

# ANALISIS KINERJA JARINGAN BERBASIS SOFTWARE DEFINITION NETWORK DENGAN PROTOKOL OPENFLOW DI RRI BANDUNG

Nanang Hunaifi<sup>1)</sup>, Rovi Fauzan Akbar<sup>2)</sup>  
Program Studi Sistem Informasi<sup>1),2)</sup>,  
AMIK BSI Surakarta<sup>1)</sup>, Universitas BSI Bandung<sup>2)</sup>  
masnaing@gmail.com<sup>1)</sup>, rovifauzan95@gmail.com<sup>2)</sup>

## ABSTRAK

Jaringan komputer merupakan salah satu yang sangat dibutuhkan saat ini dalam perusahaan ataupun sekolah untuk interkoneksi antara satu sama lainnya, jaringan komputer saat ini masih terpaku pada *vendor* tertentu karna bersifat *close source* dengan konfigurasi yang manual jika ada penambahan perangkat, hal ini menyebabkan tidak efisien karena terlalu terpaku pada *vendor* dan harus mengkonfigurasi setiap perangkat yang ada. Dari permasalahan itu dibentuklah jaringan *SDN* (*Software Definition Network*) yang memungkinkan sebuah jaringan komputer tanpa terpaku pada *vendor* tertentu dengan konfigurasi terpusat pada satu *controller* sehingga tidak perlu konfigurasi manual jika ada penambahan dan lebih mudah mengontrolnya. Dengan mensimulasikan manajemen jaringan yang fleksibel dan perbedaan kecepatan pertukaran data saat pengiriman data, contoh dari perbedaan jaringan komputer sebelum dan sesudah menggunakan jaringan *SDN* berbasis protokol *OpenFlow* dengan hasil 446Mb lebih cepat saat pertukaran data, maka jaringan *SDN* dengan protokol *OpenFlow* merupakan solusi alternatif dari manajemen jaringan dan pertukaran paket data, hal ini diharapkan dapat memecahkan permasalahan yang ada di RRI Bandung.

**Kata Kunci :** *Jaringan computer, SDN (Software Definition Network), OpenFlow, Manajemen Jaringan*

## I. PENDAHULUAN

Jaringan untuk saat ini menjadi hal yang sudah mulai diperhatikan, karena menjadi infrastruktur dalam suatu perusahaan. Banyak pendekatan pendekatan untuk menyesuaikan kinerja dari jaringan agar lebih optimal dan lebih efisien, ada salah satu pendekatan jaringan yang sungguh menarik yaitu jaringan yang dapat diprogram oleh perangkat lunak (*Software Definition Network*). Seperti yang kita tahu bahwa *switch* memiliki dua bagian yang disebut dengan bagian control dan bagian data, tugas *SDN* disini memisahkan antara kontrol dan data dalam jaringan tersebut<sup>[1]</sup>.

*OpenFlow* bisa menjadi protokol yang baik untuk komunikasi antara *switch* dan *software-based controller* dalam arsitektur jaringan *Software Definition Network* hal ini menjadi terobosan baru karena *SDN* dapat dipercaya mampu menjadi solusi untuk jaringan sekarang

yang bersifat sangat kaku, jaringan internet saat ini telah mengalami *ossification* sehingga jika tetap menggunakan jaringan yang sekarang maka pengembangan akan sangat sulit dilakukan. Jika seperti itu maka Jaringan akan sulit berkembang dan tidak signifikan. Berbeda dengan bagian lain pada perangkat komputer lainnya seperti halnya prosesor dan sistem operasi yang berkembang secara pesat sampai saat ini. apalagi pada saat ini perangkat jaringan yang ada saat ini terlalu bergantung pada *vendor* dan banyak sekali protokol *proprietary*, sehingga menyulitkan untuk berhubungan antara perangkat yang berbeda<sup>[2]</sup>.

Berdasarkan uraian diatas dapat dilihat bahwa pada saat ini jaringan komputer ini berperan penting dalam sebuah perusahaan, dan dengan menggunakan jaringan komputer yang dapat diprogram (*Software Definition Network*) yang menggunakan protokol *OpenFlow*. Kita bisa dengan mudah berhubungan walau dengan

berbeda vendor dalam perangkat lunak. Ini akan menambah efek efisiensi bagi perusahaan.

## II. LANDASAN TEORI

Jaringan komputer adalah dua komputer atau lebih yang saling terhubung dapat saling berkomunikasi dan berbagi sumber daya. Adapun media penghubungnya dapat berupa kabel atau nirkabel yang dapat berupa gelombang radio dan sinar inframerah<sup>[3]</sup>.

Dengan berkembangnya jaringan komputer yang semakin kompleks mengakibatkan rumitnya untuk manajemen suatu jaringan komputer. Karna semakin banyaknya perangkat jaringan komputer yang saling terhubung, maka dibutuhkan suatu mekanisme *monitoring* jaringan komputer yang didesain untuk memantau status dari infrastruktur jaringan komputer, memastikan perangkat dalam kondisi baik dan menjaga kestabilan operasional jaringan. maka dibutuhkan NMS (*Network Management System*). Aplikasi NMS secara umum bekerja menggunakan protokol SNMP (*Simple Network Management Protocol*), dengan informasi yang diperoleh berupa status *up/down device*, *cpu utilization*, *memory utilization*, *device type*, *traffic* jaringan dan lain sebagainya<sup>[4]</sup>.

### 1. *Software Definition Networking*

*Software Definition Networking (SDN)* adalah arsitektur jaringan yang muncul dimana kontrol jaringan dipisahkan dari forwarding dan secara langsung dapat diprogram. Migrasi kontrol inilah yang sebelumnya terikat erat pada perangkat jaringan individu, yang memungkinkan dapat mengakses ke perangkat komputasi, hal inilah yang mendasari infrastruktur untuk diabstraksikan menjadi aplikasi dan layanan jaringan, yang dapat memperlakukan jaringan sebagai entitas logis atau virtual<sup>[1]</sup>.

Kecerdasan jaringan (secara logis) terpusat pada pengontrol SDN berbasis perangkat lunak, yang digunakan untuk melihat jaringan secara global. Sebagai hasilnya, munculah aplikasi jaringan dan kebijakan mesin sebagai saklar tunggal yang logis. Dengan SDN, perusahaan dan operator dapat mengontrol independen vendor atas

keseluruhan jaringan dari satu titik logis, yang sangat menyederhanakan jaringan desain dan operasi. SDN juga sangat menyederhanakan perangkat jaringan diri mereka sendiri, karena mereka tidak perlu lagi memahami dan memproses ribuan standar protokol tetapi hanya menerima instruksi dari Kontrol SDN<sup>[1]</sup>.

### 2. Protokol OpenFlow

*OpenFlow* yaitu protokol untuk komunikasi antara *switch* dengan sebuah kontroler yang ada dalam *SDN* Arsitektur. *OpenFlow* juga menyediakan protokol terbuka untuk memprogram *flowtable* di *switch* dan *router* yang berbeda, bukan hanya itu *OpenFlow* juga dapat membuat administrator jaringan berpartisipasi langsung ke dalam lalu lintas data untuk membuat atau meneliti aliran data, dengan ini memungkinkan peneliti dapat mengontrol aliran data mereka sendiri dengan memilih rute paket data yang mereka ikuti dan memproses yang akan diterima oleh mereka<sup>[2]</sup>.

Dengan cara ini, peneliti dapat mencoba protokol *routing* yang baru, model keamanan, skema pengalamatan, dan bahkan mencari alternative untuk menuju *IP*. Pada jaringan yang sama, lalu lintas produksi yang terisolasi dan diproses dengan cara yang sama seperti hari ini<sup>[2]</sup>.

## III. METODE PENELITIAN

Metode Penelitian yang digunakan peneliti yaitu melakukan perbandingan performa jaringan komputer, antara jaringan komputer yang menggunakan jaringan *Software Definision Network based OpenFlow*, dan jaringan komputer yang tidak menggunakan jaringan *Software Definision Network*, dengan mengirim paket – paket data yang telah disesuaikan bebannya.

### 1. Metode Pengumpulan data

- a. Observasi, pada tahap ini Peneliti telah melakukan observasi pada perusahaan RRI Bandung untuk mendapatkan informasi mengenai jenis topologi yang digunakan pada perusahaan tersebut dan perangkat apa saja yang digunakan.

- b. Wawancara, dalam tahap ini peneliti melakukan wawancara secara langsung kepada pihak yang memiliki kewenangan dan pengetahuan mendalam mengenai topologi yang di terapkan pada perusahaan tersebut
- c. Studi *literature*, bahan atau materi penelitian yang dikumpulkan berupa literatur/referensi yang berhubungan dengan penelitian yang akan dilaksanakan.

## VI. Hasil dan Pembahasan

Penelitian ini secara umum terdiri dari 3 langkah utama, yaitu Rancangan Perbandingan dengan Simulasi, Perbandingan Manajemen Jaringan, Pengujian pada kedua Jaringan. Spesifikasi laptop yang digunakan sebagai berikut :

- Processor: Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz (4 CPUs), ~2.4GHz
- Memory: 4096MB RAM
- BIOS: Phoenix BIOS SC-T v2.2 P08RBD
- Card name: Intel(R) HD Graphics 4000
- Display Memory: 1696 MB
- Operating System : Linux UBUNTU 16.04

### 4.1 Rancangan Simulasi OpenFlow

Rancangan dari simulasi ini di buat dengan menggunakan linux sebagai system operasinya, *mininet* untuk membuat *prototype* jaringan komputer, *virtualbox* untuk membuat komputer virtual dan *OpenDayLight* sebagai protokol *OpenFlow* yang berperan untuk komunikasi antara perangkat dengan jaringan *SDN*. sebelum melakukan simulasi ada beberapa perangkat yang penulis gunakan untuk melakukan simulasi yaitu :

1. *OpenDayLight* yang telah di *install* dan di simpan pada direktori *home* dengan nama *odl*
2. *VirtualBox*
3. *Mininet* yang telah terinstall di dalam *virtualbox*
4. *Iperf* untuk melihat performa jaringan computer

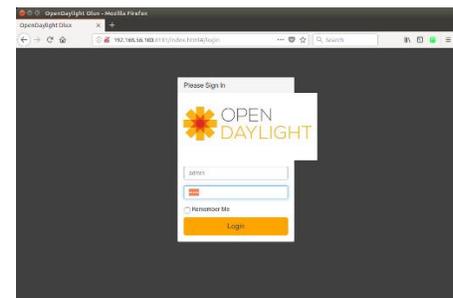
Adapun tahapan untuk melakukan simulasi ini sebagai berikut :

1. Buka terminal, lalu masukan perintah “*cd odl*” untuk berpindah direktori
2. Setelah itu maka jalankan *OpenDayLight* dengan perintah “*./bin/karaf*” maka akan tampil seperti pada gambar 1



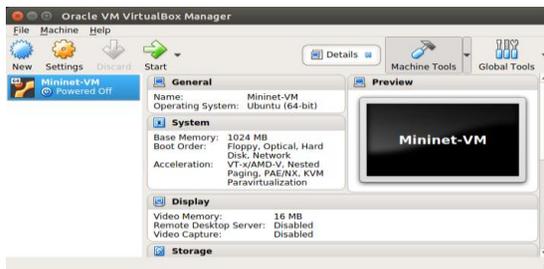
Gambar 1 *OpenDayLight*

3. Jika muncul seperti gambar di atas maka *OpenDayLight* telah berjalan, setelah itu coba lihat pada *interface OpenDayLight* pada *browser* dengan menuliskan “*192.168.56.100:8181/index.html*” maka akan muncul seperti gambar 2



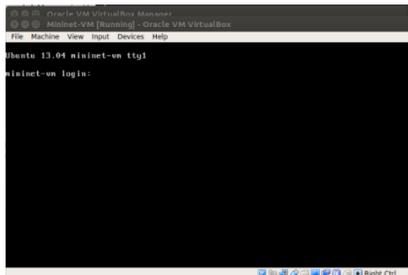
Gambar 2 *Login OpenDayLight*

4. Gambar 2 merupakan tampilan login dari *OpenDaylight*, peneliti disini menggunakan *IP 192.168.56.100* untuk *OpenDaylight*, jika *OpenDayLight* sudah berjalan, lalu kita buat *prototype* jaringan komputer dengan menggunakan *mininet* dan *virtualbox* sebagai *virtual* komputer yang baru, jadi kita disini seperti memiliki dua komputer dengan *operating system* yang berbeda.



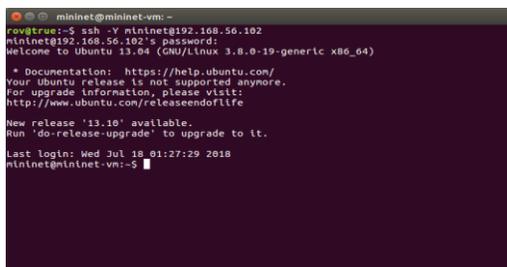
Gambar 3 VirtualBox

5. Jalankan *virtualbox* yang telah berisi *mininet* maka akan muncul gambar seperti pada Gambar 3.
6. Lalu kita klik start pada *virtualbox* untuk menjalankan system operasi yang baru, maka akan muncul seperti 4.



Gambar 4 Mininet Login

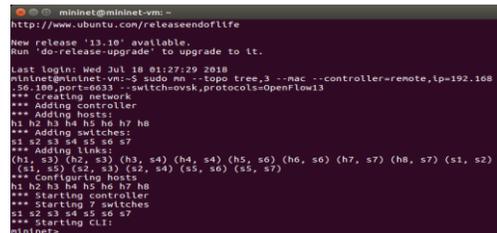
7. Gambar diatas adalah tampilan dari *mininet* yang di buka dengan *virtualbox*. Agar lebih mudah menkonfigurasi *mininet* yang ada di *virtualbox*, peneliti disini akan menggunakan SSH yang ada di linux untuk meremote *mininet* yang ada di *virtualbox*, yaitu dengan membuka terminal baru dengan menulis kan perintah seperti pada gambar 5



Gambar 5 Mininet dengan SSH

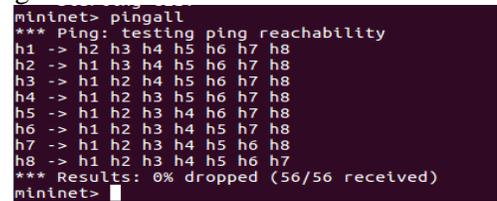
8. Seperti yang terlihat diatas disini peneliti menggunakan IP 192.168.56.102 untuk mininet, setelah kita berhasil masuk kedalam mininet dengan ssh, lalu kita buat prototype

jaringan dengan perintah “ `sudo mn -topo tree, 3 -mac -controller=remote, ip=192.168.56.100, port=663 -switch=ovsk ,protocols= OpenFlow13` ” maka akan muncul seperti pada gambar 6



Gambar 6 Pembuatan Topologi pada mininet

9. lalu kita coba lakukan ping untuk melihat konektivitas pada jaringan yang telah di buat, Jika *results 0% Dropped* maka tidak ada hambatan pada semua host seperti gambar 7



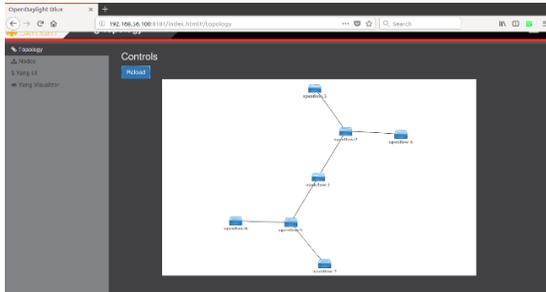
Gambar 7 Check ping pada mininet

## 4.2 Manajemen Jaringan

Manajemen jaringan sebelum menggunakan jaringan *SDN*. Ketika ada perangkat baru terhubung ke jaringan kita perlu mengkonfigurasinya dulu, kita tidak mengetahui bagaimana alur data itu berjalan, kita tidak tahu paket data apa saja yang rusak atau hilang saat terkirim dan lain sebagainya. Tetapi ketika jaringan komputer *SDN* dengan protokol *OpenFlow* ini sudah di terapkan maka semua manajemen jaringan akan terpusat hanya pada satu *controller*, hal ini sungguh sangat menguntungkan bagi kita, karna kita hanya perlu mengkonfigurasi semuanya lewat satu *controller*, kita bisa lihat *topology* apa yang sedang kita gunakan, host apa saja yang terhubung. Disini peneliti menggunakan beberapa fitur yang digunakan untuk meningkatkan manajemen jaringan dan memonitor jaringan, adapun fitur yang peneliti gunakan yaitu:

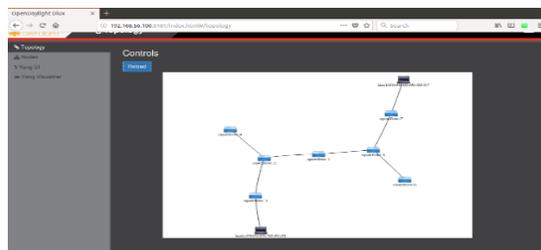
1. *Topology*

Peneliti menggunakan fitur ini untuk melihat seluruh informasi perangkat apa saja yang sedang terhubung kedalam jaringan tersebut, Seperti pada gambar berikut ini

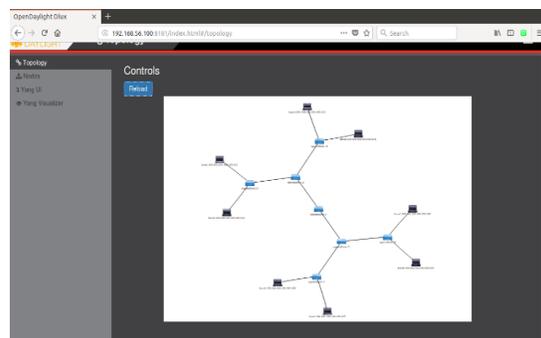


Gambar 8 Topology

Bahkan pada saat perangkat baru yang terhubung mencoba untuk melakukan ping atau pertukaran data, maka rincian informasi perangkat baru tersebut bisa langsung dilihat. Dari mulai port berapa yang menghubungkannya, berapa ip yang digunakannya dan berapa MAC address dari perangkat tersebut seperti pada gambar berikut ini



Gambar 9 Topologi test ping 1



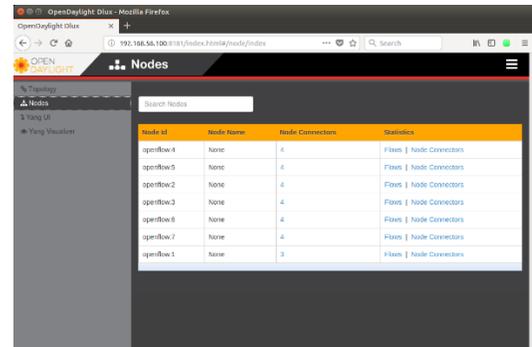
Gambar 10 Topologi test ping all host

Gambar 10 melihatkan seluruh informasi topologi beserta perangkat yang sedang terhubung, tentu ini sudah sangat memudahkan

untuk administrator jaringan dalam memonitoring jaringan komputer dan memanajemen jaringan komputer yang ada.

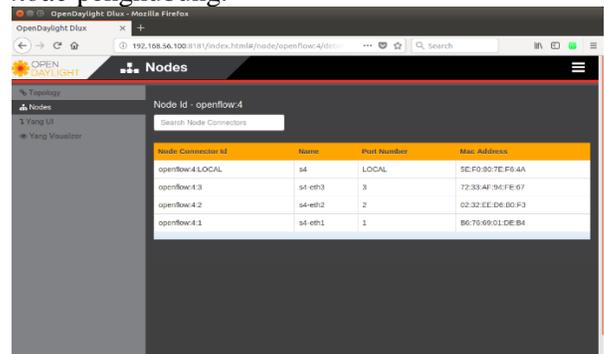
## 2. Nodes

Peneliti menggunakan fitur ini untuk memperlihatkan keseluruhan informasi lebih rinci dari tiap – tiap node yang sedang terhubung, seperti pada gambar 11



Gambar 11 Nodes

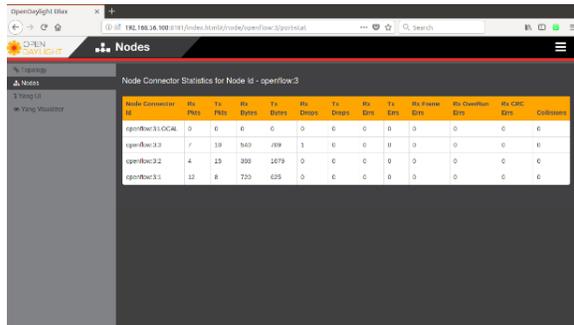
Pada gambar diatas node id berfungsi untuk memberitahu, apa node yang jadi penghubung untuk ke setiap host, node name berfungsi untuk penamaan dari tiap node yang menjadi penghubung ke tiap host, node connectors berfungsi untuk melihat berapa banyak port yang telah digunakan node penghubung, dan untuk statistics berfungsi untuk melihat lebih rinci lagi sekumpulan data informasi dari tiap node penghubung.



Gambar 12 Node Connector Id

Gambar diatas adalah informasi yang ada dalam node connector, informasi ini tentu membantu administrator jaringan dalam melihat node penghubung tersambung ke perangkat apa saja, menggunakan port apa dan informasi mac

address dari yang terhubung ke dalam *node* penghubung.

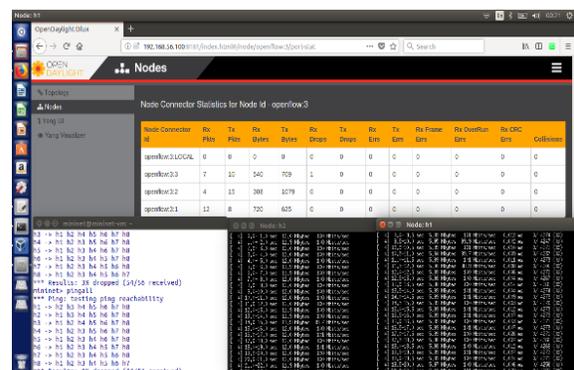


Gambar 13 Node Connector Statistic

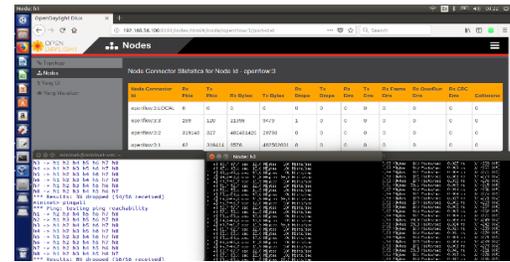
Gambar 13 adalah informasi untuk melihat alur dari paket data yang sedang berjalan, tentu informasi ini sangat di butuhkan oleh administrator jaringan untuk melihat informasi mengenai aliran paket data yang sedang berlangsung di dalam *node*, dengan begini administrator jaringan akan lebih mudah mengetahui beberapa hal yaitu :

- Jika terjadinya paket data yang rusak, hilang, atau *error*.
- Komputer yang sering melakukan pertukaran data.
- Komputer yang mencoba melakukan *flooding* kedalam jaringan.

Sebagai bukti bahwa fitur ini dapat melihat informasi dari apa yang telah dijelaskan, peneliti menggunakan *tools* Iperf untuk mengirimkan paket data sebesar 1GB dengan waktu pengiriman 100 detik, yang akan di kirimkan dari *host* pertama ke *host* ke dua seperti pada gambar 14



Gambar 14 Node Connector Statistic Sebelum di Banjiri Paket

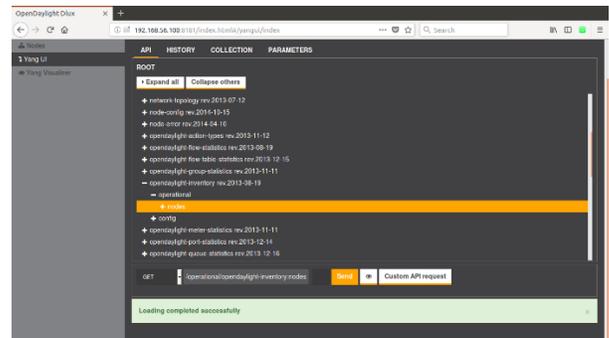


Gambar 15 Node Connector Statistic Sesudah di Banjiri Paket

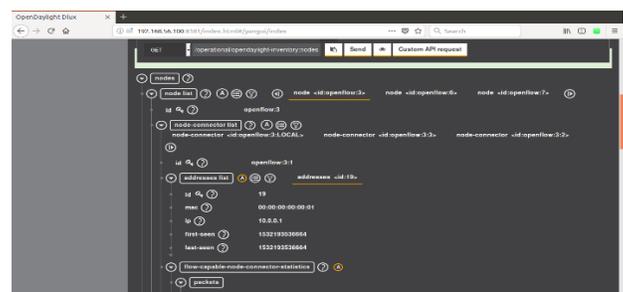
Gambar 15 adalah bukti fitur ini dapat melihat aliran paket data yang sedang berjalan.

### 3. YangUI

Peneliti menggunakan fitur ini untuk melakukan konfigurasi kepada tiap - tiap *node* yang menjadi *node* penghubung untuk mengatur jaringan yang dapat dilihat pada gambar 16



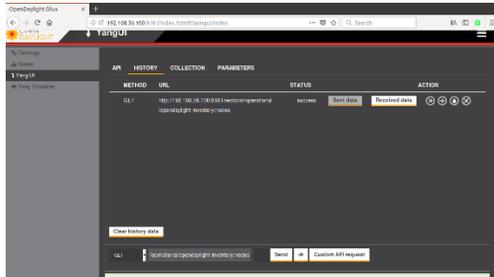
Gambar 16 YangUI



Gambar 17 YangUI Node Setting

Gambar 17 memperlihatkan keseluruhan informasi mengenai hal - hal yang berkaitan dengan *node* penghubung dan fungsi utama dari fitur ini yaitu untuk mengatur semua *configurasi* dari tiap *node* penghubung yang ada dalam jaringan. Dengan begini administrator jaringan hanya fokus mengkonfigurasi tiap *node*

penghubung dari satu titik pusat tanpa perlu repot – repot lagi mengkonfigurasi jika ada *router* atau *switch* baru yang terhubung. Karna semua konfigurasi sudah terpusat pada satu titik.



Gambar 18 YangUI History

Gambar 18 memperlihatkan keseluruhan informasi mengenai apa saja riwayat yang telah dilakukan, dengan adanya hal ini tentu ini semakin memudahkan administrator jaringan dalam manajemen jaringan, karena semua perubahan sistem jaringan komputer atau perubahan konfigurasi jaringan komputer akan tercatat semua kedalam *history*.

### 4.3 Pengujian Jaringan

Dalam pengujian jaringan komputer, peneliti akan membandingkan antara jaringan komputer yang belum menggunakan jaringan *SDN* dengan jaringan komputer yang sudah menggunakan jaringan *SDN* yang menggunakan protocol *OpenFlow*. Pada skripsi yang dibuat oleh peneliti ini, pengujian jaringan akan di bagi dua tahapan yaitu pengujian awal dan pengujian akhir

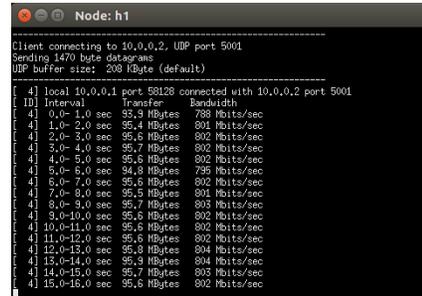
#### 4.3.1 Pengujian Jaringan Awal

Pengujian jaringan awal adalah pengujian jaringan dimana pengujian tersebut belum menggunakan jaringan *SDN*, pada tahap ini peneliti sudah menyiapkan 5 skenario dengan menggunakan *tools* yang telah di *install* yaitu *iperf*.

##### 1. Iperf

*Iperf* adalah salah satu *tool* untuk mengukur *throughput bandwidth* dalam sebuah link *network*, agar bisa dilakukan pengukuran diperlukan *Iperf* yang terinstall *point to*

*point*, baik disisi *server* maupun *client*. *Iperf* sendiri bisa digunakan untuk mengukur *performance* link dari sisi *TCP* maupun *UDP*.

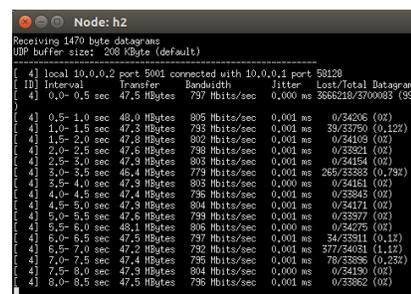


Gambar 19 Iperf sisi client

Gambar diatas adalah perintah untuk mengukur *throughput bandwidth* yang ada dalam jaringan dalam sisi *client*.

Keterangan :

- a. *-c* = *Client* atau *Host* yang mau di lihat performanya
- b. 10.0.0.1 = alamat *ip* yang di *check*
- c. *-u* = menggunakan jalur *UDP* daripada *TCP*
- d. *-i* = mengatur waktu *interval* periode *interval* antar *bandwith*
- e. 1 = waktu *interval* yg di tentukan, dalam satuan detik
- f. *-t* = waktu untuk dikirim
- g. 100= detik yang ditentukan untuk pengiriman data.
- h. *-b* = *bandwith UDP* yang dikirim dalam bit/sec
- i. 100m = ukuran yang di tentukan untuk *bandwith UDP*



Gambar 20 Iperf sisi server

Gambar diatas adalah perintah untuk menangkap data yang sedang berjalan dalam sisi *server*.

Keterangan :

- a. -s = menjalankan iperf dalam mode *server*
- b. -u = untuk membaca *UDP*
- c. -i = mengatur interval antara *bandwidth, jitter, dan loss data*
- d. l = untuk menulis semua data yang di tangkap

Pada tahap ini peneliti akan mulai membuat skenario untuk melihat performa dari jaringan komputer tanpa menggunakan *SDN*, dengan gambar sebelah kiri dari sisi *client* dan gambar sebelah kanan dari sisi *server* adapun scenario yang akan dibuat sebagai berikut :

- a. Skenario pertama dengan bandwidth 100mb
- b. Skenario kedua dengan bandwidth 250mb
- c. Skenario ketiga dengan bandwidth 500mb
- d. Skenario ke empat dengan 750mb
- e. Scenario ke lima dengan bandwidth 1000mb

Dari skenario data diatas peneliti membuat tabel untuk menyimpulkan hasil dari jaringan tanpa *SDN* seperti pada tabel berikut :

Tabel 1 Hasil Pengujian tanpa SDN

Skenario		Transfer	Bandwidth	Jitter	Lost data
1	Server	1.17 GB	100 Mb	0.016ms	0.002%
	Client	1.17 GB	100 Mb	0.016ms	0.002%
2	Server	2.71 GB	246 Mb	0.006 ms	5.3%
	Client	2.71 GB	246 Mb	0.006 ms	5.3%
3	Server	5.35 GB	459 Mb	0.003 ms	0.005%
	Client	5.35 GB	459 Mb	0.003 ms	0.005%
4	Server	8.89 GB	763 Mb	0.001 ms	0.035%
	Client	8.89 GB	763 Mb	0.001 ms	0.035%
5	Server	7.72 GB	680 Mb	0.001 ms	2.9%
	Client	7.72 GB	680 Mb	0.001 ms	2.9%

Keterangan :

- a. *Transfer*  
*Transfer* adalah pengiriman data dari komputer satu ke yang lainnya
- b. *Bandwidth*

*Bandwidth* adalah kapasitas dari suatu jalur komunikasi yang dipakai untuk mentransfer data dalam hitungan detik

c. *Jitter*

*Jitter* adalah variasi waktu dari sinyal periodik atau selisih dari *delay* paket data

d. *Lost Data*

*Lost Data* adalah data yang hilang atau rusak pada saat pertukaran data

#### 4.3.1 Pengujian Jaringan Akhir

Pengujian jaringan akhir yaitu pengujian jaringan ketika sudah menggunakan jaringan *SDN* dengan protokol *OpenFlow*. Pada tahap ini peneliti sudah menyiapkan 5 skenario sama dengan pengujian sebelumnya.

Pada tahap ini peneliti akan mulai membuat skenario untuk melihat performa dari jaringan komputer yang menggunakan *SDN*, dengan gambar sebelah kiri dari sisi *client* dan gambar sebelah kanan dari sisi *server* adapun scenario yang akan dibuat sebagai berikut :

- a. Skenario pertama dengan bandwidth 100mb
- b. Skenario kedua dengan bandwidth 250mb
- c. Skenario ketiga dengan bandwidth 500mb
- d. Skenario ke empat dengan 750mb
- e. Scenario ke lima dengan bandwidth 1000mb

Dari skenario data diatas peneliti membuat tabel untuk menyimpulkan hasil dari jaringan dengan *SDN* seperti pada tabel berikut :

Tabel 2 Hasil Pengujian dengan SDN

Skenario		Transfer	Bandwidth	Jitter	Lost data
1	Server	1.17 GB	100 Mb	0.013 ms	0%
	Client	1.17 GB	100 Mb	0.013 ms	0%
2	Server	2.90 GB	249 Mb	0.004 ms	0%
	Client	2.90 GB	249 Mb	0.004 ms	0%
3	Server	5.89 GB	506 Mb	0.003 ms	0.002%
	Client	5.89 GB	506 Mb	0.003 ms	0.002%
4	Server	8.84 GB	760 Mb	0.001 ms	0.036%
	Client	8.84 GB	760 Mb	0.001 ms	0.036%
5	Server	9.27 GB	796 Mb	0.002 ms	0.23%
	Client	9.27 GB	796 Mb	0.002 ms	0.23%
Rata - Rata	Server	5.614 GB	482.2 Mb	0.0046 ms	0.0536%
	Client	5.614 GB	482.2 Mb	0.0046 ms	0.0536%

Dapat disimpulkan dari pengujian awal dengan jaringan komputer sebelum

menggunakan jaringan SDN dan pengujian akhir dengan jaringan komputer yang telah menggunakan jaringan SDN. bahwa jaringan SDN lebih cepat dalam pertukaran data, dan lebih sedikit mengalami *loss data* dengan perbandingan hasil rata – rata sebagai berikut :

Tabel 3 Hasil Perbandingan Jaringan

Jaringan Komputer		Transfer	Bandwidth	Jitter	Lost data
Sebelum Menggunakan SDN	Server	5.168 GB	449.6 Mb	0.0054 ms	1.6484%
	Client	5.168 GB	449.6 Mb	0.0054 ms	1.6484%
Setelah Menggunakan SDN	Server	5.614 GB	482.2 Mb	0.0046 ms	0.0536%
	Client	5.614 GB	482.2 Mb	0.0046 ms	0.0536%
Selisih dari kedua Jaringan	Server	446Mb	32.2 Mb	0.0008 ms	1.5948%
	Client	446Mb	32.2 Mb	0.008 ms	1.5948%

#### IV. PENUTUP

Dari beberapa tahapan yang telah dilakukan maka peneliti menyimpulkan sebagai berikut:

1. Dengan menggunakan jaringan SDN yang menggunakan protokol OpenFlow, jaringan komputer jadi lebih mudah di konfigurasi, karena konfigurasinya terpusat pada satu *controller*.
2. Dengan menggunakan jaringan SDN yang menggunakan protokol OpenFlow, tidak bergantung lagi pada suatu vendor tertentu karna semua perangkat kontrol langsung oleh jaringan SDN.
3. Dengan menggunakan jaringan SDN yang menggunakan protokol OpenFlow, pertukaran data lebih cepat dan kemungkinan untuk terjadi *loss data* lebih kecil dari yang tidak menggunakan jaringan SDN.

#### DAFTAR PUSTAKA

- [1] Open Networking Foundation. (2012). Software-defined networking: The new norm for networks. *ONF White Paper, 2*, 2–6. <https://doi.org/citeulike-article-id:12475417>
- [2] Lara, A., Kolasani, A., & Ramamurthy, B. (2013). Network Innovation using

OpenFlow: A Survey. *IEEE Communications Surveys & Tutorials, PP(99)*, 1–20. <https://doi.org/10.1109/SURV.2013.081313.00105>

- [3] Muktafi, S. (2014). *Konsep Jaringan Dasar*.
- [4] Wahyudi, D. (2016). *Manajemen Jaring*