya	Ya
Dukungan TLS	Ya	Ya
Virtualisasi	Mininet dan OvS	Mininet dan OvS
Aplicatin Domain	Kampus	Data Center dan Transport-SDN/WAN
Terdistribusi/Terpusat	Terpusat	Terdistribusi



Gambar 1. Diagram alir penelitian dalam pengujian unjuk kerja kontroler

E-ISSN : 2540 - 8984

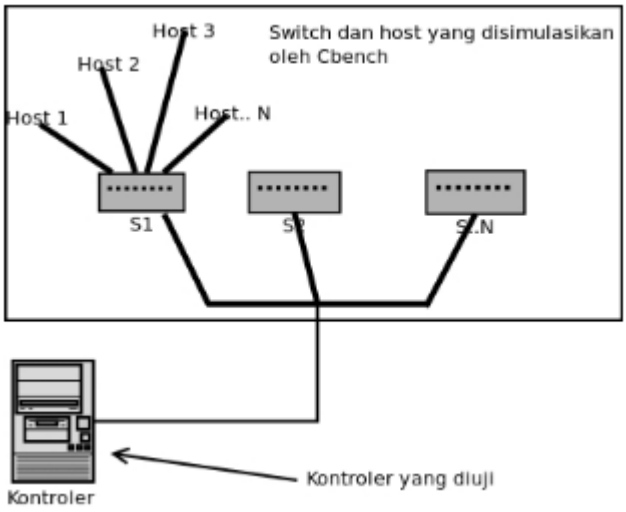
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121

Pengujian unjuk kerja kontroler dilakukan dengan skernario meletakkan kontroler dan *cbench* pada virtualbox dalam 1 host/PC dan dijalankan pada sumber daya yang sama. Skema pengujian dapat dilahat pada gambar 2 dibawah ini. Spesifikasi PC/Laptop yang digunakan dalam uji ini adalah sebagai berikut;

- CPU : Intel® CoreTM i3-4005U CPU @ 1.70GHz × 4 threat

- RAM : SODIMM DDR3 4Gb 1333MHz

- *Operation System* : Linux UBUNTU 18.04 LTS kernel 4.15.0-47-generic



Gambar 2. Skenario uji yang dilakukan

Seperti terlihat pada gambar 2 skenario uji di atas, pengujian dilakukan dengan menggunakan *cbench* dengan melibatkan jumlah switch dan *host* tertentu (sesuai dengan diagram alir penelitian yang direncanakan) untuk mendapatkan unjuk kerja dari kontroler. Kontroler diletakkan pada *host* (OS Laptop) dari virtualbox yang dijalankan sedangkan emulasi switch dan *cbench* berada pada virtualbox.

Pada pengujian digunakan seluruh *threats* CPU yang ada, tidak dilakukan perubahan/variasi pada *threats* CPU. Tidak dilakukannya pengujian pada *threats* CPU karna diasumsikan CPU server dioptimalkan dalam pelayanan terhadap kontroler, dan kontroler yang diuji sesuai dengan tabel 1 sebelumnya telah mendukung multi-threats CPU.

Uji *Latency*

Uji *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]
```

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

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
```

Pengujian dilakukan pada range 5–450 host dengan jumlah switch optimal dengan kenaikan rentangan 10 host. Kedua cara diatas digunakan pada pengujian floodlight dan ONOS.

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121

B. Asumsi Penelitian Yang Digunakan

Dalam penelitian ini menggunakan beberapa asumsi dasar yang tidak merusak atau mempengaruhi hasil. Asumsi dasar ini didasarkan pengertian bahwa diagram topologi tidak hanya berupa jaringan fisik dan komponen jaringan virtual, tetapi juga dapat termasuk aplikasi yang dipergunakan, sedang dikembangkan atau dapat juga aplikasi memegang peranan sebagai *node* dan *links* [10].

III. HASIL PENELITIAN DAN PEMBAHASAN

A. Pengujian Kontroler Floodlight

Uji Latency

Pengujian *Latency* dilakukan pada rentang switch 10 hingga 450 switch dengan kenaikan rentangan 10 switch. Jumlah pengujian setiap jumlah sesi adalah 5x pengulangan. Contoh respon dari *cbench* pada pengujian seperti terlihat dibawah ini.

```
cbench: controller benchmarking tool
   running in mode 'latency'
   connecting to controller at 192.168.56.1:6633
   faking 10 switches offset 1 :: 5 tests each; 5000 ms per test
   with 5 unique source MACs per switch
   learning destination mac addresses before the test
   starting test with 5 ms delay after features_reply
   ignoring first 1 "warmup" and last 0 "cooldown" loops
   connection delay of 0ms per 1 switch(es)
   debugging info is off
01:33:14.386 10  switches: fmods/sec:  6366  3231  3227  3034  3191  5651  3132  3063  2414  2203   total
= 7.101693 per ms
01:33:19.488 10  switches: fmods/sec:  7004  3502  3500  3496  3492  6818  3370  3310  2652  2162   total
= 7.861158 per ms
01:33:24.590 10  switches: fmods/sec:  7010  3530  3531  3534  3527  6920  3443  3379  2618  2153   total
= 7.928978 per ms
01:33:29.694 10  switches: fmods/sec:  7062  3529  3546  3547  3529  6910  3418  3400  2682  2184   total
= 7.959643 per ms
01:33:34.797 10  switches: fmods/sec:  7108  3536  3546  3554  3547  6976  3467  3388  2702  2196   total
= 8.002956 per ms
RESULT: 10 switches 4 tests min/max/avg/stdev = 7861.16/8002.96/7938.18/51.66 responses/s
```

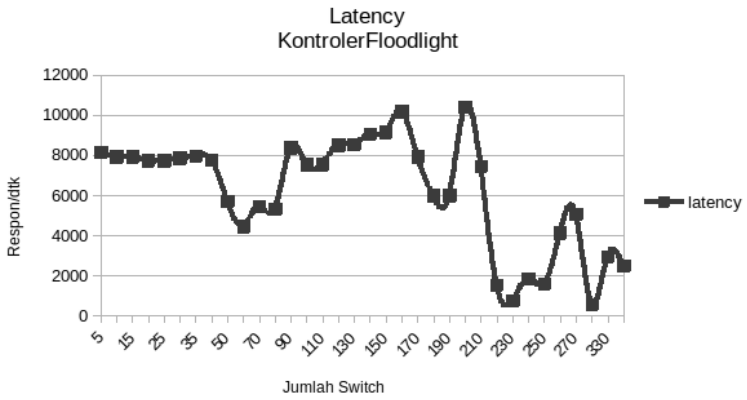
Sample data hasil uji *latency* pada kontroler Floodlight dapat dilihat pada sample data Tabel II dimana tabel ini memberikan contoh data yang diperoleh pada pengujian dengan rentang switch yang telah ditentukan. Nilai pengujian tersaji dalam bentuk grafik yang dapat dilihat pada gambar 3, pada grafik tersebut dapat dilihat perubahan respon tiap penambahan jumlah switch yang harus ditangani oleh kontroler floodlight.

TABEL II. CONTOH DATA LATENCY KONTROLER FLOODLIGHT (RESPON/DETIK)

	Jml Switch	10	90	120	330	450
Respon/s	Min	7861.16	7239.48	7184.93	1542.31	1965.52
	Max	8002.96	11291.14	12222.55	4034.36	2963.18
	Avg	7938.18	8383.59	8523.32	2951.4	2492.05

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121



Gambar 3. Hasil uji *latency* pada kontroler floodlight (Dalam Respon/Detik)

Dari data yang terlihat Tabel II dan gambar 3 diatas bila dilihat dari data rata-rata (avg) respon/detik vs switch, respon yang diberikan kontroler floodlight menunjukkan respon yang relatif stabil pada jumlah switch di bawah 220 switch. Kemampuan merespon flow pada kontroler floodlight mulai mengalami penurunan pada switch berjumlah diatas 220, sehingga jumlah switch yang masih mampu direspon dengan cendrung stabil untuk kontroler Floodlight adalah antara 1 switch hingga 220 switch. Kontroler memberikan nilai fluktuatif pada jumlah switch 50 hingga 220, sehingga jumlah switch yang digunakan pada uji selanjutnya adalah diantara 1 - 40 switch dan jumlah switch optimal menggunakan 40 switch.

Penurunan respon yang signifikan diberikan oleh floodlight pada jumlah switch diatas 220, dan kemudian kembali menuju tingkat kestabilan walaupun rendah, dapat diakibatkan karena *cbench* mempunyai algoritma menungg kecocokan flow_mod baru memberikan respon. Seperti dikutip dari github mininet, bahwa algoritma *latency* pada *cbench* adalah mensimulasikan *n* switch (*n* = 16 adalah *default*), kemudian buat *n* sesi openflow ke Kontroler. Untuk setiap sesi *cbench* memiliki langkah; 1)mengirim paket, 2)menunggu *flood_mod* yang cocok untuk kembali sebagai respon, 3)mengulangi langkah (1) dan (2), kemudian 4)menghitung berapa kali langkah (1)-(3) terjadi per detik [11], sehingga kemampuan kontroler floodlight merespon paket data hanya pada rentan switch 1-50 switch.

Uji Throughput

Pengujian *throughput* dilakukan pada rentang switch 10 hingga 450 *host* dengan kenaikan rentangan 10 *host*. Jumlah switch pengujian 20 switch, setiap sesi dilakukan pengulangan sebanyak 5x pengulangan. Sample data dapat dilihat pada Tabel III.

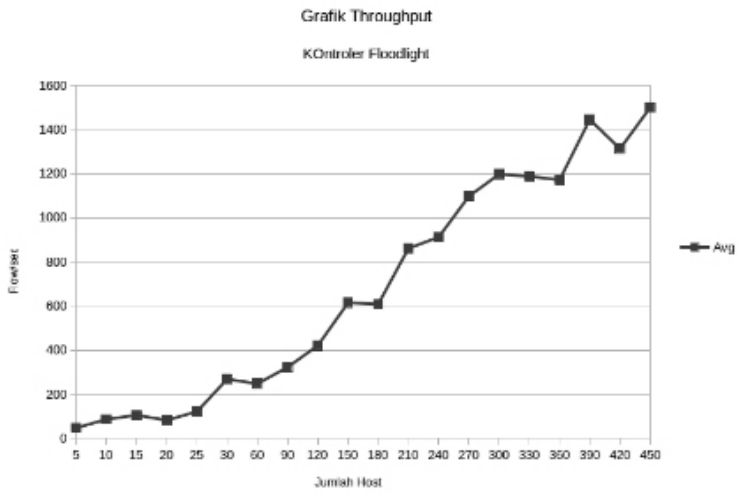
TABEL III . CONTOH DATA THROUGPUT KONTROLER FLOODLIGHT (RESPON/DETIK)

Jml Host		**20**	**90**	**120**	**330**	**450**
Respons/s	Min	79,06	322,66	419,88	1185,06	1498,26
	Max	84,55	326,11	422,04	1190,84	1503,06
	Avg	80,62	323,11	421,57	1187,79	1500,38

Dari data yang diperoleh, dibuatkan grafik hasil pengujian *throughput* pada floodlight seperti terlihat pada gambar 4 berikut ini.

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121



Gambar 4. Hasil uji *throghput* pada kontroler floodlight (Dalam Respon/Detik)

Bila dilihat dari data rata-rata (avg) Respon/detik vs jumlah host yang ditampilakan pada Tabel III dan gambar 4 diatas, kontroler Floodlight dapat memberikan kemampuan yang baik karena dengan meningkatnya jumlah host yang ada, maka semakin tinggi pula jumlah flow/detik yang dapat dilayani oleh kontroler. Algoritma pengujian *throughput* menggunakan *cbench* [11], menyatakan dalam mode throughtput (mis., dengan '-t'): untuk setiap sesi: selama buffer tidak penuh: (1)package_in diantrikan untuk diproses, (2)menghitung flow_mod saat mereka kembali. Dengan Algoritma yang lebih sederhana, maka *cbench* memberikan hasil yang jauh lebih stabil pada pengujian *throughput* dan berbanding lurus dengan jumlah *host* yang ditangani oleh kontroler.

B. Pengujian Kontroler Onos

Uji Latency

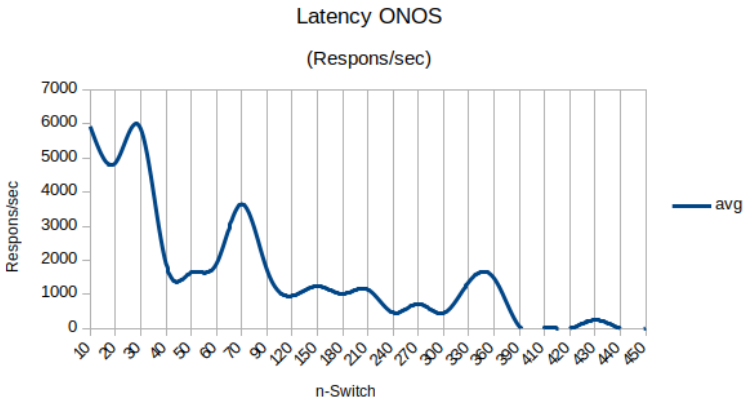
Sample data hasil uji *latency* pada kontroler ONOS dapat dilihat pada sample data Tabel IV dimana pengujian dilakukan pada jumlah switch 10 hingga 450 switch dengan kenaikan rentangan 10 switch. Jumlah pengujian setiap jumlah switch adalah 10x pengulangan.

TABEL IV. CONTOH DATA LATENCY KONTROLER ONOS (RESPON/DETIK)

Jml Switch		**30**	**70**	**210**	**360**	**450**
Respons/sec	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

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121



Gambar 5. Hasil uji *Latency* pada kontroler ONOS (Dalam Respon/Detik)

Dari data yang terlihat Tabel IV dan gambar 5 diatas bila dilihat dari data rata-rata (avg) respon/detik vs switch kontroler ONOS terlihat bahwa semakin banyak switch yang terlibat semakin rendah pula respon yang dapat diberikan oleh kontroler dalam mengangani flow. Kontroler ONOS mengalami keterlambatan dimulai pada switch dengan jumlah diatas 40, sehingga jumlah switch optimum untuk kontroler ONOS adalah antara 1 switch hingga40 switch. Jumlah switch yang digunakan pada pengujian throughput sebanyak 30 switch karna memiliki respon tertinggi perdetiknya.

Berbeda dengan kontroler floodlight, ONOS menunjukkan kemampuan yang terus menurun sejalan dengan bertambahnya jumlah switch yang terlibat. Penurunan ini dapat disebabkan oleh sifat dari kontroler ini yang lebih merupakan kontroler dengan sifat terdistribusi. ONOS menurut pengembangnya memiliki kemampuan dalam scalability, dan memiliki core yang terdistribusi, dengan algoritma *cbench* yang melakukan pengulangan dan melakukan penentuan *latency* berdasarkan waktu respon f*lood_mod* dapat dimungkinkan *cbench* bukan merupakan *tool* yang cocok dalam pengujian untuk kontroler ONOS.

Uji Throughput

Sample data hasil uji *throughput* pada kontroler ONOS dapat dilihat pada sample data Tabel V dimana pengujian dilakukan pada jumlah switch 5 hingga 450 *host* dengan kenaikan rentangan 10 *host*. Jumlah pengujian setiap jumlah switch 30 adalah 10x pengulangan.

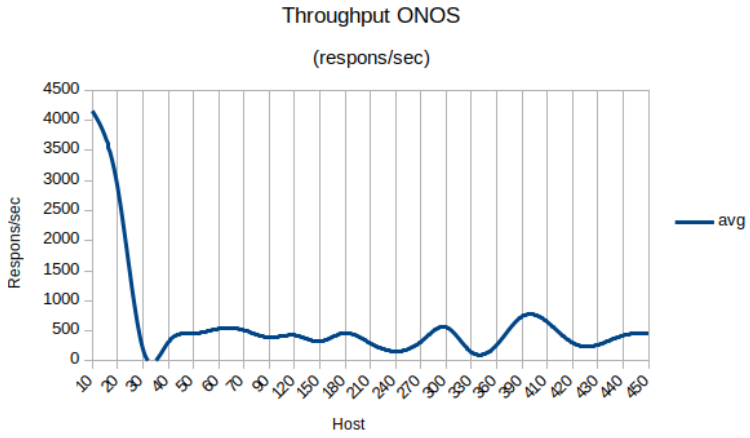
TABEL V. CONTOH DATA THROUGHPUT KONTROLER ONOS (RESPON/DETIK)

Jml Switch		**30**	**70**	**210**	**360**	**450**
respons/sec	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	511,41	293,41	267,32	466,09

Dari data yang diperoleh, hasil pengujian *throughput* pada ONOS dapat dilihat pada gambar 6 berikut ini.

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 02, Desember 2019 : 111 - 121



Gambar 6. Hasil uji *Latency* pada kontroler ONOS (Dalam Respon/Detik)

Bila dilihat dari data rata-rata (avg) respon/detik vs jumlah *host* yang ditampilakan pada Tabel V dan gambar 6 diatas, kontroler ONOS memberikan kemampuan respon yang cendrung stabil pada jumlah *host* diatas 30 *host*, namun sempat mengalami penurunan draktis pada jumlah 10 hingga 30 *host*.

Hasil dari pengujian througput pada kontroler ONOS jumlah respon/detik yang dihasilkan berada pada nilai 500 respon/detik dan tetap stabil walaupun dengan jumlah host yang terus bertambah. Walau menghasilkan nilai yang relatif kecil, namun kontroler ONOS memberikan nilai yang stabil.

IV. KESIMPULAN

Kontroler merupakan bagian yang menentukan dalam arsitektur SDN. Performa dari Kontroler ternyata harus pula diuji. Pengujian dari kontroler memberikan pandangan yang lebih baik dalam pemilihan kontroler yang akan digunakan, walau dalam pengujian ini masih perlu penyempurnaan terutama dalam pengujian dengan elemen lain. Pengujian ini memberikan hasil bahwa kontroler floodlight lebih unggul dalam memberikan respon pada pengujian *Througput* dan pengujian *Latency* dibandingkan dengan kontroler ONOS.

Peneliti menyadai, hasil penelitian ini belum jauh dari kata sempurna dan dapat dijadikan acuan, namun dengan hasil penelitian ini dapat memberikan wawan yang lebih tentang unjuk kerja kontroler. Elemen penentu dalam pengujian unjuk kerja adalah ketelitian dalam membaca data dan dalam penentuan tool yang akan digunakan. Dalam penentuan tool harus lebih dipertimbangkan apakah tool cocok digunakan untuk instrumen yang akan diujikan atau tidak dan pada respon kontroler yang mana yang akan kita ujikan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang besar kepada rekan-rekan Prodi Pendidikan Teknologi Informasi, Civitas STKIP PGRI Tulungagung, DPRM DIKTI dengan kontrak nomor 113/SP2H/LT/DPRM/2019 dan perjanjian nomor 096/SP2H/LT/MONO/L7/2019 dan 317/STKIP PGRI/TA/III/2019 dengan SK nomor 7/E/KPT/2019 yang telah menunjang penelitian ini.

DAFTAR PUSTAKA

[1] G. R. D. T. Muntaner, 'Evaluation of OpenFlow Controllers', 2012.
[2] D. Turull, M. Hidell, and P. Sjödin, 'Performance evaluation of OpenFlow controllers for network virtualization', High Perform. Switch. Routing (HPSR), 2014 IEEE 15th Int. Conf., pp. 50–56, 2014

[3] S. Rowshanrad, V. Abdi, and M. Keshtgari, "PERFORMANCE EVALUATION OF SDN CONTROLLERS : FLOODLIGHT AND OPENDAYLIGHT," IIUM Eng. J., vol. 17, no. 2, hal. 47–57, 2016
[4] S. Asadollahi and B. Goswami, "Experimenting with Scalability of Floodlight Controller in Software Defined Networks," in International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT) Experimenting, 2018, no. December, pp. 288–292
[5] S. Asadollahi and B. Goswami, "Experimenting with Scalability of Floodlight Controller in Software Defined Networks," in International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT) Experimenting, 2018, no. December, pp. 288–292
[6] L. Zhu, K. Sharif, F. Li, X. Du, and M. Guizani, "SDN Controllers : Benchmarking & Performance Evaluation," CoRR, vol. abs/1902.0, pp. 1–14, 2019
[7] B. Xiong, K. Yang, J. Zhao, W. Li, and K. Li, "Performance evaluation of OpenFlow-based software-defined networks based on queueing model," Comput. Networks, vol. 102, pp. 172–185, 2016
[8] I. Z. Bholebawa and U. D. Dalal, "Performance Analysis of SDN / OpenFlow Controllers :POX Versus Floodlight," Wirel. Pers. Commun., no. September, 2017
[9] L. Mamushiane, A. Lysko, S. Dlamini, "A comparative evaluation of the performance of popular SDN controllers", In2018 Wireless Days (WD) 2018 Apr 3 (pp. 54-59). IEEE
[10] R. Kartadie, F. Rozi, MA. Nugroho, "SOFTWARE-DEFINED NETWOR, PARADIGMA BERINOVASI DALAM JARINGAN KOMPUTER," , edisi ke-1, Bandung, Indonesia, 2018, bab 4, hal.82.
[11] R. Sherwood, "oflops/cbench at master · mininet/oflops · GitHub," 2013. [Daring]. Tersedia pada: https://github.com/mininet/oflops/tree/master/cbench. [Diakses: 16-Okt-2019].