

# OPTIMASI MANAJEMEN JARINGAN DATA UNRI MENGGUNAKAN TEKNOLOGI JARINGAN TERDEFENISI PERANGKAT LUNAK

Ery Safrianti, Linna Oktaviana Sari, Rian Arighi Mahan  
Jurusan Elektro Fakultas Teknik Universitas Riau  
email : esafrianti@eng.unri.ac.id

*Abstrak*— Jaringan komputer adalah salah satu bagian utama dalam sistem telekomunikasi. Untuk mendukung teknologi jaringan yang handal maka jaringan terpusat dibutuhkan agar trafik jaringan dapat diatur dengan lebih mudah. Teknologi jaringan terdefinisi perangkat lunak atau lebih dikenal dengan istilah *Software Defined Network (SDN)* adalah jaringan terpusat yang menyediakan pemisahan antara *control plane* dan *data plane* di dalam sistem yang berbeda. Penelitian ini membahas optimasi manajemen jaringan Universitas Riau (UNRI) dengan SDN. Optimasi dilakukan dengan mendesain jaringan komputer UNRI kedalam bentuk SDN kemudian disimulasikan menggunakan *software mininet*. Analisis *Quality of Service (QoS)* dilakukan dari hasil pengukuran menggunakan *software wireshark*. Hasil simulasi jaringan memberikan nilai *delay* 0.506 ms, *packet loss* 0%, *throughput* 590.392 Mb/s dan *jitter* 0.093 ms. Jaringan SDN memberikan performa *delay* dan *jitter* yang lebih baik jika dibandingkan jaringan konvensional UNRI dengan nilai *delay* 13.874 ms, *packet loss* 0%, *throughput* 635.1 Mb/s dan *jitter* 2.6 ms. Disain jaringan SDN UNRI patut untuk dipertimbangkan karena memiliki nilai QoS lebih baik, *delay* dan *jitter* dibawah standar ITU dan jaringan konvensional.

*Kata Kunci*—optimasi, jaringan komputer, SDN, UNRI

## I. PENDAHULUAN

KONSEP jaringan konvensional menempatkan *data plane* dan *control plane* pada satu alat yang disebut *router*. Jika suatu *router* sudah dikonfigurasi menggunakan suatu protokol, maka akan sangat sulit untuk memodifikasi protokol tersebut karena keputusan untuk mengirim data dilakukan secara tertutup. Masalah ini dapat diatasi dengan merancang jaringan berbasis *Software Define Network (SDN)*. Konsep SDN menempatkan *data plane* dan *control plane* secara terpisah. Komunikasi antara *data plane* dan *control plane* menggunakan suatu protokol yang dinamakan *OpenFlow*.

Saat ini jaringan Universitas Riau (UNRI) menggunakan konsep jaringan konvensional dengan 4 *router*. *Router* tersebut terdiri dari 1 *router* utama, dan

3 buah *router* dibawahnya. Manajemen jaringan UNRI dilakukan oleh 4 orang, yaitu 2 orang sebagai admin dan 2 orang sebagai pembantu (*support*). Selain itu juga memerlukan 1 orang pengawas di tiap-tiap fakultas untuk memeriksa konektivitas dengan *router* utama. Adanya perbedaan jenis *router* pada jaringan UNRI menyebabkan keharusan untuk konfigurasi ulang setiap *router* bila terjadi kerusakan. Perbaikan memakan waktu 7x24 jam agar jaringan dapat bekerja kembali.

Jaringan SDN memberikan solusi untuk mengoptimasi manajemen jaringan UNRI agar dapat dilakukan secara terpusat sehingga memudahkan admin untuk mengatur dan memperbaiki kerusakan bila terjadi pada skala besar. SDN juga memberikan kemudahan untuk memperbesar jaringan UNRI dengan kemampuan untuk merubah perilaku jaringan pada masa yang akan datang.

Penelitian ini akan menganalisis Penerapan *Software Defined Network (SDN)* pada jaringan Universitas Riau yang disimulasikan dengan *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. Penggunaan *software MiniNet* yaitu 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 meliputi : *Throughput*, *Delay*, *Jitter* dan *Packet Loss*.

## II. LANDASAN TEORI

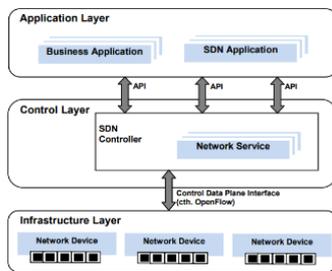
### Pengenalan SDN

*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 antar-muka (*interface*) yang standar.

Arsitektur SDN dapat dilihat pada gambar 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)* [1]

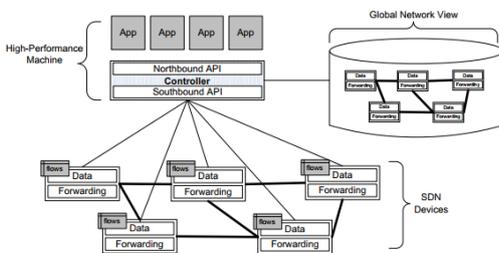


Gambar 1. Arsitektur jaringan SDN [2]

Operasi *Software Defined Networks (SDN)*

Operasi SDN dapat dilihat pada Gambar 2. Pada gambar terlihat jaringan SDN mempunyai 3 pembagian yaitu SDN devices, controller dan aplikasi. Cara cepat untuk memahami operasi SDN dapat dilihat dari bawah, dimulai dari SDN devices. Seperti pada gambar 2, SDN devices mempunyai fungsi mengarahkan untuk setiap data yang datang. SDN devices menuntun data yang datang yang mana data tersebut direpresentasikan sebagai *flow* oleh controller.

*Flow* dideskripsikan oleh 1 set paket dikirim dari satu *end point* ke *end point* lainnya yang mana *end point* tersebut biasa didefinisikan sebagai IP Address TCP-UDP port, VLAN endpoints, input port dan lain-lain.

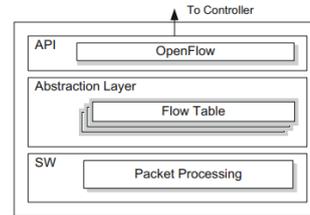


Gambar 2. Operasi SDN [3]

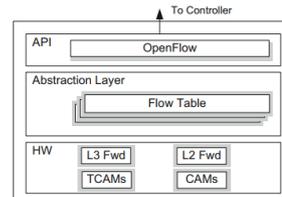
SDN Devices

SDN devices terdiri dari komunikasi API (*Applications Programmer Interface*), lapisan abstraksi (*abstraction layer*) dan fungsi pengolahan

paket (*packet-processing function*), seperti pada gambar 3. Dalam kasus *switch* virtual, fungsi pengolahan paket adalah perwujudan dalam perangkat keras untuk logika pengolahan paket (*packet-processing logic*), terlihat pada gambar 4.



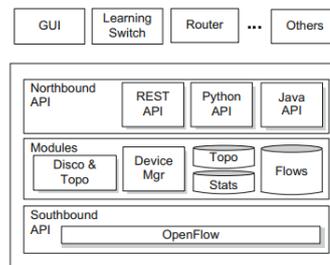
Gambar 3. SDN Software Switch Anatomy [3]



Gambar 4. SDN Hardware Switch Anatomy [3]

SDN Controller

SDN Controller menjaga bentuk dari keseluruhan jaringan, memberikan kebijakan keputusan, dan kontrol semua SDN devices yang terdiri dari infrastruktur jaringan dan menyediakan *northbound* API untuk *applications*. Controller yang mengimplementasikan kebijakan keputusan terkait *routing, forwarding, directing*, penyeimbangan beban (*load*) mengacu kepada *controller* dan *applications* yang mengatur *controller* tersebut. Ini dapat dilihat pada gambar 5.



Gambar 5. SDN Controller Anatomy [3]

SDN Applications

SDN Applications berjalan diatas SDN controller, tatap muka ke jaringan menggunakan *northbound* API controller. SDN applications sangat bertanggung jawab untuk mengelola *flow entries* yang terprogram di devices jaringan menggunakan API controller untuk mengelola aliran (*flow*). Melalui API, applications ini dapat melakukan:

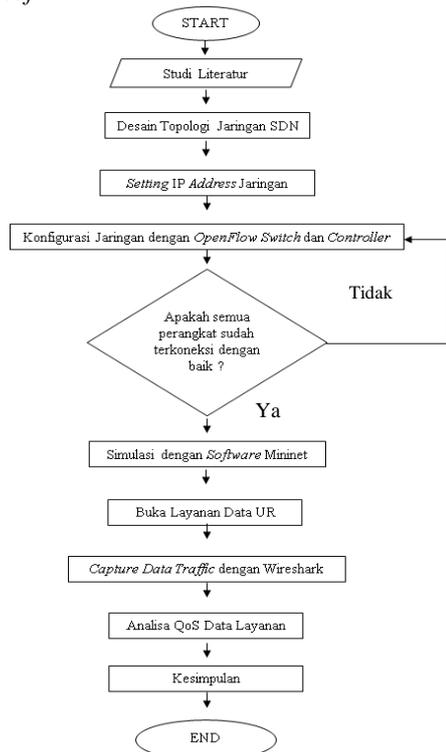
1. Konfigurasi rute aliran paket melalui jalur terbaik dari 2 buah *end point*.
2. Menyeimbangkan beban trafik melalui berbagai jalur untuk menetapkan *end points*.
3. Bereaksi saat adanya perubahan topologi jaringan seperti gagal *link*, dan adanya penambahan *device* atau jalur.

- Mengalihkan trafik dengan tujuan inspeksi, otentifikasi, segregasi, dan keamanan.

### III. METODOLOGI

#### Flowchart penelitian

Pada Gambar 6 dijelaskan tentang proses simulasi dan pemodelan jaringan *Software Defined Network* (SDN) untuk manajemen jaringan data Universitas Riau. Kajian dilakukan melalui buku-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 *software mininet*.



Gambar 6. Diagram Alir Penelitian

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 UR dapat dilihat pada gambar 6.

#### Perencanaan Model Jaringan

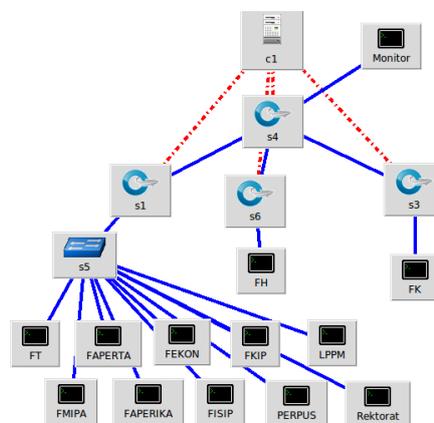
Perencanaan model jaringan yang akan digunakan adalah pemodelan jaringan Universitas Riau. Jaringan tersebut meliputi semua fakultas, pustaka dan rektorat yang terhubung ke jaringan Universitas Riau. Total yang terhubung ke jaringan UR sebanyak 9 fakultas, 1 pustaka UR, 1 rektorat, dan 1 LPPM (Lembaga Penelitian dan Pengabdian

Masyarakat). Perencanaan jaringan SDN yang akan disimulasikan menggunakan jaringan UR menggunakan semua *node* yang telah di sebutkan diatas.

Tabel 2. Konfigurasi *Device* pada jaringan SDN UR

Device	Type	Version
c1	<i>Openflow Reference</i>	-
s1-s4	<i>Openflow vSwitch</i>	<i>Openflow vSwitch 1.0</i>
s5	<i>Legacy Switch</i>	-
MONITOR - FK	<i>Host</i>	-

Tabel 2 adalah konfigurasi dari *devices* yang ada pada jaringan SDN UR yang didesain. C1 merupakan *controller* dari jaringan SDN dengan menggunakan tipe *controller Openflow Reference* atau menggunakan *controller jenis openflow*. S1-S4 merupakan *Openflow vSwitch* yang terhubung ke *controller* sehingga mempunyai kerja sama dengan *router* pada jaringan konvensional. S5 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 7 Jaringan SDN UR di *software mininet*

Pada gambar 7 terlihat bahwa jaringan UR memiliki 4 *router*, pada jaringan SDN UR diganti menjadi 4 buah *OpenFlow switch* yang masing-masing terhubung ke 1 buah *controller*. Untuk memeriksa *link* yang terhubung pada jaringan dapat menggunakan perintah *pingall* pada CLI *controller* sehingga akan terlihat seperti pada gambar 8.

### IV. HASIL DAN PEMBAHASAN

#### Pengukuran Delay

Untuk dapat mengukur *delay*, diperlukan pertukaran data pada jaringan. Agar dapat membanjiri suatu jaringan dengan data, digunakan *software* lain yang bernama *iperf*. *Iperf* memungkinkan kita untuk membanjiri suatu jaringan dengan data dan mengetahui kualitas jaringan tersebut. Dikarenakan informasi kualitas jaringan yang diberikan tidak

selengkap menggunakan *wireshark*, maka dengan menggunakan 2 *software* tersebut dapat menampilkan hasil informasi QoS yang dibutuhkan.



Gambar 8. Tampilan CLI untuk mengecek *link* pada jaringan

Pada gambar 8 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 9. Tampilan hasil pengukuran pada *wireshark*

Gambar 9. 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* jaringan.

Pengukuran *delay* pada jaringan SDN UR untuk melihat kemampuan jaringan UR berbasis SDN dan dibandingkan dengan *delay* jaringan konvensional UR. Adapun tabel 3. memperlihatkan *delay* dari jaringan UR.

Tabel 3. *Delay* jaringan UR

Standar ITU	Jaringan Konvensional	Jaringan SDN
< 150 ms	13.874 ms	0.506 ms

Pada tabel diatas terlihat bahwa nilai pada jaringan konvensional adalah 13.874 ms. Nilai *delay* jaringan SDN memiliki nilai 0.506 ms yang mempunyai nilai 27x dari jaringan konvensional UR. Bila mengacu pada standar ITU, kedua jaringan UR memiliki kualitas yang sangat baik tetapi nilai *delay* jaringan SDN jauh lebih baik.

**Packet Loss**

Nilai dari *packet loss* jaringan SDN UNRI dibagi menjadi 3 buah. Tabel 4 menampilkan hasil *packet loss* dari jaringan SDN UNRI.

Tabel 4 Nilai *packet loss* jaringan SDN UNRI

	Switch 1	Switch 2	Switch 3	Total
Packet Loss (%)	0	0	0	0

Tabel diatas memperlihatkan nilai dari *packet loss* pada 3 bagian simulasi jaringan SDN UNRI.

Selanjutnya pada tabel 5 menunjukkan hasil total dari nilai *packet loss* jaringan SDN UNRI yang dibandingkan dengan jaringan konvensional UNRI dan standar ITU-T G.1010.

Tabel 5 Perbandingan *packet loss* jaringan UNRI

Jaringan Konvensional UNRI	Jaringan SDN UNRI	ITU-T G.1010
0%	0%	1%

Pada tabel 5, kedua jaringan UNRI konvensional dan jaringan SDN UNRI mempunyai *packet loss* yang sama yaitu 0%. Ini berarti tidak ada sama sekali paket data yang dibuang pada saat pengiriman. Bila dibandingkan dengan standar ITU-T G.1010 untuk pengiriman *video* 2 arah yaitu sebesar 1%, membuat jaringan UNRI secara konvensional maupun dalam bentuk jaringan SDN memiliki jaringan yang sangat baik dan dapat diandalkan (*reliable*). Untuk melihat lebih rinci data hasil simulasi menggunakan *wireshark* dapat dilihat pada lampiran hasil pengukuran *packet loss*.

**Throughput**

Bila suatu jaringan mempunyai besar nilai pengiriman data yang dapat dikirim maka disebut dengan *bandwidth*. *Throughput* masih berkaitan dengan *bandwidth* karena nilai *throughput* tidak mungkin lebih dari nilai *bandwidth*. Ini karena *throughput* merupakan besar nilai pengiriman data yang diukur secara *realtime*. Tidak ada standar besar nilai *throughput* pada suatu jaringan. Indikator suatu jaringan yang baik adalah besarnya nilai *throughput* yang tercapai dan mendekati nilai besaran *bandwidth* yang telah ada.

Besar nilai *throughput* dapat diukur menggunakan *wireshark*. Nilai *throughput* yang telah didapat menggunakan *wireshark* dapat dilihat pada tabel 6 berikut.

Tabel 6 Nilai *throughput* jaringan SDN UNRI

	Switch 1	Switch 2	Switch 3	Total
Throughput (Mbit/sec)	523.5	46.93	19.963	590.392

Pada tabel 6 terlihat bahwa nilai *throughput* hasil simulasi menggunakan *mininet* mendekati nilai *bandwidth* pada jaringan. Untuk dapat melihat performa dari jaringan konvensional UNRI dan jaringan SDN UNRI, tabel 7 memberikan nilai dari kedua *throughput* jaringan UNRI.

Tabel 7 Perbandingan *throughput* jaringan UNRI

Jaringan Konvensional UNRI	Jaringan SDN UNRI
635.1 Mbit/s	590.392 Mbit/s

**Jitter (Delay Variation)**

*Jitter (Delay Variation)* merupakan besarnya nilai variasi dari *delay* yang terjadi pada jaringan. Semakin kecil nilai *jitter*, maka koneksi jaringan yang terbentuk mempunyai kualitas yang baik. Standar untuk nilai *jitter* mengacu pada ITU-T G.1010 dengan besar nilai *jitter* untuk jaringan *video* yaitu <1 ms.

*Jitter* pada simulasi *mininet* dapat dilakukan dengan menggunakan *iperf*. Pada *command iperf*, kita menambahkan *command -u* yang mempunyai arti mengirimkan data UDP serta mengukur *throughput* dan *jitter* pada *link* yang ada. Hasil pengukuran *jitter* untuk jaringan SDN UNRI dapat dilihat pada tabel 8.

Tabel 8 Nilai *jitter* jaringan SDN UNRI

	Switch 1	Switch 2	Switch 3	Total
<i>Jitter</i> (ms)	0.109	0.042	0.129	0.093

Pada tabel 8, nilai *jitter* pada jaringan SDN UNRI memiliki nilai yang sangat baik yaitu dibawah 1 ms. Untuk melihat nilai perbandingan *jitter* jaringan UNRI dengan SDN dan jaringan konvensional UNRI yang telah ada dapat dilihat pada tabel 9.

Tabel 9 Perbandingan *jitter* jaringan UNRI

Jaringan Konvensional UNRI	Jaringan SDN UNRI	ITU-T G.1010
2.6 ms	0.093	< 1 ms

Pada tabel 9, nilai *jitter* jaringan UNRI dengan SDN mempunyai nilai lebih baik jika dibandingkan jaringan konvensional. *Jitter* pada jaringan konvensional UNRI lebih besar nilainya jika mengacu pada standar ITU untuk pengiriman *video* 2 arah. Namun, nilai *jitter* tersebut masih dibawah 10 ms bila mengacu pada standar *cisco* untuk pengiriman data sehingga jaringan tersebut dapat diandalkan (*reliable*). Untuk melihat lebih rinci data hasil simulasi menggunakan *iperf* dapat dilihat pada lampiran hasil pengukuran *jitter*.

## V. KESIMPULAN

1. Dengan jaringan SDN UNRI, konfigurasi dilakukan secara terpusat karena adanya automasi jaringan dan virtualisasi sehingga konfigurasi perangkat jaringan dapat dilakukan selama 1 hari. Jika dibandingkan dengan jaringan konvensional yang memerlukan waktu 7 hari untuk konfigurasi semua perangkat jaringan.
2. Hasil simulasi nilai *delay* jaringan SDN UNRI menunjukkan nilai yang sangat baik yaitu 0.506 ms. Jika nilai *delay* dibandingkan dengan jaringan konvensional UNRI, maka nilai *delay* jaringan konvensional UNRI lebih tinggi dengan nilai 13.874 ms. Bila mengikuti standar ITU-T G.114, kedua jaringan UNRI memiliki kualitas yang sangat baik. Tetapi, jaringan UNRI dengan SDN menawarkan kelebihan pada kualitas *delay* yang baik dibanding jaringan konvensional UNRI.
3. Hasil simulasi nilai *packet loss* jaringan SDN UNRI menunjukkan nilai yang sangat baik yaitu 0% dari semua data yang telah dikirim. Nilai *packet loss* ini sama dengan nilai *packet loss* yang diukur pada jaringan konvensional UNRI saat ini sehingga kedua jaringan ini memberikan nilai *packet loss* yang baik. Pada standar ITU-T G.1010, nilai toleransi dari *packet loss* pada jaringan yaitu saat pengiriman *video* 2 arah yaitu 1% dan ini

merupakan standar *packet loss* yang paling kecil. Kedua jaringan UNRI mampu untuk dapat mengirimkan data tersebut.

4. Hasil simulasi nilai *throughput* jaringan SDN UNRI menunjukkan nilai yang baik yaitu 590.392 Mb/s dari 1 Gb/s yang disediakan. Sedangkan nilai *throughput* pada jaringan konvensional UNRI memiliki nilai 635.1 Mb/s.
5. Hasil simulasi nilai *jitter* jaringan SDN UNRI menunjukkan nilai yang sangat baik jika dibandingkan dengan jaringan konvensional. Bila mengacu pada standar ITU-T G.1010, maka jaringan SDN UNRI mempunyai kualitas yang sangat baik dengan nilai dibawah 1 ms yaitu 0.093 ms. Sedangkan *jitter* pada jaringan konvensional UNRI memiliki nilai 2.6 ms. Bila mengacu pada standar *cisco*, *jitter* pada jaringan UNRI memiliki kualitas yang baik untuk pengiriman data karena masih dibawah 10 ms.

## REFERENSI

- [1] Mulyana, Eueung., 2015. Buku Komunitas SDN – RG. *Published with GitBook*.
- [2] Azwir, M., 2015. SDN (*Software Defined Networking*) Konsep dan Framework Jaringan yang *Centralized, Flexible dan Programmable*.
- [3] Goransson, P., Black, C. and Culver, T., 2014. *Software defined networks: a comprehensive approach 2<sup>nd</sup> Edition*. Morgan Kaufmann.
- [4] Forouzan, Behrouz A., 2017. *Data Communications and Networking 4<sup>th</sup> Edition*. Higher Education.
- [5] ITU-T. 2003. *Recommendation ITU-T G.114 : One – Way Transmission Time*.