SIMULASI JARINGAN SOFTWARE DEFINED NETWORK MENGGUNAKAN PROTOKOL ROUTING OSPF DAN RYU CONTROLLER

SIMULATION OF SOFTWARE DEFINED NETWORK USING OSPF ROUTING PROTOCOL AND RYU CONTROLLER

Roni Fernando Simarmata[1], Rohmat Tulloh[2], Yuli Sun Haryani[3]

[1,2,3]Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom
[1]ronifernando@student.telkmuniversity.ac.id, [2]rohmattulloh@tass.telkomuniversity.ac.id, [3]yulisun@tass.telkomuniversity.ac.id

Abstrak

Perkembangan teknologi jaringan semakin besar, tentunya membutuhkan kecepatan yang lebih cepat lagi dengan cara melakukan mekanisme konfigurasi untuk membangun jaringan internet yang disebut dengan *routing*. Semakin besar jaringan maka dibutuhkan teknologi jaringan yang semakin baik dengan munculnya *Software Defined Network*(SDN) merupakan sebuah konsep baru yang digunakan untuk mengatasi masalah jaringan tradisional dengan melakukan pemisahan antara *control plane* dan *data plane* dalam suatu perangkat yang berbeda. Komunikasi antara *control plane* dan *data plane* menggunakan protokol *openflow*. SDN memiliki beberapa kemampuan dalam banyak metode teknologi jaringan seperti *routing*, sehingga dengan adanya SDN ini diharapkan dapat menjalan kan metode yang terdapat pada jaringan konvensional.

Untuk mengatasi masalah di atas maka perlu dilakukan simulasi routing *open shortest path first (OSPF)* pada jaringan SDN dan. Simulasi ini dilakukan menggunakan simulasi jaringan pada *Virtual Machine* (VMWare) dan menggunakan *Ryu controller* yang terhubung dengan 8 switch yang di *setting* menjadi *switch openflow* kemudian akan dilihat perbandingan dengan jaringan konvensional dengan topologi yang sama dengan simulasi jaringan SDN.

Pengerjaan proyek akhir simulasi tersebut yaitu dapat membuktikan kinerja *Ryu Controller* pada *routing* OSPF di jaringan SDN dengan routing OSPF. Kinerja jaringan ini akan dibuktikan dengan cara mengirim paket dari satu user ke user lain dan dilihat juga dari sisi *Quality Of Service*(QOS) yaitu *Throughput*, *Delay*, *Packet loss*, dan *Jitter*, dan juga dan dibandingkan nilai QOS antara jaringan SDN dan jaringan Konvensional yang dibuat dengan topologi yang sama, dan nilai rata-rata dari QOS TCP adalah *Throughput* Konvensional 2,07 Mbit/s, SDN 2,03 Mbit/s untuk *delay* Konvensional 604,58 ms, SDN 0,21 ms untuk *jitter* Konvensional 3,17ms , SDN 0,069ms sedangkan UDP *Throughput* Konvensional 2,07 Mbit/s dan SDN 1,99 Mbit/s untuk *delay* Konvensional 598,84ms, SDN 0,18ms untuk *jitter* Konvensional 3,13ms, SDN 0,03ms, untuk *packet loss* Konvensional 0,007%, SDN 0%.

Kata Kunci ***: Software Define Network ; OSPF; Vmware; Ryu controller; Openflow.***

Abstract

The development of network technology is greater, of course require a faster speed again by way of configuration mechanisms to build an Internet network called routing. The bigger the network, the better network technology is needed with the emergence of Software Defined Network (SDN) is a new concept that is used to overcome the problem of traditional network by doing separation between control plane and data plane in a different device. Communication between control plane and data plane using openflow protocol. SDN has several capabilities in many network technology methods such as routing, so with the existence of SDN is expected to run the method contained in conventional network.

On how to solve the problem above, we need to carry out an open shortest path first(OSPF) routing simulation on the SDN network. This simulation is set up using Virtual Machine(VMWare) networking simulation and integrated with Ryu Controller which is connected to 8 openflow switches and then with the same topology, any ratio difference between conventional network and the SDN network simulation will be discussed and reviewed.

The fulfillment of this final simulation project is to prove the performance of Ryu Controller on OSPF routing in SDN network. This networking performance will be proved by sending any package from one user to others and observe its Quality of Service(QoS) aspect which is throughput, delay, packet loss, and jitter. QOS value between the conventional network and the SDN network with the same topology will also be paid attention. Average values of the QOS TCP are as follow Conventional throughput 2,07 Mbit/s, SDN 2,03 Mbit/s for conventional delay 604,58ms, SDN 0,21ms for conventional jitter 3,17ms, SDN 0,069ms while for UDP

conventional throughput 2,07 Mbit/s and SDN 1,99 Mbit/s for conventional delay 598,84ms, SDN 0,18ms for conventional jitter 3,13ms, SDN 0,03ms, for conventional packet loss 0,007%, SDN 0%.

Keywords: Software Define Network; OSPF; Vmware; Ryu Controller; Openflow.

1. Pendahuluan

Teknologi jaringan yang sangat berkembang pesat menjadi salah satu tuntutan evolusi jaringan untuk terus berkembang lebih baik lagi. Pada kenyataan nya saat ini banyak jaringan yang mulai jenuh sehingga membuat banyaknya penelitian dan percobaan platform software defined network (SDN) dengan tujuan memperbaiki kondisi jaringan tersebut. Dibandingkan dengan jaringan konvensional, *Software Defined Networking* (SDN) memberikan kemudahan kepada pengguna dalam mengembangkan aplikasi pengontrol jaringan dengan memisahkan fungsi *data plane* dari *control plane*. Pemisahan *data plane* dan *control-plane* pada perangkat jaringan komputer seperti *Router* dan *Switch* memungkinkan untuk memprogram perangkat tersebut sesuai dengan yang diinginkan secara terpusat. Pemisahan inilah yang mendasari terbentuknya paradigma baru dalam jaringan komputer yang disebut *Software Defined Networking* (SDN)(US: Open Networking Foundation. 2013).

Berdasarkan dari proyek akhir sebelumnya yang dilakukan oleh Ayu Irmawati[4] membahas mengenai simulasi dan implementasi jaringan SDN menggunakan *pox controller* dan OSPF sebagai *routing* dengan 4 *switch openflow*, dan jurnal Yuli Sun Haryani [13] membahas mengenai implementasi *openvswitch* dan *ryu controller* menggunakan *Rasberry pi* sebagai tempat *controller* di implementasi dan hasil yang dilihat pada implementasi ini adalah kualitas dari setiap link atau jalur yang terdapat pada jaringan , Rohmat Tulloh dan Ridha Muldina[7] pada jurnal ini membahas mengenai simulasi penerapan algoritma OSPF di jaringan SDN di jurnal ini juga memakai pengukuran *background traffic*, dan pengerjaan proyek akhir ini mengembangkan dengan menerapkan kinerja *routing* OSPF Pada SDN dengan menggunakan *Ryu controller*. Kehebatan teknologi SDN dalam jaringan komputer dianggap menarik oleh penulis, sehingga tertarik untuk mensimulasikan *routing OSPF* dengan menggunakan *Ryu controller* dengan simulasi dengan 8 buah *switch openflow* , dan dilakukan pengukuran QOS pada jaringan ini untuk melihat perbandingannya dengan jaringan konvensional dan dibandingkan..

2. Dasar Teori

2.1 Software Defined Network

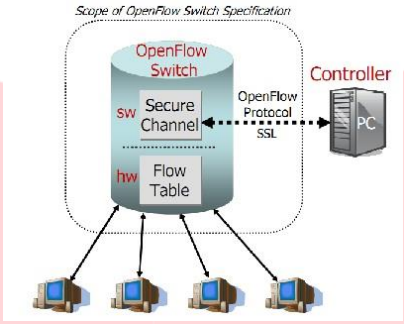
Software Defined Network (SDN) adalah konsep atau paradigma baru dalam mendisain, mengelola dan mengimplementasi jaringan terutama untuk mendukung kebutuhan dan inovasi dalam jaringan komputer semakin kompleks. Konsep dasar SDN adalah dengan melakukan pemisahan antara *control plane* dan *forwarding/ data plane* sehingga memiliki kontrol yang terpusat sehingga tidak perlu melakukan banyak konfigurasi perangkat di masing-masing perangkat jaringan. Kalau dalam jaringan konvensional, perangkat jaringan yang memiliki fungsi *control plane* dan *data plane* berada dalam fungsi satu perangkat sehingga harus mengkonfigurasinya satu persatu sehingga kurang efisien. Supaya lebih efisien dalam pengimplementasian jaringan, maka tercipta *Software Defined Network* (SDN) yang dikembangkan di UC Berkley dan Standford University pada tahun 2008 dan pada tahun 2011 teknologi SDN dan Openflow dipromosikan oleh *Open Networing Foundation[11]*.

Teknologi SDN memiliki dua karakterisitik, yang pertama SDN memisah antara *control plane dan data plane*. Kedua SDN menggabungkan *control plane* setiap perangkat menjadi sebuah kontroler yang berbasiskan *programmeable software*. Sehingga sebuah kontroler tersebut dapat mengontrol banyak perangkat dalam sebuat *data plane*. SDN mensentralisasikan jaringan dalam sebuah kontroler sehingga lebih mempermudah dalam pengoperasian, dan memelihara jaringan secara keseluruhan.

2.2 OpenFlow

OpenFlow adalah protokol yang memungkinkan server memberitahukan switch jaringan tempat mengirim paket[8]. Dalam jaringan konvensional, setiap switch memiliki perangkat lunak berpemilik yang memberitahukan apa yang harus dilakukan[8].

Openflow mendefinisikan infrastruktur *flow-based forwarding* dan *Application Programmatic Interface* (API) standart yang memungkinkan *controller* untuk mengarahkan fungsi dari *Switch* melalui saluran yang aman (*secure channel)*[10*]*. Bisa dilihat pada Gambar 2.2 bahwa fungsi openflow sebagai penghubung antara *controller* yaitu termasuk dalam *control plane* dengan *data plane* melewati *secure channel* lalu ke *flow tabel* dan diteruskan ke user.



Gambar 2.1 *OpenFlow*[8]

2.3 *Ryu Controller*[13]

Ryu Controller adalah sebuah perangkat lunak yang terbuka, perangkat lunak yang didefinisikan (SDN) Controller yang dirancang untuk meningkatkan kelincahan jaringan dengan mudah mengatur dan menyesuaikan bagaimana lalu lintas ditangani. Secara umum, SDN Controller adalah otak lingkungan SDN, mengkomunikasikan informasi ke switch dan router dengan APIs, dan sampai pada aplikasi dan logika bisnis dengan APIu. Pengendali Ryu didukung oleh NTT dan juga ditempatkan di data center NTT.

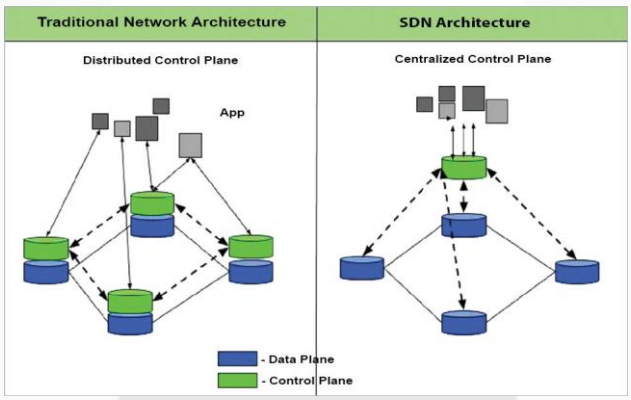
2.4 *Protocol Routing OSPF*

Protokol OSPF adalah salah satu protokol ip *routing* dan merupakan *Protocol Gateway Interior* (IGP) untuk internet yang digunakan untuk mendistribusikan informasi routing keseluruh jaringan yang saling terhubung. Pada OSPF dikenal sebuah istilah *Autonomus System* (AS) yaitu sebuah gabungan dari beberapa jaringan yang sifatnya routing dan memiliki kesamaan metode serta policy pengaturan network, yang semuanya dapat dikendalikan oleh network administrator[6]. Dan memang kebanyakan fitur ini digunakan untuk management dalam skala jaringan yang sangat besar. Oleh karena itu untuk mempermudah penambahan informasi routing dan meminimalisir kesalahan distribusi informasi routing, maka OSPF bisa menjadi sebuah solusi [6].

2.5 Perbedaan arsitektur jaringan SDN dan jaringan Konvensional

Pada Gambar 2.2 mengilustrasikan perbedaan antara arsitektur jaringan SDN dan jaringan konvensional. Pada gambar di sisi kiri, menunjukan arsitektur jaringan SDN dimana control plane dan data plane dipisahkan.

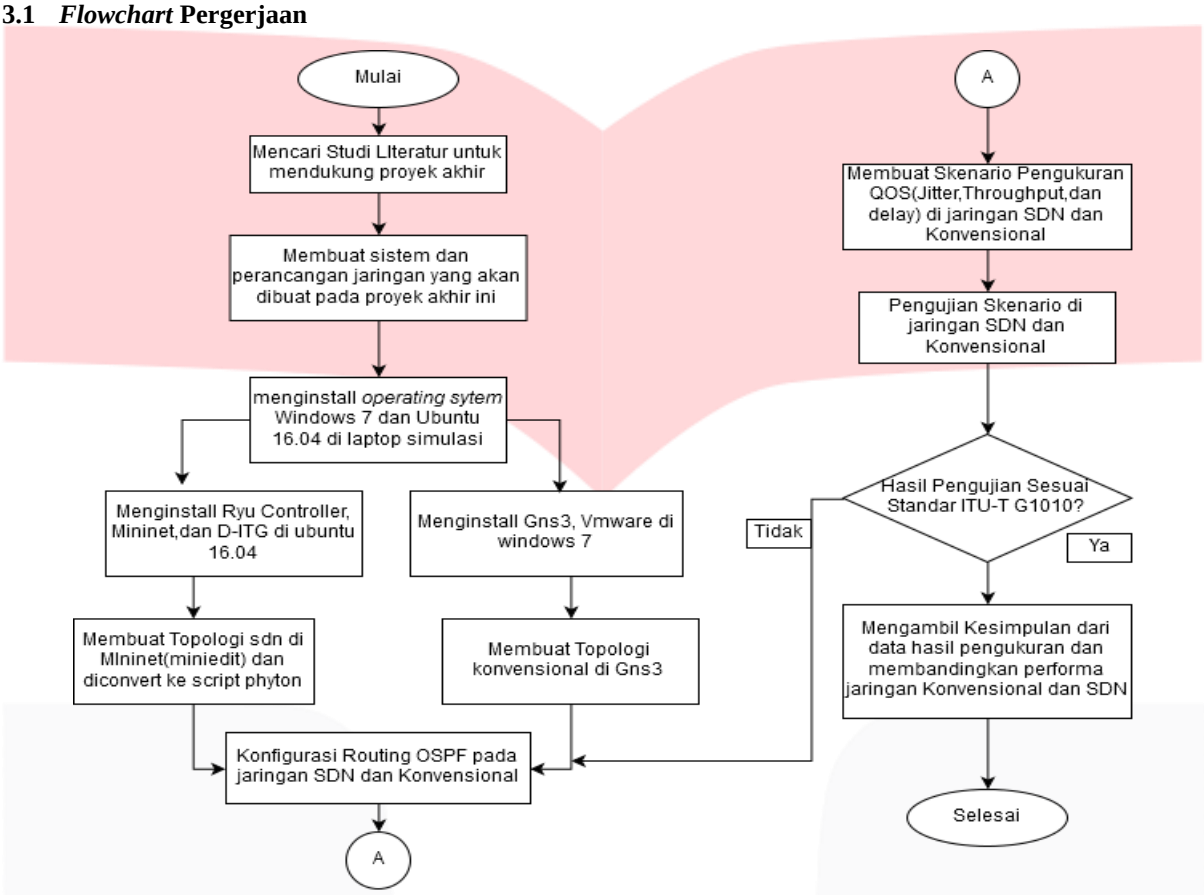
Sedangkan pada gambar di sisi kanan, menunjukan arsitektur jarigan konvensional dimana router memiliki fungsi control plane dan data plane berada di masing-masing router. Ini menunjukan bahwa pada jaringan SDN, control plane menggunakan struktur tersentralisasi, dan Sedangkan pada jaringan konvensional, control plane menggunakan struktur terdistribusi.



Gambar 2. 2 perbedaan traditional network dan SDN network architecture [3]

3. Perancangan dan Simulasi

3.1 *Flowchart* Pergerjaan



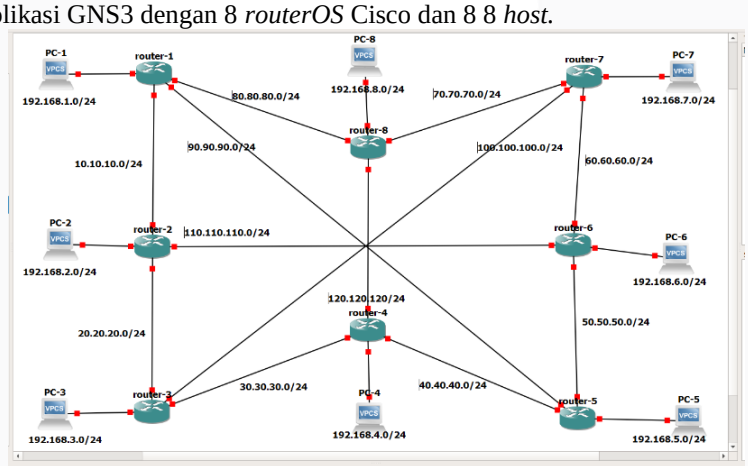
Gambar 3.1 *Flowchart* Metode Pengerjaan

3.2 Desain Topologi jaringan

Proyek akhir ini melakukan dua pembuatan topologi yaitu topologi Konvensional dan topologi SDN berikut gambar topologinya :

3.2.1 Topologi jaringan Konvensional

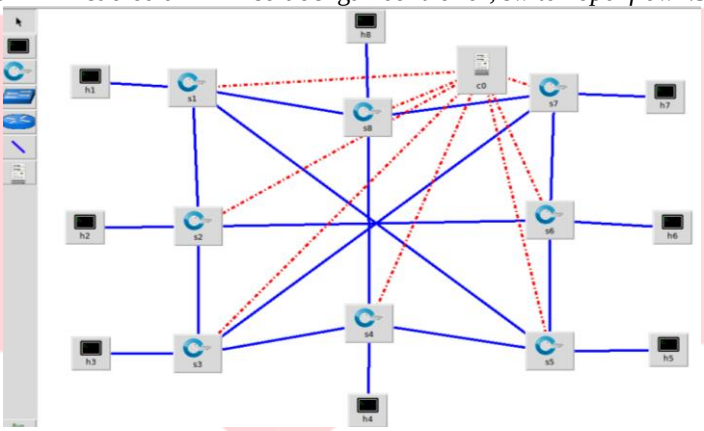
Dibuat di aplikasi GNS3 dengan 8 *routerOS* Cisco dan 8 8 *host.*



Gambar 3.2 Topologi Jaringan Konvensional GNS3

3.2.2 Topologi jaringan SDN

Dibuar di aplikasi Mininet didalam miniedit dengan *controller, switch openflow1.3*,dan *host.*



Gambar 3.3 Topologi Jaringan SDN di miniedit

3.3 Pembuatan Simulasi Jaringan Konvensional dan Jaringan SDN

Pembuatan simulasi ini dibuat di dua simulator yaitu GNS3 untuk membuat jaringan konvensional dan emulator mininet digunakan untuk membuat simulasi jaringan SDN berikut ini langkah-langkah membuat simulasi jaringan Konvensional dan jaringan SDN:

a. Pembuatan Simulasi Jaringan Konvensional

Pembuatan ini dilakukan di GNS3, Berikut adalah langkahnya:

1. Pembuatan Topologi

   Topologi ini dibuat dengan 8 *Router* yang saling terhubung satu dengan yang lain, dan masing-masing *Router* terhubung dengan 1 *Host.*

2. Konfigurasi *interface* yang terdapat pada *router*

   Konfigurasi ini dilakukan untuk mengenalkan jalur antar *router,* dan konfigurasi ini dilakukan di semua *router* dari *router* 1 sampai *router* 8, dan dilakukan satu persatu setiap *router.*

3. Konfigurasi *routing OSPF* pada *router*

   Konfigurasi *routing* OSPF ini dilakukan untuk membuat jalur pengiriman paket atau yang biasa disebut jalur *routing* yang *dynamic* atau bisa berubah jika ada jalur yang terputus, dan konfigurasi ini dilakukan di semua *router* dari *router* 1 sampai dengan *router* 8, dengan cara config Network ID yang terhubung didalam semua *interface* dalam *router*, dan proses ini dilakukan juga secara satu persatu setiap router, dan jika sudah selesai maka akan dilihat apakah sudah masuk dengan cara cek ip route di masing-masing router.

4. Setting PC simulasi

   Pc simulasi yang dipake adalah Ubuntu 12.04 dilakukan setting ip dilakukan sesuai dengan ip yang ada pada router yang terhubung.

b. Pembuatan Simulasi Jaringan

Pembuatan ini dilakukan di Mininet, Berikut adalah langkahnya:

1. Pembuatan Topologi

   Topologi ini dibuat dengan 1 *controller* terhubung ke 8 *switch OpenFlow1.3* yang saling terhubung satu dengan yang lain, dan setiap *switch OpenFlow1.3* terhubung dengan 1 *Host*.

2. Mengganti *setting controller*

   Setting controller diganti menjadi *remote controller*

3. *Setting Preferences*

   Dilakukan setting didalam *preferences* diantaranya adalah mengganti *subnet mask* di *ip base* dan juga setting "*start CLI*" di *Ceklist* dan yang terakhir ganti menjadi *openflow1.3* untuk jenis *switch*nya.

4. *Save* dan *export file* topologi

   Save file miniedit ini akan di *save* dengan format *file* .mn dan akan di*export* dalam bentuk *file python* dengan format .py yang akan dijalankan nanti sebagai *data plane*

5. Menjalankan *Controller Ryu*

   Script yang telah ada akan dijalankan *controller ryu* ini berfungsi sebagai control plane pada jaringan SDN ini dan *script* yang telah ada ini adalah *script* dengan fungsi algoritma *routing* OSPF yaitu algoritma Djikstra.

6. Menjalankan file topologi

   File yang telah di*export* dalam format .py ini dijalankan untuk memastikan bahwa topologi yang dibuat tidak *error*.

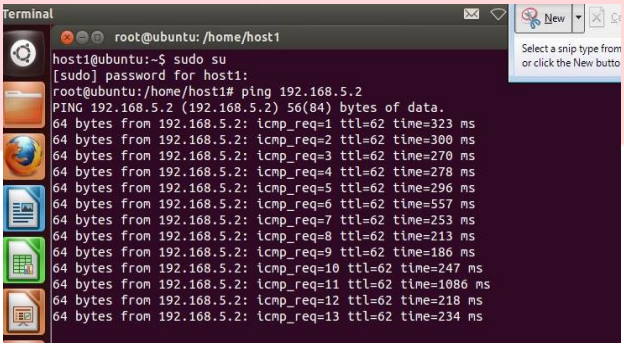
4. Pengukuran dan Analisa simulasi

4.1 Test koneksi

Test koneksi dilakukan di dua simulasi jaringan, dijaringan Konvensional pada PC yang terhubung dengan *Router*, dan dijaringan SDN di *Host* yang terhubung pada *switch OpenFlow*

a. Test Koneksi Jaringan Konvensional

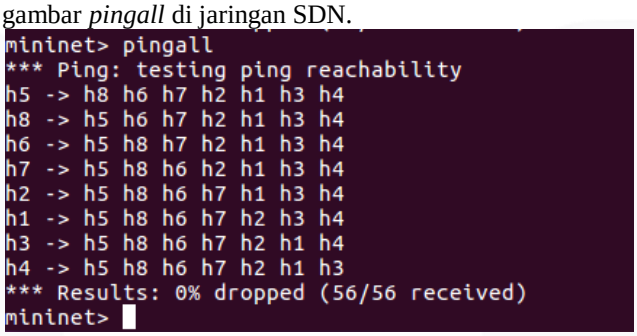
Test koneksi dijaringan Konvensional dilakukan dengan melakukan ping ke host lain melalui ping ip pc lain yang telah terhubung. Berikut adalah gambar test ping pc lain di pc Vmware.



Gambar 3.4. Tampilan test ping ke host lain melalui VMware

b. Test koneksi Jaringan SDN

Test koneksi di jaringan SDN dengan cara pertama menjalankan Ryu Controller setelah itu panggil topologi yang telah di export ke format .py dan setelah dipanggil lalu *pingall* untuk menguji koneksi jaringan SDN berikut gambar *pingall* di jaringan SDN.



Gambar 3.5 Tampilan test *pingall* pada jaringan SDN

4.2 Hasil Pengukuran

4.2.1. Pengukuran dengan setting default

Berdasarkan hasil pengujian yang telah dilakukan selama 10 kali pada proyek akhir, dan selanjutkan hasil pengukuran yang telah didapat akan dibandingkan dengan stardarisasi ITU-T G1010 *End-user multimedia* QOS *Catagories.* Berikut standart yang digunakan:

Tabel 4.1. Refrensi Standarisasi QOS ITU-T G1010

Parameter Performansi	Data
One way Delay (*Latency*)	*Preffered <15s; Acceptable<60s NB : Amount of Data 10 kb - 10 MB*
Jitter	NA
Throughput	NA
Packet Loss	3%

Pengukuran telah berhasil dan telah didapatkan hasil yang sesuai dengan standar ITU-T G1010. Berikut merupakan rangkuman hasil pengukuran rata-rata dari pengambilan 10 kali pengukuran:

Tabel 4.2. Hasil Pengukuran rata-rata dari 10 kali pengukuran

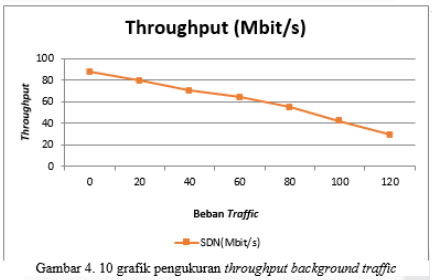
No	Parameter	Hasil pengukuran		ITU-T G1010
		Konvensional	SDN	
1	*Throughput* TCP	2077,3579 Kbit/s	2036,151 Kbit/s	NA
	UDP	2070,582 Kbit/s	1998,2654 Kbit/s	NA
2	*Delay* TCP	0,604587 s	0,00021 s	Preffered <15s; Acceptable<60s
	UDP	0,598841 s	0,00017 s	
3	*Jitter* TCP	0,003172 s	0,000069	NA
	UDP	0,003139 s	0,000034 s	NA
4	*Packet Loss*(UDP)	0,01%	0%	3%

4.2.2. Pengukuran dengan *Background Traffic*

Analisa performa jaringan Konvensional dan jaringan SDN dengan *routing protocol* OSPF menggunakan *tools* pengukuran Iperf dengan menambahkan *background traffic* disimulasi jaringan yang telah dibuat adalah dengan mengatur nilai beban *traffic* jika dilakukan komunikasi dari h1 ke h5 yang digunakan pada simulasi jaringan Konvensional dan jaringan SDN mulai dari tidak ada *setting background traffic* sampai dengan *setting background traffic* mencapai 100 Mbps. Adapun yang diukur adalah UDP pada jaringan Konvensional dan jaringan SDN. Berikut parameter yang didapat dari pengukuran jaringan Konvensional dan jaringan SDN :

Tabel 4. 9 Pengukuran *throughput background traffic*

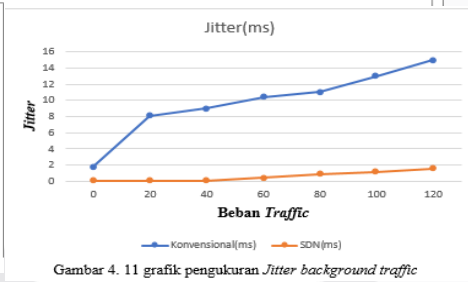
Beban *Traffic (Mb)*	SDN(Mbit/s)
0	88
20	80
40	70,5
60	64,1
80	55
100	42
120	30



Gambar 4. 10 grafik pengukuran *throughput background traffic*

Tabel 4. 10 pengukuran *Jitter background traffic*

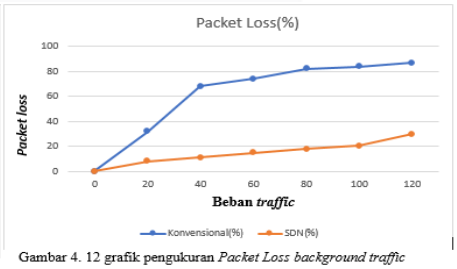
Beban *Traffic (Mb)*	Konvensional(ms)	SDN(ms)
0	1,8	0,016
20	8,1	0,035
40	9	0,037
60	10,4	0,4
80	11	0,82
100	13	1,1
120	15	1,6



Gambar 4. 11 grafik pengukuran *Jitter background traffic*

Tabel 4. 11 pengukuran *Packet Loss background traffic*

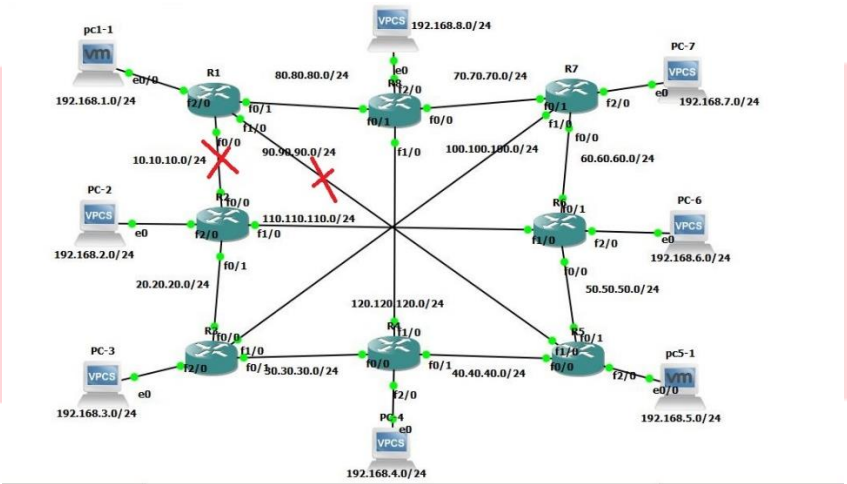
Beban *Traffic*	Konvensional(%)	SDN(%)
0	0	0
20	32	8
40	68	11
60	74	15
80	82	18
100	84	21
120	87	30



Gambar 4. 12 grafik pengukuran *Packet Loss background traffic*

4.2.3. Pengukuran *Time Convergence*

Time Convergence adalah waktu yang dibutuhkan untuk suatu jaringan melakukan *list table routing* baru atau pencarian jalur baru jika jalur diputus. Percobaan ini dilakukan 4 kali pemutusan jalur dan akan dilihat dan dibandingkan performa jaringan Konvensional dan jaringan SDN jika dilakukan pemutusan jalur dan akan dilihat perbedaan *time convergence* Jaringan konvensional dan jaringan SDN, dan pengukuran ini dilakukan menggunakan *stopwatch*.

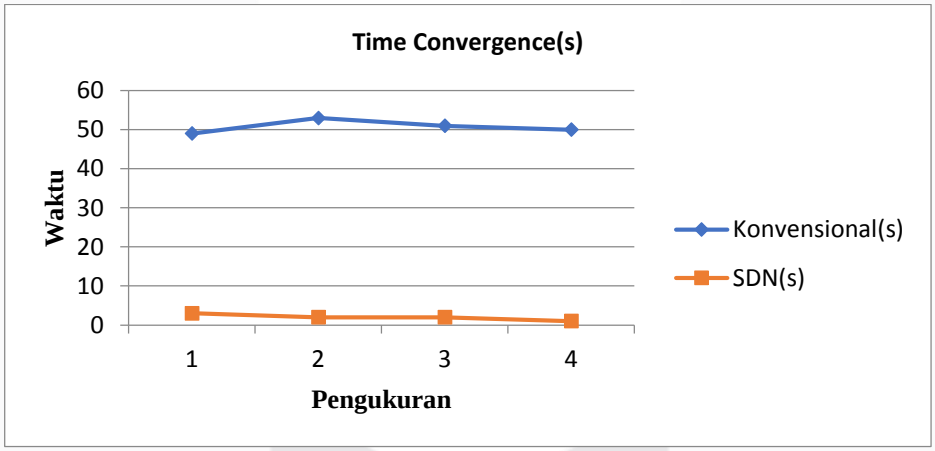


Gambar 4. 1 Gambar pemutusan jalur pada pengukuran time convergence

Gambar 4.13 diatas menunjukkan pemutusan jalur yang akan dilakukan pada pengukuran time convergence pada jaringan Konvensional dan jaringan SDN pada proyek akhir ini dan diukur perbandingan performa jaringan Konvensional dan jaringan SDN.

Tabel 4. 1 *Time Convergence jaringan Konvensional* dan jaringan *SDN*

Pengukuran	Konvensional(s)	SDN(s)
1	49	3
2	53	2
3	51	2
4	50	1



Gambar 4. 2 grafik pengukuran Time Convergence

Dari pengukuran *time convergence* dapat dilihat teknologi jaringan SDN lebih bagus dikarnakan *control plane* dan *data plane* terpisah jadi yang di data ulang semua jalur hanya pada satu tempat/perangkat berbeda dengan jaringan konvensional dimana *control plane* dan *data plane* masih pada setiap perangkat jadi lebih lama dalam mendata ulang jalur yang ada.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Pada proyek akhir ini dapat disimpulkan sebagai berikut :

1. Pada simulasi jaringan SDN dan jaringan Konvensional dapat dilihat dari performa pengukuran *protocol* TCP dan UDP, jaringan SDN memiliki performa yang lebih bagus dibanding dengan Konvensional karna *Control Plane* terpusat satu tidak seperti jaringan konvensional yang *control plane* dan *data plane* masih pada satu perangkat.
2. Pada simulasi jaringan Konvensional dan jaringan SDN berikut adalah nilai rata-rata dari setiap parameter pengukuran TCP dan UDP adalah *Throughput*, *Delay*, *Packet loss*, dan *Jitter*, dan juga dan dibandingkan nilai QOS antara jaringan SDN dan Konvensional dibuat dengan topologi yang sama, dan nilai rata-rata dari QOS TCP adalah *Throughput* Konvensional 2,07 Mbit/s, SDN 2,03 Mbit/s untuk *delay* Konvensional 604,58 ms, SDN 0,21 ms untuk *jitter* Konvensional 3,17ms , SDN 0,069ms sedangkan UDP *Throughput* Konvensional 2,07 Mbit/s dan SDN 1,99 Mbit/s untuk *delay* Konvensional 598,84ms, SDN 0,18ms untuk *jitter* Konvensional 3,13ms, SDN 0,03ms, untuk *packet loss* Konvensional 0,007%, SDN 0%.
3. Dari analisa *background traffic* yang didapat jaringan sdn lebih handal dapat dilihat dari nilai *packet loss*, *jitter*, dan *throughput*nya yang lebih baik dari jaringan konvensional dikarenakan *control plane* dan *data plane* jaringan sdn ini terpisah tidak seperti pada jaringan konvensional yang masih menyatu *control plane* dan *data plane*nya.
4. Dari analisa *time convergence* di jaringan SDN dan jaringan Konvensional didapat nilai *time convergence* dari jaringan SDN lebih baik/cepat dalam *time convergence* dibanding dengan jaringan konvensional dikarnakan *list table routing* yang didata hanya pada satu perangkat dikarnakan sudah ada *centralisasi control plane* dari jaringan SDN berbeda dengan jaringan konvensional yang semua perangkat masih dalam satu tempat *control plane* dan *data plane*

5.2 Saran

Saran untuk proyek akhir ini sebagai berikut :

1. Dapat mengembangkan simulasi Jaringan *Software Defined Network*(SDN) kembali dengan topologi yang sama menggunakan *controller* lain seperti ONOS atau NOX.
2. Dapat menggembangkan konfigurasi yang sama dengan simulasi untuk diimplementasikan.
3. Dapat mengembangkan simulasi yang lebih kopleks dibandingkan proyek akhir ini.
4. Dapat mengembangkan konfigurasi jaringan pada *controller* dan SDN *application*

Daftar Pustaka

[1] Advisor, I. M., *Openflow*, 08 August 2017. [Online]. Available: http://www.ingrammicroadvisor.com: http://www.ingrammicroadvisor.com/data-center/7-advantages-of-software-defined-networking. . [Accessed 2017 Desember 06].

[2] Azodolmolky, S, *Software Define Network with OpenFlow*. 1st ed, Birmingham: Packt publishing, 2013.

[3] Bhatia,J., *opensource*, 2017. [Online]. Available: http://opensourceforu.com/2017/10/primer-software-defined-networking-sdn-openflow-standard/. [Accessed 2018 january 20].

[4] Imawati A, *Implementasi Protokol Routing OSPF pada software Defined Network Berbasis RouterFlow. Proyek Akhir Fakultas Ilmu Terapan Universitas Telkom*, Bandung, 2017.

[5] Iwan Iskandar, A. H., *Analisa Quality of Service (QoS) Jaringan Internet Kampus (Studi Kasus: UIN Suska Riau)*, *CoreIT*, vol. 1, no. 2, 2015.

[6] Mikrotik, *Konfigurasi dasar OSPF*, t.thn. [Online]. Available: www.mikrotik.co.id: http://mikrotik.co.id/artikel_lihat.php?id=154 . [Accessed 07 Oktober 2017].

[7] Ridha Muldina Negara, R. T., "Infotel," *Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada jaringan Software Defined Network(SDN)*, no. IX(172), pp. 75-83, 2017.

[8] Rouse, M., June 2012. [Online]. Available: www.techtarget.com: http://whatis.techtarget.com/definition/OpenFlow . [Accessed 06 Oktober 2017].

[9] Santoso, K. A., *Konfigurasi dan Analisis Performasi Routing OSPF Pada Jaringan LAN dengan Simulasi Cisco Packet Tracer Versi 6.2*.,Teknik Elektro, vol. 1, no. 1, p. 70, 17 Agustus 1945.

[10] Siamak Azodolmolky, *Software Defined Networking with OpenFlow, Livery Street Birmingham B3 2PB*, UK: Packet Publishing Ltd, 2013.

[11] Tiade, A., *OSPF (Open Shortest Path First)*, Palembang: : Pusat Pengembangan, (t.thn.).

[12] Wildan, *Contribute to skripsi_sdn development by creating an account on*, 2017. [Online]. Available: https://github.com/wildan2711/skripsi_sdn/(accessed. [Accessed 15 April 2018].

[13] Yuli Sun Hariyani, I. I. D. D. S. M. N., *ROUTING IMPLEMENTATION BASED-ON SOFTWARE DEFINED NETWORK USING RYU CONTROLLER AND OPENVSWITCH*, vol. 78, pp. 295-298, 2016.