

Simulasi *Software Defined Network* (SDN) untuk Manajemen Jaringan Data PT. Chevron Pasifik Indonesia

Angreany Yasmin¹⁾, Linna Oktaviana Sari²⁾

¹⁾Mahasiswa Program Studi Teknik Elektro ²⁾Dosen Teknik Elektro

Laboratorium Telekomunikasi

Program Studi Teknik Elektro S1, Fakultas Teknik Universitas Riau

Kampus Bina Widya Jl. HR. Soebrantas Km 12,5 Simpang Baru, Panam,
Pekanbaru 28293

Email: angreany.yasmin@student.unr-i.ac.id

ABSTRACT

The rapid development of telecommunications requires better network technology. Computer networks are a very important and fundamental part of the development of network technology. There are still some unsatisfactory things in the conventional network architecture such as non-centralized networks that cause difficulty in managing networks on a large scale. Centralized network is needed to be able to control network traffic more flexibly. SDN (Software Defined Network) is a centralized network concept that physically separates the control and forwarding plane. In terms of capabilities, SDN is able to monitor networks more efficiently than conventional networks. This research applied a SDN network at PT Chevron. Mininet was used to design and simulate SDN networks at PT Chevron. The simulation results of network streaming audio show a delay of 12.81 ms, throughput of 0.291 mbit/sec and packet loss of 0%. The simulation results show better performance compared to conventional networks that are currently still used by PT Chevron.

Keywords: Network, SDN, Delay, Throughput, Packet loss.

1. Pendahuluan

Pada masa perkembangan teknologi seperti sekarang, banyak aspek mengalami perkembangan yang pesat, termasuk teknologi dalam bidang arsitektur jaringan. Banyak masalah serta rasa ketidakpuasan menjadi salah satu alasan mengapa bidang arsitektur jaringan juga harus berkembang.

Dengan arsitektur jaringan konvensional, masih ada beberapa hal yang tidak memuaskan seperti *high operational costs*, *difficult to manage*, selalu ada masalah dengan skalabilitas jaringan, *decentralization*, dan lain sebagainya. Dengan adanya permasalahan diatas dibutuhkan teknologi untuk mengatasinya. Salah satu teknologi baru dalam mengatasi masalah tersebut adalah *Software Defined Network* (SDN).

Software Defined Network (SDN) merupakan suatu konsep jaringan baru

dengan melakukan pemisahan antara *control* dan *forwarding plane*. Keuntungan dari jaringan SDN yaitu kemampuan untuk dapat memonitor jaringan lebih mudah baik dari sisi sumber daya, konektivitas, dan lain-lain. Selain itu, jaringan SDN dapat memaksimalkan penggunaan perangkat jaringan, misalnya optimisasi *bandwidth*, beban trafik, rekayasa trafik dan lain-lain. Komunikasi antara *data plane* dan *control plane* menggunakan suatu protokol yang dinamakan *OpenFlow*.

Penelitian tentang SDN ini sudah banyak diterbitkan didalam jurnal maupun skripsi. Pada jurnal Institute Teknologi Bandung yaitu membangun jaringan berbasis SDN pada Politeknik Caltex Riau dengan menggunakan *software* simulasi *Mininet*. (Ramadona, 2015)

Jurnal Eka Putri Aprilianingsih et al, menganalisis fail path pada jaringan SDN dengan menggunakan algoritma Dijkstra. Jaringan SDN yang dibangun pada jurnal ini mencoba membandingkan performa dari jaringan tersebut saat belum terjadinya fail path, lalu menggunakan fail path dengan memutus salah satu link dan mencari jarak terpendek untuk mencari jalur pengiriman data. (Eka Putri Aprilianingsih, 2017)

Penelitian jurnal Saputra et al dalam jurnalnya yang berjudul “Uji Performasi Algoritma Floyd-Warshall pada Jaringan Software Defined Network (SDN)”. Pengujian dilakukan dengan mengirimkan paket data, VoIp dan video dengan melihat *overhead traffic* dan QoS (*delay* dan *packet loss*). (Saputra, 2016)

Berdasarkan latar belakang diatas, pada peneltian ini akan dilakukan Analisa perancangan jaringan SDN untuk manajemen data di PT CPI. Dalam proses Analisa ini dilakukan simulasi dengan bantuan *Software Mininet*.

Mininet adalah sebuah program untuk membuat jaringan *virtual* yang realistis dalam satu buah mesin saja (*virtual machine or cloud*) dengan satu buah perintah. *Mininet* juga *software* yang bisa digunakan untuk pengembangan dan eksperimen dengan *OpenFlow* dan SDN. *Software Mininet* digunakan untuk mensimulasikan perancangan jaringan sehingga tidak perlu dilakukan pengujian langsung. Proses simulasi ini dilakukan untuk mengetahui nilai dari *Quality of Service* (QoS) dimana QoS tersebut adalah nilai *delay*.

2. Landasan Teori

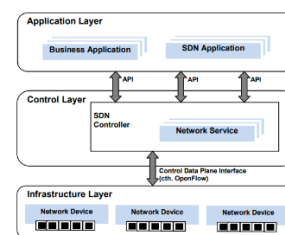
2.1. Pengenalan SDN

Software Defined Network (SDN) adalah istilah yang merujuk pada konsep/paradigma baru dalam mendesain, mengelola dan mengimplementasikan jaringan, terutama untuk mendukung kebutuhan dan inovasi di bidang ini yang semakin lama semakin kompleks. Konsep dasar SDN adalah dengan melakukan pemisahan eksplisit antara *control* dan *forwarding plane*, serta kemudian melakukan

abstraksi sistem dan meng-isolasi kompleksitas yang ada pada komponen atau sub-sistem dengan mendefinisikan antarmuka (*interface*) yang standar.

Arsitektur SDN dapat dilihat pada gambar 2.1 yang terdiri dari 3 lapis/bidang:

1. Infrastruktur (*data-plane / infrastructure layer*): terdiri dari elemen jaringan yang dapat mengatur SDN Datapath sesuai dengan instruksi yang diberikan melalui *Control-Data-Plane Interface* (CDPI).
2. Kontrol (*control plane / layer*): entitas kontrol (SDN Controller) mentranslasikan kebutuhan aplikasi dengan infrastruktur dengan memberikan instruksi yang sesuai untuk SDN Datapath serta memberikan informasi yang relevan dan dibutuhkan oleh SDN Application.
3. Aplikasi (*application plane / layer*): berada pada lapis teratas, berkomunikasi dengan sistem via *NorthBound Interface* (NBI). (Mulyana, 2015)



Gambar 2.1. Arsitektur jaringan SDN (Azwir, 2015)

Adapun karakteristik dasar dari jaringan SDN yang membedakan jaringan SDN dengan jaringan konvensional yaitu: (Goransson, P, 2014)

1. Pemisahan *Plane* (*Control* dan *forwarding Plane*)
Pemisahan antara *control plane* dan *forwarding/data plane* merupakan karakteristik dasar yang membedakan jaringan SDN dengan jaringan konvensional pada umumnya.
2. Sederhana dan Kontrol secara sentral
3. Pada jaringan konvensional setiap alat mempunyai *control plane* sendiri, sehingga setiap alat jika mempunyai protokol

dan routing yang berbeda maka perlu konfigurasi alat tersebut satu persatu. Sedangkan pada jaringan SDN, kita tidak perlu konfigurasi alat tersebut satu persatu, cukup dengan mengonfigurasi pada controller yang ada.

4. Automasi jaringan dan virtualisasi
5. Keterbukaan (*Openness*)

2.2. OpenFlow

SDN dan OpenFlow bagi yang masih baru mendengar atau mengikuti tentang SDN akan beranggapan bahwa ini adalah hal yang sama, padahal tidak.

OpenFlow merupakan teknologi yang berbeda tetapi masih dalam ranah SDN. OpenFlow merupakan alat dan inovasi yang sangat penting untuk komunikasi antara SDN data plane dan SDN control plane dan beberapa tingkah laku dari controller itu sendiri. Banyak sekali pendekatan untuk SDN, tetapi hanya OpenFlow, pendekatan non eksklusif, yang memiliki protokol yang umum untuk programming pada forwarding plane untuk SDN Switches.

2.3. Quality of Service (QoS)

Quality of Service (QoS) adalah merupakan permasalahan tentang internet yang mendiskusikan tentang kualitas suatu jaringan. Dengan ini kualitas jaringan dapat diukur menjadi suatu karakteristik jaringan yang akan dicapai.

Pengukuran QoS ini akan dilakukan di software Wireshark yang dikonfigurasi pada Mininet. Berikut parameter yang akan diuji adalah delay.

• Delay

Delay merupakan waktu tunda suatu paket data yang disebabkan oleh proses transmisi dari satu titik ke titik lain. Delay dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Dalam kasus telepon, konferensi suara, konferensi video, membutuhkan delay yang rendah sedangkan dalam file transfer atau email, delay tidak terlalu diutamakan.

(Forouzan ., 2017). Besarnya delay dapat diklasifikasikan seperti pada tabel berikut ini.

Tabel 2.1 Performansi kinerja jaringan berdasarkan delay (ITU-T G.114, 2003)

Kategori Latensi	Besar Delay
Baik	< 150 ms
Cukup	150 ms s/d 400 ms
Buruk	> 400 ms

• Packet Loss

Packet loss adalah jumlah paket yang hilang yang disebabkan oleh pembuangan paket di jaringan (*network loss*) atau pembuangan paket di gateway/terminal sampai kedatangan terakhir (*late loss*). Network loss secara normal disebabkan kemacetan, perubahan rute secara seketika, kegagalan link dan lossy link seperti suara nirkabel. Kemacetan pada jaringan merupakan penyebab utama dari packet loss.

Tabel 2.2 Standar packet loss ITU-T G.1010 untuk beberapa aplikasi (ITU-T G.1010, 2001)

Medium	Application	Degree of symmetry	PLR
Audio	Conversational Voice	Two-way	< 3% Packet Loss Ratio (PLR)
Audio	Voice Messaging	One-way	< 3% Packet Loss Ratio (PLR)
Video	Videophone	Two-way	< 1% Packet Loss Ratio (PLR)
Video	Streaming	One-way	< 1% Packet Loss Ratio (PLR)

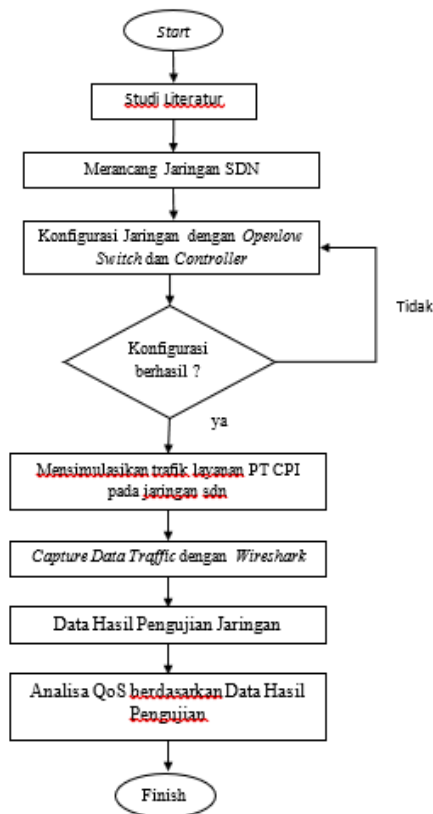
• Throughput

Throughput adalah kecepatan (*rate*) data transfer efektif yang diukur dalam Mbps. Throughput dapat dihitung dengan melihat jumlah paket yang datang terhadap yang terkirim. Throughput juga merupakan jumlah

total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

3. Metodologi

3.1. Flowchart penelitian



Gambar 3.1Diagram Alir Penelitian

Pada Gambar 3.1 dijelaskan tentang proses simulasi dan pemodelan jaringan *Software Defined Network* (SDN) untuk manajemen jaringan data PT Chevron. Kajian dilakukan melalui buku, teks pendukung, jurnal yang relevan dan menganalisa data menggunakan tulisan yang berhubungan dengan model jaringan SDN yang dikembangkan, baik dari perpustakaan, artikel, maupun internet. Berikutnya adalah memulai perencanaan simulasi model jaringan SDN di *softwaremininet*. Hal-hal yang perlu di rencanakan adalah jenis *routing protocol*, pendistribusian IP, dan jenis switch dan *controller* yang digunakan pada perangkat jaringan serta bentuk topologi jaringan yang

akan di modelkan dalam *software* simulasi. Setelah itu hasil perancangan akan di uji coba dan di analisa apakah hasilnya sudah sesuai dengan yang diinginkan. Diagram alir penelitian simulasi dan pemodelan jaringan *Software Defined Network* (SDN) untuk manajemen jaringan data PT Chevron dapat dilihat pada gambar 3.1.

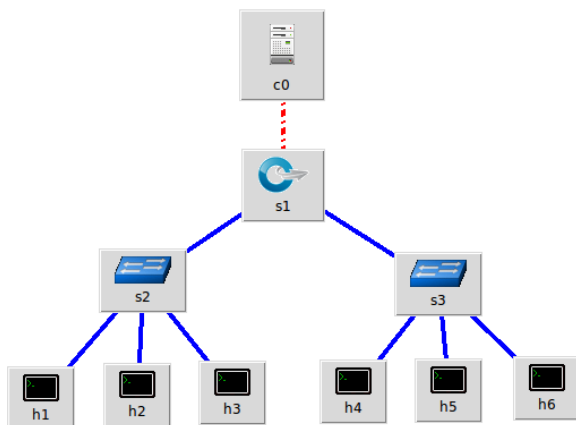
3.2. Perencanaan Model Jaringan

Perencanaan model jaringan yang akan digunakan adalah pemodelan jaringan PT Chevron. Jaringan tersebut meliputi daerah di Indonesia yaitu Pekanbaru dan Jakarta. Total yang terhubung ke jaringan Chevron ada 6 PC yang mana h1,h2,h3,h4,h5 dan h6. Perencanaan jaringan SDN yang akan disimulasikan menggunakan jaringan Chevron menggunakan semua *node* yang telah di sebutkan diatas.

Tabel3.1 Konfigurasi *Device* pada jaringan SDN UR

Device	Type	Version
c1	<i>Openflow Reference</i>	-
s1	<i>Openflow vSwitch</i>	<i>Openflow vSwitch 1.0</i>
s2-s3	<i>Legacy Switch</i>	-

Tabel 3.1 adalah konfigurasi dari *devices* yang ada pada jaringan SDN Chevron yang didesain. C1 merupakan *controller* dari jaringan SDN dengan menggunakan tipe *controller Openflow Reference* atau menggunakan *controller* jenis *openflow*. S1 merupakan *Openflow vSwitch* yang terhubung ke *controller* sehingga mempunyai kerja sama dengan *router* pada jaringan konvensional. S2 dan S3 merupakan *switch* tipe *legacy switch* atau *switch* biasa digunakan untuk membagi 1 buah *interface* ke banyak *node* pada jaringan. Selain dari yang diatas merupakan *host* yang akan saling berinteraksi pada jaringan



Gambar 3.2Jaringan SDN Chevron di *software mininet*

Pada gambar 3.2 terlihat bahwa jaringan Chevron memiliki 1router yang mana ini merupakan rancangan pada bagian wilayah Pekanbaru. Pada jaringan SDN diganti menjadi 1 buah *OpenFlow Switch* yang terhubung ke *controller*.

4. Hasil dan Pembahasan

4.1. Pengukuran QoS

Dalam melakukan pengukuran QoS membutuhkan pengiriman data. Dan data yang akan dikirimkan pada simulasi ini berupa *Audio streaming*. Dan proses yang akan dilakukan ada melakukan pengiriman paket data dari h1 ke h4 dan h2 ke h5 yang akan dilihat hasilnya QoS yang berupa *Delay*, *Packet Loss*, dan *throughput*.

```
mininet@ubuntu: ~
h5 h3 h2 h6 h4 h1
**** Starting 1 controllers
c0
**** Starting 3 switches
s2 s1 s3
No NetFlow targets specified.
No sFlow targets specified.

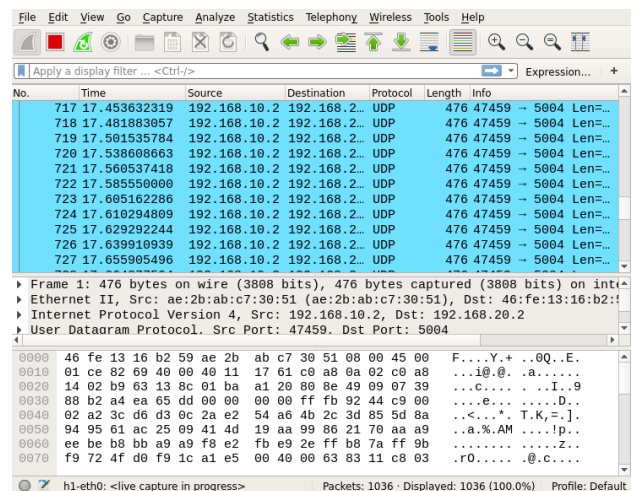
NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exiting will prevent Mininet from quitting and will prevent you from starting the network again during this session.

*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h5 -> h3 h2 h6 h4 h1
h3 -> h5 h2 h6 h4 h1
h2 -> h5 h3 h6 h4 h1
h6 -> h5 h3 h2 h4 h1
h4 -> h5 h3 h2 h6 h1
h1 -> h5 h3 h2 h6 h4
*** Results: 0% dropped (30/30 received)
mininet>
```

Gambar 4.1Tampilan CLI untuk mengecek *link* pada jaringan

Pada gambar 4.1 terlihat bahwa semua *link* yang telah dirancang terhubung, ini terlihat pada gambar semua paket yang

dikirim pada setiap *link* diterima dengan 0% data yang di *drop*.



Gambar 4.2Tampilan hasil pengukuran pada *wireshark*

Gambar 4.2 merupakan tampilan *sub-menu summary* pada *wireshark*. *Sub-menu* ini bisa diakses setelah menjalankan simulasi dan mengukur data yang lewat pada suatu jaringan. Hasil diatas merupakan hasil dari pengukuran jaringan SDN untuk mencari *delay*, *Packet loss* dan *throughput* pada jaringan.

1. Pengukuran *delay* pada jaringan SDN Chevron untuk melihat kemampuan jaringan Chevron berbasis SDN. Adapun tabel 4.1 memperlihatkan *delay* dari jaringan Chevron.

Tabel 4.1*Delay* jaringan Chevron

Jenis Traffic	Delay (ms)
Audio	12.91

Dari tabel dapat dilihat bahwa *delay* didapat pada *audio streaming* sebesar 23.84 ms, apabila dibandingkan dengan standar *delay* maka *delay* yang didapatkan *traffic* termasuk kategori sangat bagus karena lebih kecil dari 150 ms.

2. Pengukuran *packet loss* pada jaringan SDN Chevron untuk melihat kemampuan jaringan Chevron yang berbasis SDN.

Adapun table 4.2 memperlihatkan *packet loss* dari jaringan Chevron.

Tabel 4.2 *Packet loss* jaringan Chevron

Jenis Traffic	Packet loss (%)
Audio	0%

Dari tabel terlihat nilai *packet loss* dari traffic itu 0% yang berarti tidak terdapat *packet loss* pada saat pengiriman paket data.

3. Pada perhitungan *throughput* jaringan SDN Chevron dapat dilihat pada table 4.3 yang memperlihatkan hasil pengujian *throughput*.

Tabel 4.3 *Throughput* jaringan Chevron

Jenis Traffic	Throughput (Mbit/sec)
Audio	0.291

Dari tabel terlihat nilai *throughput* dari hasil pengukuran jaringan Chevron berbasis SDN yaitu sebesar 0.291 Mbit/sec.

5. Kesimpulan

Hasil simulasi yang didapat dari pengukuran QoS trafik multimedia pada jaringan SDN didapat *delay* pada jaringan SDN sebesar 12.81 ms dan dikategorikan bagus menurut standar ITU-T G.114. Sedangkan Nilai *throughput* yang dihasilkan sebesar 0.291 Mbit/sec untuk *audio streaming*. Dan Nilai *packet loss* yang didapatkan untuk *packet loss audio streaming* yaitu sebesar 0% sesuai dengan standar ITU-T G.1010.

Daftar Pustaka

Eka Putri Aprilianingsih, Rakhmadhany Primananda, Aswin Suharsono. 2017. Analisis *Fail Path* Pada Arsitektur *Software Defined Network* Menggunakan *Dijkstra Algorithm*. Jurnal Pengembangan

Teknologi Informasi dan Ilmu Komputer – Vol 1, No.3, Maret 2017.

Forouzan, Behrouz A., 2017. *Data Communications and Networking 4th Edition*. Higher Education.

ITU-T Study Group 12. 2001. *Recommendation G.1010 – End User Multimedia QoS Categories*.

ITU-T. 2003. *Recommendation ITU-T G.114 : One – Way Transmission Time*.

Mulyana, Eueung., 2015. Buku Komunitas SDN – RG. Published with GitBook.

Goransson, P., Black, C. and Culver, T., 2014. *Software defined networks: a comprehensive approach 2nd Edition*. Morgan Kaufmann.

Ramadona, S., Hidayatulloh, B.A., Siswanto, D.F. and Syambas, N., 2015, November. *The simulation of SDN network using POX controller: Case in Politeknik Caltex Riau*. In *Telecommunication Systems Services and Applications (TSSA), 2015 9th International Conference on* (pp. 1-6). IEEE..

Saputra, Ihsan Aris., M.R., Rumani., and Hertiana, Sofia Naning., 2016. Uji Performansi ALgorithm a Flyod-Warshall pada Jaringan *Software Defined Network* (SDN). Universitas Telkom.Bandung.