

**ANALISIS RESOURCE SHARING SOFTWARE-DEFINED NETWORK.
(KASUS TOPOLOGI KOMANDO RESOR MILITER 062/TARUMANEGARA)**

**ANALYSIS OF RESOURCE SHARING SOFTWARE-DEFINED NETWORK.
(TOPOLOGY CASE OF MILITARY RESORT COMMAND 062/TARUMANEGARA)**

Nigel Ghifari¹, Sofia Naning Hertiana², Nyoman Bogi Aditya Karna³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung

¹nigelghifari@student.telkomuniversity.ac.id,²

sofiananing@telkomuniversity.ac.id @telkomuniversity.ac.id, ³aditya@telkomuniversity.ac.id

Abstrak

Perkembangan *Software-Defined Network* (SDN) terus memberikan inovasi untuk kebutuhan bidang tertentu. Jaringan komunikasi militer menjadi uji coba dalam terobosan ini karena semua sistem didominasi oleh komputer yang berintegrasi dengan tiga jenis perangkat komunikasi: *wireless*, *wired* dan satelit. Cabang aplikasi SDN militer dikelompokkan berdasarkan fungsi *layer* yang digunakan dan kebutuhan pelayanan jenis militer, salah satunya adalah *resource sharing*. Penelitian ini menganalisa performansi *resource sharing* berupa pengiriman *video streaming* di Komando Resor Militer (Korem) 062/Tarumanegara. Jaringan menggunakan *custom topology* dengan jumlah *switch* berbeda sebanyak 3, 4 dan 5 buah oleh Mininet. Performansi *Quality of Service* (QoS) tiap parameter direkap oleh wireshark dengan skenario *no link down* dan *link down*. Hasil yang diperoleh adalah topologi dengan 4 *switch* direkomendasikan untuk Korem 062/Tarumanegara dengan *packet loss* = 0 - 16%, *delay* = 22,913 - 27,56 ms, *jitter* = 0,00012206 – 0,385 ms dan *throughput* = 387,163 – 464,746 Kbps.

Kata Kunci: *Software-Defined Network (SDN)*, *resource sharing*, *video streaming*, **Komando Resor Militer 062/Tarumanegara (Korem)**, *ryu*, *Quality of Service (QoS)*.

Abstract

The development of the *Software-Defined Network* (SDN) continues to provide innovations for all needs of specific fields. A military communication network is a test for this breakthrough because all systems are dominated by computers that integrate with three types of communication devices: *wireless*, *wired*, and *satellite*. Military SDN application branches are grouped based on *layer* function used and *military-type service* needs, which one of them is *resource sharing*. This research analyzes the performance of *resource sharing* in form of *video streaming* delivery at *Military Resort Command* (Korem) 062/Tarumanegara. The network used *custom topology* with a number of different switches as many as 3, 4 and 5 pieces by Mininet. The *Quality of Service* (QoS) performance from each parameter is recapitulated by wireshark with *no link down* and *link down* scenario. The results obtained topology with 4 switches are recommended for Korem 062/Tarumanegara with *packet loss* = 0 - 16%, *delay* = 22,913 - 27,56 ms, *jitter* = 0,00012206 – 0,385 ms and *throughput* = 387,163 – 464,746 Kbps.

Keywords: *Software-Defined Network (SDN)*, *resource sharing*, *video streaming*, **Military Resort Command 062/Tarumanegara (Korem)**, *ryu*, *Quality of Service (QoS)*.

1. Pendahuluan

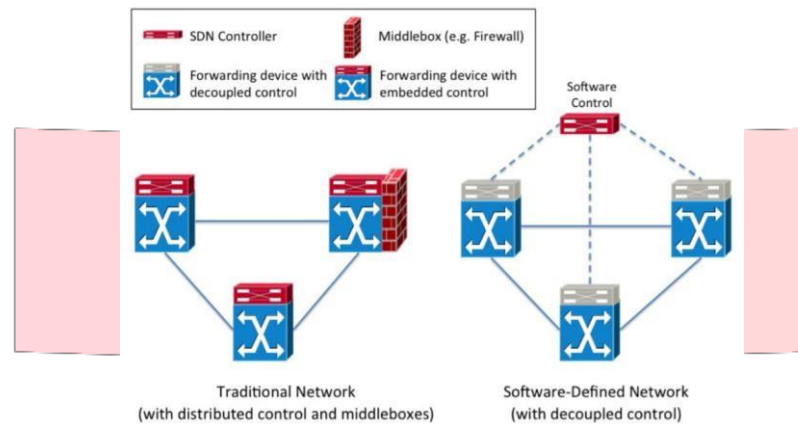
Jaringan komputer terus berkembang mengikuti terobosan sesuai keunggulan dan fungsi utama. Salah satu terobosan yang sedang berkembang saat ini adalah *Software-Defined Network* (SDN). SDN merupakan jaringan komputer berkonsep pemisahan *control plane* dan *data plane* dimana sebuah *controller* menyalurkan data informasi pada trafik jaringan yang sedang dikirimkan [1]. Keunggulan SDN antara lain memudahkan operator dan *network administrator* dalam mengelola jaringan, memberikan solusi untuk mengintegrasikan teknologi baru dan mengatasi kinerja yang buruk karena operasi yang berlebihan pada protokol *layer* atau sulitnya menyediakan layanan-layanan baru [2].

SDN dalam bidang militer belum sepenuhnya diterapkan secara menyeluruh. Tapi, masih dikerjakan secara bertahap. Spencer dan Willink menyebutkan bahwa tantangan yang harus dihadapi pada jaringan militer adalah *bandwidth* terbatas, konektivitas nirkabel yang tidak dapat diandalkan dan bergantungnya *controller* jaringan, *east-west control plane* serta integritas data pada *open network*. Penerapan SDN untuk jaringan komunikasi militer khususnya *Battlefield Networking* sebagai *policy-based management*, *resource sharing*, *monitoring and control*, perancangan, keamanan dan simulasi [3]. *Resource sharing* menjadi kebutuhan militer dalam pengiriman informasi secara langsung, salah satunya *video streaming real-time* ke tiap wilayah komando [4]. Militer mengharapakan tidak ada *delay* dan kualitas video buruk pada saat *streaming*.

2. Dasar Teori dan Metodologi

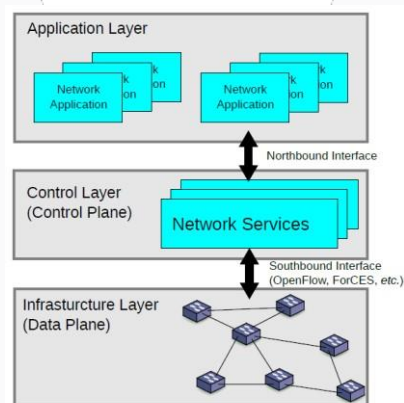
2.1 Software Defined Network

Software-Defined Network (SDN) adalah paradigma jaringan komputer yang sedang berkembang terkait pemisahan *control plane* dan *data plane*. Tujuannya untuk membentuk arsitektur jaringan secara terpusat dan mudah diprogram. Perlunya sebuah protokol untuk bisa berkomunikasi antara *control plane* dan *data plane* seperti ForCES, OpenFlow, dll. Dibandingkan dengan jaringan konvensional semua komponen *data plane* seperti *router* atau *switch* masih bersatu dalam perangkat jaringan [5].



Gambar 1. Perbandingan Jaringan Tradisional dengan SDN.

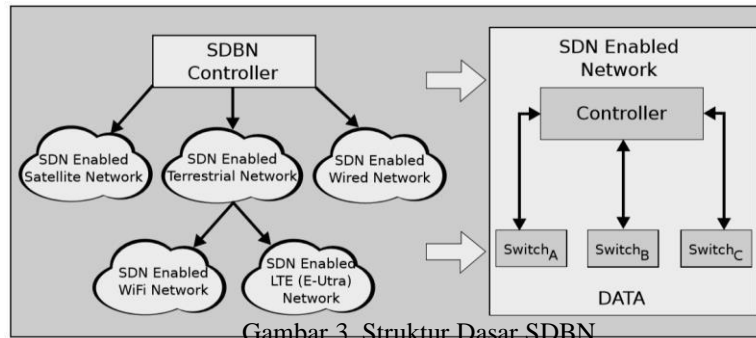
Dari perbandingan kedua jaringan tersebut memiliki arsitekturnya masing-masing. Untuk arsitektur SDN terdiri dari 3 layer, yaitu *application layer* yang menyediakan layanan *network application*, seperti membantu *control layer* dalam konfigurasi jaringan, keamanan dan pengelolaan. Kemudian ada *control layer* atau *control plane* adalah layer yang bertanggung jawab dalam mengatur data yang diteruskan dari *infrastructure layer*. Dan terakhir *Infrastructure layer/Data plane* berfungsi sebagai penerus data pada *router* atau *switch*. Untuk bisa berkomunikasi ke *control layer*, perlunya protokol seperti OpenFlow [6].



Gambar 2. Arsitektur SDN.

2.2 Jaringan Komunikasi Militer

Pemapanan jaringan komunikasi militer dimulai pengertian teknologi militer [7] bahwa: "Teknologi militer adalah ilmu yang menggambarkan dan menjelaskan bagaimana teknologi mempengaruhi aktivitas militer dalam semua tingkatan dan profesi perwira. Teknologi militer berdasarkan pada beberapa bidang studi beragam dari perpaduan pemahaman profesi militer yang diturunkan dari ilmu masyarakat dengan ilmu dasar dan superstruktur serta dinamika pasokan teknik." (Axberg, 2006). Umumnya, jaringan militer berintegrasi dengan perangkat lebih rumit dibandingkan jaringan komputer, tergantung dengan penerapannya. Penggabungan teknologi jaringan militer dengan SDN menjadi arsitektur *Software-Defined Battlefield Network (SDBN)* [4] dimana *plane* yang terbentuk sama seperti arsitektur SDN, namun perbedaannya dari sistem jaringan perangkat komunikasi. Struktur dasar SDBN dipandang dalam gambar 3. bahwa beberapa *controller* menjadi tanggung jawab besar dalam merangkul beragam perangkat komunikasi dan bekerja sekaligus dalam protokol *southbound* berbeda.



Gambar 3. Struktur Dasar SDBN.

2.3 Resource Sharing

Resource sharing adalah teknik jaringan komputer dengan tujuan untuk membuat semua program, perangkat dan menyediakan data untuk semua orang dalam jaringan tanpa terpengaruh oleh lokasi *resource* dan pemakai [8]. Keunggulan dari *resource sharing* pada jaringan komunikasi militer bahwa: "...*resource sharing* sangat penting untuk efisiensi penggunaan aset, mengukur gambar operasional umum yang komprehensif..." [3]. Mengenai konsep *resource sharing* bahwa hubungan *client* dengan *server* wajib merespon setiap permintaan untuk bisa mengirimkan informasi. *Requests and replies* berjalan dari *client* mengirimkan permintaan pesan ke *server* melalui jaringan untuk disampaikan. Ketika sampai, *server* melaksanakan proses pengumpulan pesan secara utuh dan menjawab kembali isi pesan tersebut ke *client*, begitu juga sebaliknya

2.4 Real-Time Protocol

Saat melakukan *streaming* media, perlunya sebuah protokol untuk bisa menjalankannya. Salah satunya adalah *Real-Time Protocol* (RTP), yaitu protokol jaringan yang mengirimkan audio dan video tersinkronisasi melalui IP address. RTP digunakan sebagai *streaming media* seperti *video streaming* dan *telephony* dengan bantuan *transport protocol* berupa *User Datagram Protocol* (UDP). Alasan RTP sebagai protokol *video streaming* karena perlunya protokol yang bisa mengabaikan *data error*, *connectionless* dan tidak ada *flow control* seperti pada *Transport Control Protocol* (TCP). RTP tidak terlepas dengan *Real-Time Control Protocol* (RTCP) sebagai *feedback* yang melaporkan pesan eror dalam pengirim dan penerima. Pesan tersebut berupa statistik *packet loss* dan *jitter*. Biasanya, RTCP terkait banyaknya pengguna (*host*) *video streaming* [9].

2.5 OpenFlow

OpenFlow merupakan seperangkat protokol SDN dimana berfungsi mengatur paket yang masuk ke *switch*. OpenFlow digadai sebagai *controller* penting di *control layer* SDN karena posisinya berada di *southbound interface* dimana saat paket diteruskan, OpenFlow langsung berinteraksi sebagai protokol standar. Peran OpenFlow untuk mengatasi *traffic flow* secara langsung dari beragam perangkat, baik fisik maupun visual dengan Open vSwitch (OVS). Tidak hanya itu, OVS sebagai *switching* dasar dalam jaringan *hardware* sehingga *switch* maupun *router* dapat dihubungkan secara langsung [10].

2.6 Mininet

Mininet merupakan *emulator* dalam virtualisasi jaringan SDN. Keunggulannya adalah pembuatan *prototype* cepat dari jaringan yang ditentukan *software*, pengujian topologi kompleks tanpa memasang jaringan fisik dan beberapa pengembang konkuren bekerja secara mandiri pada topologi yang sama [11].

2.7 Ryu

Ryu adalah *controller* SDN yang menyediakan *Application Programming Interfaces* (API) untuk memudahkan *developer* dalam mengembangkan pengaturan jaringan baru dan aplikasi. *Controller* ini mendukung beragam protokol seperti OpenFlow versi 1.0 s.d. 1.5 [12]. Ryu memiliki beragam fungsi dalam *program python*, salah satunya *Spanning Tree Protocol* (STP) yaitu protokol *link management* yang menyediakan redundansi jalur sambil mencegah *loop* yang tidak diinginkan [13]. Protokol ini (STP: IEEE 802.1D) digunakan untuk memastikan tidak ada *loop* di *layer 2*, tepatnya di *bridge*, yaitu perangkat yang menghubungkan beberapa segmen dalam suatu *domain* tabrakan. Untuk *switch* dianggap sebagai *bridge* itu sendiri [14]. Cara kerjanya dengan mengatur *port* masing-masing *switch*, sehingga paket *Bridge Protocol Data Unit* (BPDU) saling bertukar informasi dan menentukan *port* yang tersedia atau tidak. Ada beberapa peran pada STP berdasarkan *cost* untuk mencapai *root bridge* [15]:

1. *Root port*

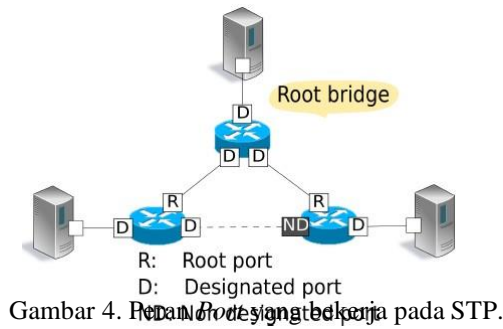
Port dengan *cost* terkecil dimana menerima paket BPDU dari *root bridge*.

2. *Designated ports*

Semua *non-root ports* yang masih diperbolehkan untuk meneruskan *traffic* pada jaringan. Hanya satu *designated port* yang diperbolehkan pada tiap *segment*.

3. *Non designated ports*

Port yang dikonfigurasi sebagai pemblokir untuk mencegah terjadinya *traffic loop*.



2.8 Quality of Service

Quality of Service (QoS) adalah mekanisme jaringan aplikasi atau layanan dapat beroperasi sesuai harapan. Salah satu kegunaannya yaitu meningkatkan performansi untuk aplikasi yang sensitif terhadap delay seperti suara (voice) atau video. Berikut adalah parameter dalam QoS:

1. Packet Loss

Packet loss adalah kondisi saat paket dikirim tidak dapat diterima secara menyeluruh akibat corrupt packet dan hardware failure. Untuk menentukan packet loss dapat dituliskan

$$Pa = \frac{\text{Paket yang hilang}}{\text{Paket yang dikirim}} \times 100\% \quad (1).$$

2. Delay

Delay adalah waktu yang dibutuhkan sebuah paket untuk mengirim sampai ke tujuan. Faktornya bisa karena jarak, media fisik dan durasi yang lama. Untuk delay dapat dirumuskan

$$t_2 - t_1 \quad (2)$$

dimana t_2 merupakan waktu paket ke-2 dan t_1 adalah waktu paket ke-1.

3. Jitter

Jitter adalah variasi delay yang diakibatkan panjang antrian dalam waktu pengolahan data, penyempitan bandwidth dan waktu penyusunan ulang data yang diterima. Jitter dapat menyebabkan sampling di sisi penerima menjadi tidak tepat sasaran, sehingga informasi menjadi rusak serta delay antrian pada router atau switch. Jitter dapat dituliskan

$$\sum_{i=1}^n (t_i - t_{i-1}) \quad (3)$$

dimana total variasi delay merupakan jumlah selisih tiap nilai delay yang dirumuskan sebagai berikut

$$\sum_{i=1}^n (t_i - t_{i-1}) = (t_2 - t_1) + (t_3 - t_2) + \dots + \quad (4).$$

4. Throughput

$$(t_i - t_{i-1})$$

Throughput adalah jumlah data yang berhasil diterima selama durasi tertentu.

$$\frac{\text{Jumlah data}}{\text{Waktu}} \quad (5).$$

Tabel 2.1 QoS Standarisasi TIPHON [16][17].

Kategori	Throughput	Packet Loss	Delay	Jitter
Sangat bagus	> 1200 Kbps	0%	< 150 ms	0 ms

Bagus	700 s/d 1200 Kbps	1 s/d 3%	150 s/d 300 ms	0 s/d 75 ms
Sedang	338 s/d 700 Kbps	4 s/d 15%	300 s/d 450 ms	75 s/d 125 ms
Buruk	0 s/d 338 Kbps	16 s/d 25%	> 450 ms	125 s/d 225 ms

3. Desain Sistem

3.1 Kondisi Nyata pada Korem

Terkait kondisi nyata jaringan komunikasi pada militer masih sangat terbatas. Pertemuan Pertama ASEAN *Our Eyes* (AOE) yang dilaksanakan pada tanggal 29-30 Oktober 2018 di Jakarta membuah lima hasil musyawarah, salah satunya adalah “Indonesia mengusulkan penggunaan 3 sistem jaringan untuk AOE yaitu telepon/fax, *video teleconference* dan *e-mail* dengan sistem yang akan disiapkan oleh Indonesia” [18]. *Video conference* digunakan sebagai alternatif dalam sistem laporan jika telepon rusak dan biaya sangat mahal. Keterbatasan jumlah komputer harus menghadapi sulitnya akses dalam pengiriman *e-mail* dan hanya bisa berintegrasi sesuai divisi militer. Keuntungan utamanya adalah *user* cenderung menyerap informasi lebih baik jika mereka melihat dan mendengarnya langsung [19].

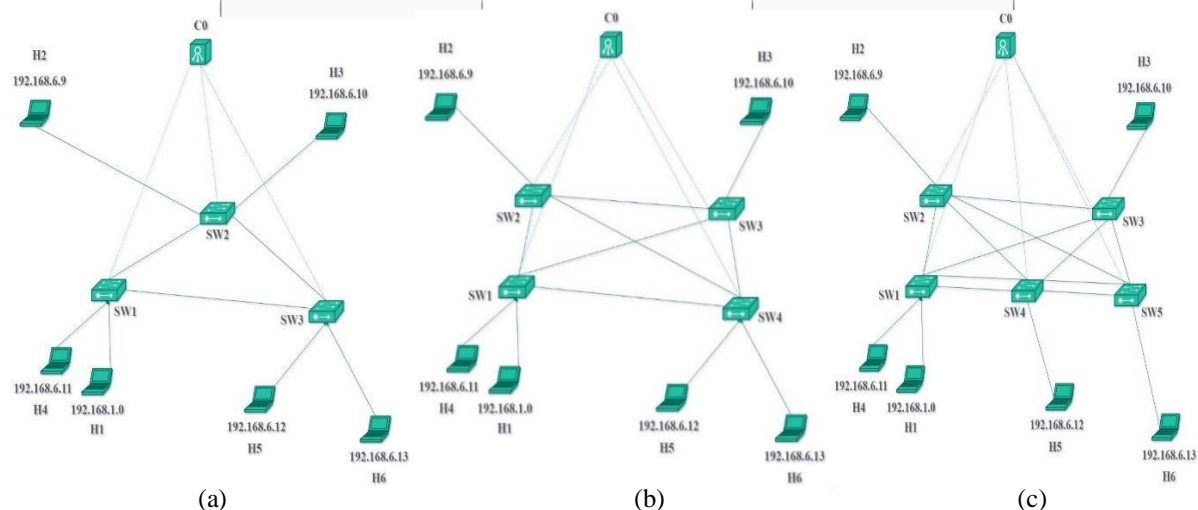
Proses pelaporan pada militer bersifat komando. Pada penelitian ini dibuat untuk komunikasi militer antara Komando Distrik Militer (Kodim) dan Komando Resor Militer (Korem), karena untuk satu kodam terdiri lebih dari tiga korem. Selain itu, masalah sumber daya manusia mengenai perkembangan telekomunikasi belum diterapkan dalam sistem komputer, khususnya pada tingkat Bintara Pembina Desa (Babinsa) dan Korem. Untuk pengiriman laporan, hanya bisa dilakukan secara satu arah antara kodim dan korem.

3.2 Topologi Jaringan dan Pengujian

Proses membuat topologi menggunakan *python script* seperti pada gambar 5 yang terdiri dari satu buah *controller*, 6 buah *host* serta beragam *switch* sebanyak 3, 4, dan 5 buah. *Host* 1 merupakan markas Korem 06/Tarumanegara terletak di kota Garut, Jawa Barat dan 5 *host* lainnya adalah kodim yang membawahnya [20]:

1. Kodim 0609/Kabupaten Bandung
2. Kodim 0610/Sumedang
3. Kodim 0611/Garut
4. Kodim 0612/Tasikmalaya
5. Kodim 0613/Ciamis

Topologi yang terbentuk berupa *custom topology* dimana tujuannya semua kodim bisa terhubung ke korem. IP address tiap kodim mengikuti nomor kode wilayah komando yang sudah ditentukan. Open vSwitch adalah *switch* yang akan digunakan. Setelah itu, *controller* ryu bekerja menggunakan *Spanning Tree Protocol* (STP) serta didukung OpenFlow versi 1.3 untuk mencegah terjadinya *traffic loop* pada jaringan. Tujuan pengujian jumlah *switch* beragam untuk mengetahui pengaruhnya terhadap QoS saat *video streaming* berlangsung. Setiap kali pengiriman dipantau oleh *wireshark* dari *host* antar kodim dan korem. Dari konsep *resource sharing*, korem berlaku sebagai *client* penerima video dan kodim sebagai *server* untuk mengirimkan video. Pengujian menggunakan STP dengan diberlakukannya dua skenario yaitu *no link down* dan *link down*. Skenario *no link down* dilakukan untuk mengetahui proses kerja STP yang bisa mengatur rute *port switch* secara otomatis dan pencegahan *traffic loop*. *Link down* dilakukan pada salah satu *link network* untuk mengetahui perhitungan ulang STP pada tahap *link up* dan apakah berpengaruh terhadap proses *video streaming* berlangsung. Pemutusan *link* dilakukan minimal satu kali dan dipilih sesuai *link network* saat komunikasi kodim dan korem berlangsung. Pengujian berlaku untuk semua topologi dengan *switch* sebanyak 3, 4, dan 5 buah.



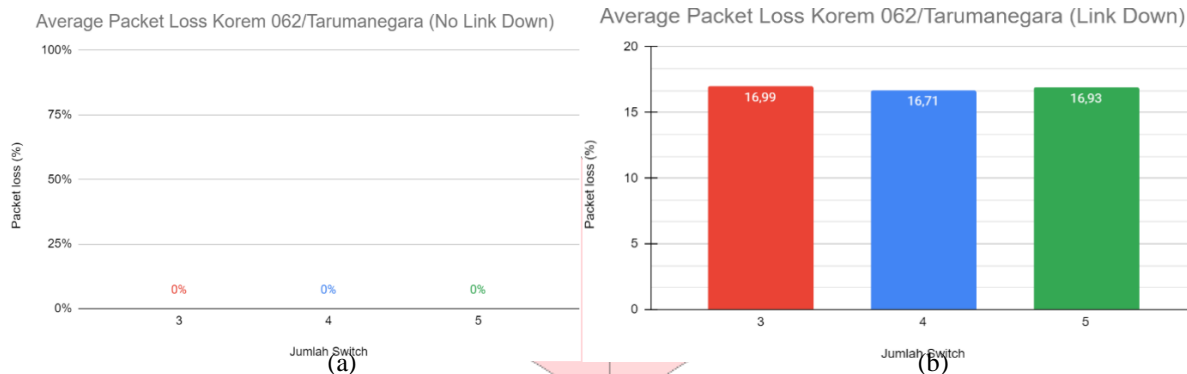
Gambar 5. Topologi Jaringan Korem 062/Tarumanegara dengan (a) 3 Switch, (b) 4 Switch, dan (c) 5 Switch.

4. Hasil dan Analisa

4.1 Packet Loss

Packet loss merupakan banyaknya data yang gagal saat dikirimkan dari *host* akibat paket rusak (*corrupt*). Pengujian parameter ini dengan mengirimkan video berdurasi selama 3 menit 12 detik dan mengaktifkan *Spanning*

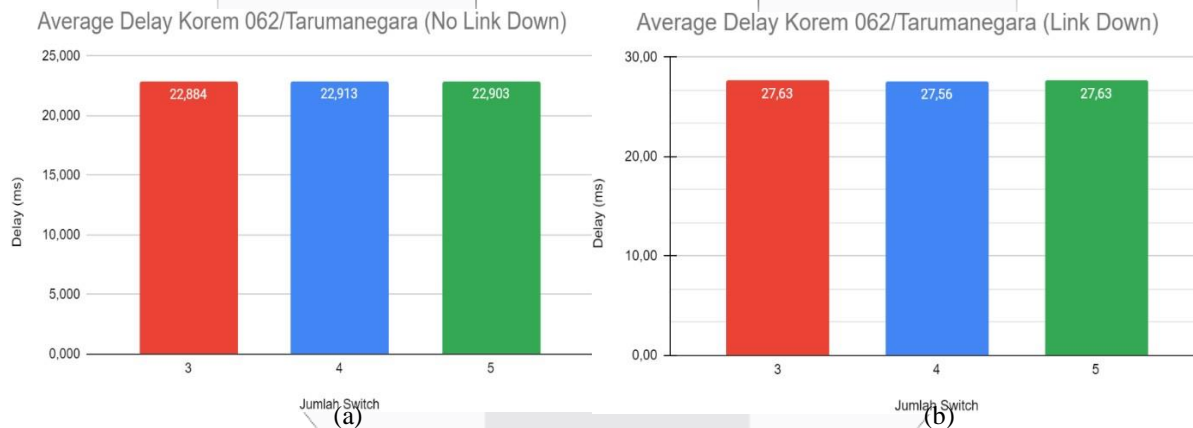
Tree Protocol (STP) pada ryu controller dengan dukungan OpenFlow 1.3. Pada skenario **no link down** dilakukannya STP untuk pencegahan *loop traffic* dan tidak harus mengkonfigurasi switch satu per satu. Nilai *packet loss* menunjukkan bahwa presentase dikategorikan **sangat bagus** karena berdasarkan standarisasi TIPHON, nilai *packet loss* sebesar 0%. Hasil tersebut disimpulkan untuk pengiriman video tidak ada yang hilang atau terlewatkan. Skenario selanjutnya diberlakukan **link down** dengan sengaja untuk menguji perhitungan ulang STP supaya tercapainya *link up*. Nilai rata-rata *packet loss* menunjukkan bahwa presentase bernilai lebih dari 16% pada semua jumlah *switch*, sehingga kategori QoS berstandar TIPHON termasuk **buruk**. Dampak tersebut berpengaruh pada tampilan video pada *host* korem mengalami *lagging* atau video tidak diterima dengan baik.



Gambar 4.1 Perbandingan *Packet Loss* Rata-Rata (a) *No Link Down* dan (b) *Link Down*.

4.2 Delay

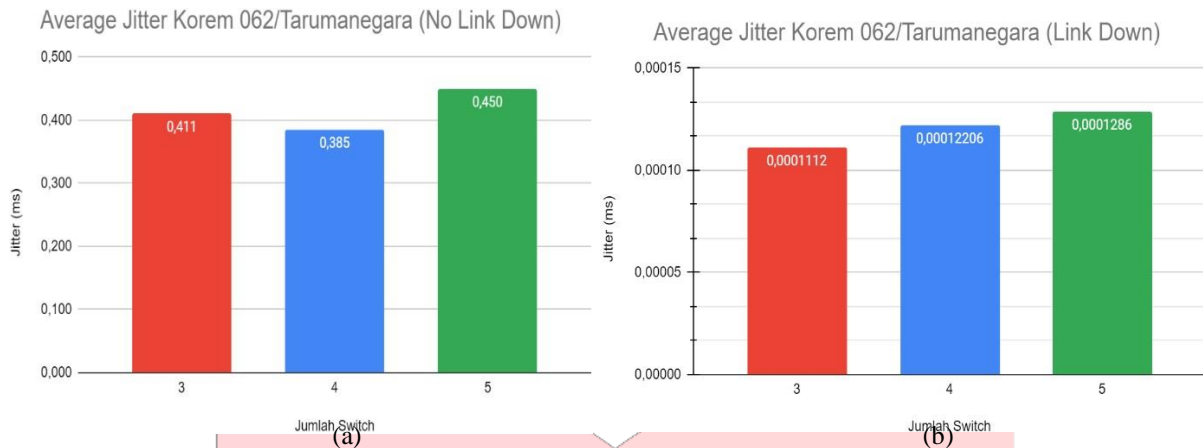
Delay adalah waktu yang dibutuhkan data untuk mengirim dari masing-masing *node* yang melaluinya. Pada skenario **no link down**, nilai *delay* rata-rata mencapai 22,884 – 22,913 ms dan dikategorikan **sangat bagus** karena berdasarkan standarisasi TIPHON, nilainya kurang dari 150 ms. Untuk skenario **link down** menunjukkan bahwa rata-rata *delay* sebesar 27,56 – 27,63 ms pada semua jumlah *switch* dan dikategorikan **sangat bagus**. Dari kedua skenario tersebut menunjukkan bahwa tidak ada pengaruh *delay* pada jumlah *switch*, karena hasilnya murni saat *video streaming* berlangsung.



Gambar 4.2 Perbandingan *Delay* Rata-Rata (a) *No Link Down* dan (b) *Link Down*.

4.3 Jitter

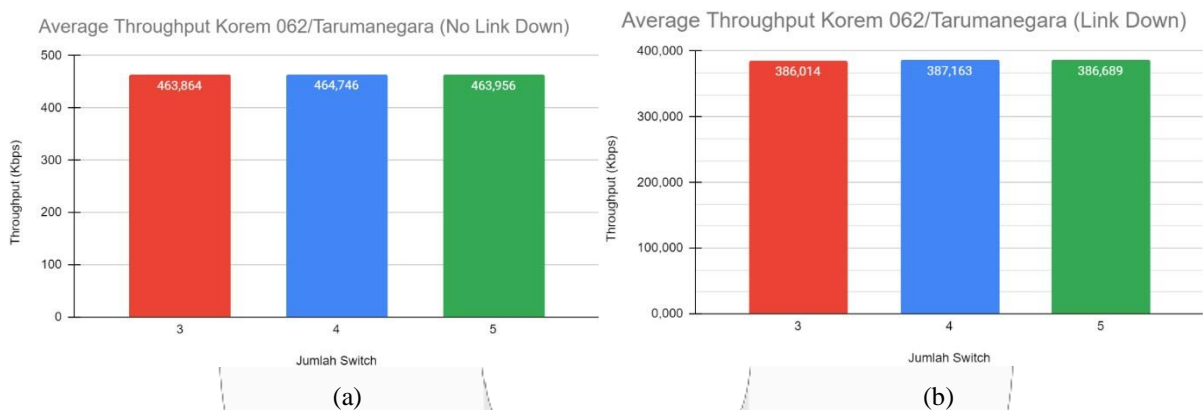
Jitter merupakan variasi *delay* dari paket diterima. Skenario **no link down** diperoleh *jitter* rata-rata mencapai 0,385 – 0,45 ms dan dikategorikan **bagus** karena berdasarkan standarisasi TIPHON, nilainya berkisar 0 - 75 ms. Dari diagram (a) menunjukkan bahwa perubahan *jitter* mengalami penurunan untuk *switch* sebanyak 4 buah karena kondisi jaringan stabil dan banyaknya rute yang bisa ditempuh pada *switch*. Sama seperti *delay*, nilai *jitter* merupakan hasil murni saat *video streaming* berlangsung. Pada skenario **link down**, hasil *jitter* rata-ratanya sangat kecil sebesar 0,0001112 – 0,0001286 ms dan dikategorikan **bagus**. Dari hasil diagram (b), semakin banyak *switch* nilai *jitter* semakin bertambah karena pada saat *link down* terjadi, paket mengalami keterlambatan dan sebagiannya rusak. Jaringan yang tidak stabil atau gangguan menjadi salah satu faktor saat paket harus melakukan pengiriman ulang.



Gambar 4.5 Perbandingan *Jitter* Rata-Rata (a) *No Link Down* dan (b) *Link Down*.

4.4 Throughput

Throughput adalah kecepatan (*rate*) transfer data sesungguhnya saat dikirimkan. Untuk skenario *no link down*, nilai *throughput* rata-rata mencapai 463,864 – 464,746 Kbps dan dikategorikan **sedang** berdasarkan standarisasi TIPHON. Dan pada skenario *link down*, nilai QoS berstandar TIPHON *throughput* termasuk **sedang** dengan 386,014 – 387,163 Kbps pada semua jumlah *switch*. Dari kedua skenario tersebut menunjukkan bahwa tidak ada pengaruh *throughput* pada jumlah *switch*, sama seperti *delay*. Hal tersebut dikarenakan hasil murni *video streaming* berlangsung dan perekapan tidak hanya UDP saja, melainkan beberapa protokol lainnya seperti STP, *Real-Time Control Protocol* (RTCP), *Address Resolution Protocol* (ARP), dll.



Gambar 4.7 Perbandingan *Throughput* Rata-Rata (a) *No Link Down* dan (b) *Link Down*.

4.5 Proses Video Streaming pada Kodim dan Korem

Pada saat perekapan, protokol *video streaming* berupa UDP dimana proses media *stream* secara *real-time* dapat dikirimkan melalui IP address [9]. Karena pengiriman menggunakan *Real-Time Protocol* (RTP), semua paket data dilakukan *decoding* dalam UDP port ke RTP sehingga menunjukkan MPEG TS (*Moving Picture Experts Group Transport Stream*). Terjadinya format tersebut karena pada *stream output* menggunakan metode RTP/MPEG Transport Stream di VLC Media Player.


```

asus@asus-E2025A: ~/ryu
File Edit View Search Terminal Help
packet in 2 00:00:00:00:00:02 33:33:00:00:00:02 1
[STP][INFO] dpid=0000000000000001: [port=1] DESIGNATED_PORT / LEARN
[STP][INFO] dpid=0000000000000001: [port=2] DESIGNATED_PORT / LEARN
[STP][INFO] dpid=0000000000000001: [port=3] DESIGNATED_PORT / LEARN
[STP][INFO] dpid=0000000000000001: [port=4] DESIGNATED_PORT / LEARN
[STP][INFO] dpid=0000000000000003: [port=1] DESIGNATED_PORT / LEARN
[STP][INFO] dpid=0000000000000003: [port=2] DESIGNATED_PORT / LEARN
[STP][INFO] dpid=0000000000000003: [port=3] ROOT_PORT / LEARN
[STP][INFO] dpid=0000000000000003: [port=4] NON_DESIGNATED_PORT / LEARN
packet in 2 00:00:00:00:00:03 ff:ff:ff:ff:ff:ff 2
packet in 2 00:00:00:00:00:03 ff:ff:ff:ff:ff:ff 2
packet in 2 00:00:00:00:00:03 ff:ff:ff:ff:ff:ff 2
packet in 2 00:00:00:00:00:03 ff:ff:ff:ff:ff:ff 2
packet in 2 00:00:00:00:00:03 ff:ff:ff:ff:ff:ff 2
packet in 2 00:00:00:00:00:03 ff:ff:ff:ff:ff:ff 2
[STP][INFO] dpid=0000000000000001: [port=1] DESIGNATED_PORT / FORWARD
[STP][INFO] dpid=0000000000000001: [port=2] DESIGNATED_PORT / FORWARD
[STP][INFO] dpid=0000000000000001: [port=3] DESIGNATED_PORT / FORWARD
[STP][INFO] dpid=0000000000000001: [port=4] DESIGNATED_PORT / FORWARD
[STP][INFO] dpid=0000000000000003: [port=1] DESIGNATED_PORT / FORWARD
[STP][INFO] dpid=0000000000000003: [port=2] DESIGNATED_PORT / FORWARD
[STP][INFO] dpid=0000000000000003: [port=3] ROOT_PORT / FORWARD
[STP][INFO] dpid=0000000000000003: [port=4] NON_DESIGNATED_PORT / BLOCK

```

Gambar 10. Proses Konfigurasi Ulang STP saat Mengalami *Link Down*.

Proses dilakukannya *link down* pada skenario berpengaruh pada saat *streaming* berlangsung. Keuntungan STP yang dapat mengatur *routing switch* secara otomatis, hal itu termasuk melakukan *link up* jika terjadi *link down*. Proses konfigurasi ulang STP dilakukan saat tidak menerima paket BPDU selama 20 detik (secara *default*). Ketika *link down*, *port* mengalami tahap *LISTEN* dan *LEARN* untuk mengetahui *switch* manakah yang harus dituju. *FORWARD* terjadi ketika tujuan *port switch* sudah terbentuk kembali dan tahap *BLOCK* untuk memblokir jalur yang tidak dilalui [15]. Saat STP melakukan konfigurasi ulang, tampilan *video streaming* mengalami *lagging* pada korem. Namun kodim terus melakukan pengiriman dalam kondisi *link down*, sehingga menyebabkan nilai *packet loss* dan *jitter* bervariasi.



Gambar 11. Tampilan Video pada (a) *Host Korem* dan (b) *Host Kodim*.

5. Kesimpulan

Dari hasil pengujian dan analisis *resource sharing Software-Defined Network (SDN)* pada Korem 062/Tarumanegara dapat diambil kesimpulan sebagai berikut:

1. Penggunaan *ryu controller* lebih mudah digunakan dengan beragam program *python*, termasuk *Spanning Tree Protocol (STP)*. Penggunaannya sangat efektif dalam menentukan *port switch* tanpa harus mengkonfigurasi *traffic flow* secara manual. Selain itu, STP dapat digunakan untuk *link up* jika terjadi *link down* saat paket data dikirimkan.
2. Mengenai tujuan analisa parameter *QoS video streaming* dengan standarisasi TIPHON memiliki nilai yang berbeda. Dua skenario berbeda menghasilkan kategori masing-masing serta faktor penyebabnya.
 - 2.1 Pada pengujian *packet loss*, untuk skenario *no link down* digolongkan **sangat bagus** dengan presentase 0%. Komunikasi *streaming* sukses dikirimkan tanpa ada paket rusak atau hilang. Pada *link down* termasuk **buruk** dimana nilai rata-ratanya pada semua *switch* sebesar 16,71 – 16,99%. Hal tersebut karena dilakukannya *link down* untuk melihat proses *link up* pada STP *ryu controller*. Hasilnya terlihat di *video streaming* yang mengalami *lagging* atau hilangnya paket data saat dikirimkan ke korem.
 - 2.2 Pada pengujian *delay*, untuk kedua skenario digolongkan **sangat bagus** dengan nilai rata-rata sebesar 22,884 – 27,63 ms. Hasilnya tidak berpengaruh pada jumlah *switch* dimana hasil perekaman dilakukan saat *video streaming* berlangsung tanpa penambahan atau pengurangan dari alat atau sistem.

- 2.3 Pada pengujian *jitter*, untuk kedua skenario digolongkan **bagus** dengan nilai rata-rata sebesar 0,0001112 – 0,45 ms. Dari kedua skenario tersebut, perubahan *jitter* mengalami perubahan karena kestabilan jaringan, banyaknya rute yang bisa ditempuh pada *switch*, dan perlunya pengiriman ulang paket. Akibat pada skenario *link down*, proses *video streaming* mengalami *lagging* dan berpengaruh terhadap *packet loss*.
 - 2.4 Pada pengujian *throughput*, untuk kedua skenario digolongkan **sedang** dengan nilai rata-rata 386,014 – 464,746 Kbps. Hasil tersebut menunjukkan bahwa tidak ada pengaruh *throughput* pada jumlah *switch*, sama seperti *delay*. Kestabilan jaringan dan hasil murni saat *video streaming* berlangsung dengan beragam protokol lainnya menjadi akibat tercapainya nilai tersebut, tidak kurang atau lebih.
3. Dari hasil diagram parameter QoS, topologi dengan *switch* sebanyak 4 buah direkomendasikan untuk jaringan Korem 062/Tarumanegara.

Referensi

- [1] T. Bhandarkar, K. Bhuwarka, M. Khare, and K. Nanmaran, "A Survey On Application Software Defined Network," *Int. J. Pure Appl. Math.*, vol. 115, no. 6, pp. 553–559, 2017.
- [2] I. Ummah and D. Abdillah, "Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking," *Indones. J. Comput.*, vol. 1, no. 1, pp. 95–106, 2016, doi: 10.21108/indojc.2016.1.1.20.
- [3] V. Gkioulos, H. Gunleifsen, and G. K. Weldehawaryat, "A Systematic Literature Review on Military Software Defined Networks," *Futur. Internet*, vol. 10, no. 9, 2018, doi: 10.3390/fi10090088.
- [4] J. Nobre, D. Rosario, C. Both, E. Cerqueira, and M. Gerla, "Toward Software-Defined Battlefield Networking," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 152–157, 2016, doi: 10.1109/MCOM.2016.7588285.
- [5] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014, doi: 10.1109/SURV.2014.012214.00180..
- [6] W. Braun and M. Menth, "Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices," *Futur. Internet*, vol. 6, no. 2, pp. 302–336, 2014, doi: 10.3390/fi6020302.
- [7] J. Sigholm, "Secure Tactical Communications for Inter-Organizational Collaboration," p. 102, 2016, doi: 0000-0002-4376-9800.
- [8] A. S. Tanenbaum, *Computer Networks*, Fourth Edi. New Jersey: Pearson Education, Inc., 2003.
- [9] D. Austerberry, *The Technology of Video and Audio Streaming*, 2nd ed., vol. 110, no. 9. Massachusetts: Focal Press, 2005.
- [10] T. D. Nadeau and K. Gray, *SDN: Software Defined Network*, First Edit. California: O'Reilly Media, Inc., 2013.
- [11] Available: <https://www.opennetworking.org/mininet/> [Access on 26th March 2020]
- [12] Available: <https://osrg.github.io/ryu/> [Access on 26th March 2020].
- [13] I. Irawati and M. Nuruzzamanirridha, "Spanning Tree Protocol Simulation Based on Software Defined Network Using Mininet Emulator," *Commun. Comput. Inf. Sci.*, vol. 516, no. July 2018, pp. 395–403, 2015, doi: 10.1007/978-3-662-46742-8..
- [14] G. A. Donahue, *Network Warrior*, 2nd ed. Sebastopol, CA: O'Reilly Media, Inc., 2011.
- [15] R. project Team, *Ryu SDN Framework*. 2014. Available: <https://book.ryu-sdn.org/en/Ryubook.pdf> [Access on 5th November 2020].
- [16] ETSI, *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)*, 1999. Available: https://www.etsi.org/deliver/etsi_tr/101300_101399/101329/02.01.01_60/tr_101329v020101p.pdf [Access on 9th December 2020].
- [17] ETSI, *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)*, 1998. Available: https://www.etsi.org/deliver/etsi_tr/101300_101399/101329/01.02.05_60/tr_101329v010205p.pdf [Access on 12th December 2020].
- [18] A. Armawi and D. Wahidin, "Eksistensi TNI dalam Menghadapi Ancaman Militer dan Nir Militer Multidimensional di Era Milenial," *WIRA Ed. Khusus HUT TNI ke 74 tahun 2019*, pp. 1–60, 2019. Available: <https://www.kemhan.go.id/wp-content/uploads/2019/12/wiraedsus2019-web.pdf>. [Access on 11th September 2020].
- [19] Available: <https://science.howstuffworks.com/how-military-video-conferencing-works.htm/printable> [Access on 12th September 2020, 11:04:56 WIB].
- [20] Available: https://id.wikipedia.org/wiki/Komando_Resor_Militer_062 [Access on 11th September 2020].