

PENGARUH *FAILOVER* PADA JARINGAN *SOFTWARE-DEFINED NETWORK* DAN KONVENSIONAL

Riffat Hasan Saputra, Alif Subardono
Departemen Teknik Elektro dan Informatika
Universitas Gadjah Mada
riffat.hasan.s@mail.ugm.ac.id, alif@ugm.ac.id

Abstract – *The growing use of the Internet resulted in the number of agencies that use two or more internet gateway paths or so-called Internet Service Providers (ISPs) to maintain network availability. Using two or more gateway paths allows break-up of connections on one of the gateway paths. This study aims to determine the effect of failover on SDN and conventional networks that are integrated with load balancing method of ECMP on MikroTik RB951-2n router device. The controller used on the SDN network is OpenDaylight. Testing is done by doing failover test which then data of bandwidth traffic from both network architecture recorded using Graphing tool from MikroTik. For average traffic efficiency without failover influence on conventional network better with difference of 9.8%. And with the difference in efficiency of 0.5% menunjukkan that the effect of failover on the SDN network is still better than the conventional network.*

Keywords-- *Software-Defined Network, Gateway, Load Balancing, Failover.*

Intisari – Berkembangnya penggunaan internet mengakibatkan banyaknya instansi yang menggunakan dua atau lebih jalur *gateway* internet atau yang biasa disebut *Internet Service Provider (ISP)* untuk menjaga ketersediaan jaringan. Dengan menggunakan dua atau lebih jalur *gateway* memungkinkan terjadinya putus koneksi pada salah satu jalur *gateway*. Penelitian ini bertujuan mengetahui pengaruh *failover* pada jaringan SDN dan konvensional yang diintegrasikan dengan *load balancing* metode ECMP pada perangkat *router* MikroTik RB951-2n. Kontroler yang digunakan pada jaringan SDN adalah *OpenDaylight*. Pengujian dilakukan dengan melakukan uji *failover* yang kemudian data trafik *bandwidth* dari kedua arsitektur jaringan dicatat menggunakan *tools Graphing* dari MikroTik. Untuk rata-rata efisiensi trafik tanpa pengaruh *failover* pada jaringan konvensional lebih baik dengan selisih 9,8%. Dan dengan selisih efisiensi sebesar 0,5% menunjukkan bahwa pengaruh *failover* pada jaringan SDN masih lebih baik dibanding dengan jaringan konvensional.

Kata Kunci—*Software-Defined Network; Load Balancing; ECMP; Failover.*

I. PENDAHULUAN

Perkembangan teknologi layanan internet beberapa tahun terakhir ini mengalami perkembangan yang pesat, khususnya pada jaringan yang memiliki tingkat kompleksitas rendah. Hal ini mengakibatkan meningkatnya layanan akses internet pada jaringan tersebut, yang membuat penggunaan dua atau lebih jalur *gateway* internet sangat diperlukan untuk menjaga ketersediaan jaringan [1]. Sejalan dengan itu, kendala yang harus dihadapi guna menjaga kinerja jaringan agar tetap optimal tentunya juga mengalami peningkatan. Banyaknya permintaan dari pengguna yang menyebabkan *overload* pada salah satu jalur koneksi yang diakibatkan pembagian beban trafik yang tidak berimbang dan putusnya jalur koneksi pada salah satu jalur *gateway* merupakan beberapa masalah yang muncul dari penerapan dua jalur *gateway* atau lebih.

Load balancing yang mempunyai fungsi sebagai pembagi dan penyeimbang beban trafik pada dua atau lebih jalur koneksi, merupakan salah satu solusi yang tepat guna mengatasi kendala tersebut. Terdapat beberapa metode yang tepat untuk menangani masalah pada skala jaringan tersebut, di antaranya adalah ECMP, Nth, dan PCC. Metode tersebut merupakan metode yang paling umum digunakan pada penerapan *load balancing*. Metode ECMP merupakan metode *load balancing* terbaik di antara dua metode lainnya untuk menangani pembagian beban secara

merata pada dua jalur *gateway* dan juga handal terhadap efek *failover*.

Software-Defined Network (SDN) yang merupakan sebuah teknologi yang mempunyai konsep pemisahan antara sistem kontrol (*control plane*) dan sistem *forwarding* (*forwarding plane*), memungkinkan jaringan untuk dapat menyederhanakan manajemen dan skalabilitas jaringan yang diwujudkan melalui implementasi sederhana dari penambahan komponen dan layanan jaringan [2]. Dalam penelitian ini kontroler SDN yang digunakan adalah *OpenDaylight Controller*. *Controller* tersebut memiliki keunggulan dibanding *controller* lain yaitu *up-to-date*, mudah digunakan, menggunakan antarmuka Java, menggunakan Maven dan kerangka OSGi [3].

Pengaruh *failover* pada jaringan SDN dan jaringan konvensional yang dihasilkan dari penerapan *load balancing*, memungkinkan memiliki hasil kinerja yang berbeda. Penelitian ini diharapkan menghasilkan keluaran yang dapat mengetahui sejauh mana pengaruh *failover* pada arsitektur konvensional dan arsitektur SDN.

II. TEORI PENDUKUNG

A. Jaringan Komputer

Jaringan Komputer merupakan kumpulan dari beberapa PC (*Personal Computer*) atau peripheral yang saling terhubung melalui media transmisi (melalui kabel atau nirkabel) dan melakukan akses bersama terhadap suatu

resource. Secara lebih sederhana, jaringan komputer dapat diartikan sebagai sekumpulan komputer beserta mekanisme dan prosedurnya yang saling terhubung dan berkomunikasi. Komunikasi yang dilakukan oleh komputer tersebut dapat berupa transfer berbagai data, instruksi, dan informasi dari satu komputer ke komputer yang lain [4].

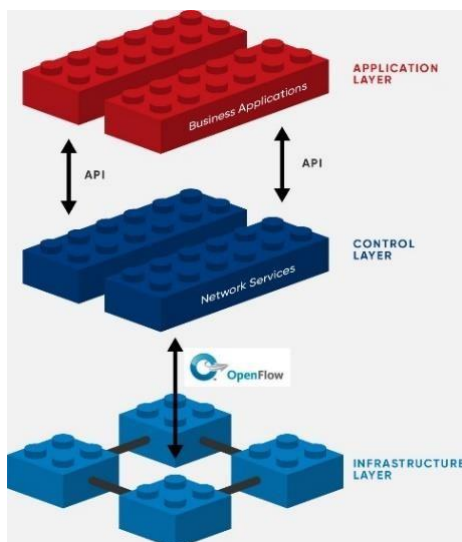
Dibangunnya sebuah sistem jaringan komputer memiliki tujuan agar pengiriman data dan informasi dari pengirim ke penerima dapat dilakukan dengan cepat dan tepat serta meminimalisir kesalahan pada media transmisi atau media komunikasi tertentu [5].

Manfaat dengan dibangunnya sebuah jaringan komputer dapat diperoleh beberapa keuntungan, diantaranya.

1. Berbagi Sumberdaya
2. Media Komunikasi
3. Integrasi Data
4. Keamanan Data
5. Manajemen Sumberdaya
6. Pengembangan dan Pemeliharaan

B. Software-Defined Network

Software-Defined Network adalah sebuah arsitektur yang mana dinamis, mudah diatur, hemat, dan mudah diadaptasi, membuatnya ideal untuk kebutuhan *bandwidth* tinggi yang merupakan sifat dinamis dari aplikasi saat ini. Arsitektur ini memisahkan *control plane* dan *forwarding plane* yang memungkinkan kontrol jaringan menjadi terprogram langsung dan infrastruktur yang mendasarinya untuk diabstraksikan untuk aplikasi dan layanan jaringan. Protokol *OpenFlow* adalah elemen dasar untuk membangun solusi SDN [6].



Gambar 1 Arsitektur *Software-Defined Network*
(Sumber Gambar : <https://www.opennetworking.org/>)

Terlihat pada Gambar 2.1, menunjukkan bahwa SDN memiliki tiga lapis/bidang:

a. Lapisan Aplikasi

Lapisan yang berada paling atas, memiliki fungsi untuk menyediakan *interface* untuk pembuatan program

aplikasi yang mengatur *network requirement* dan *network behavior* sesuai yang diinginkan. Sehingga dapat mengoptimalkan jaringan secara baik dan fleksibel.

b. Lapisan Kontrol

Entitas kontrol (*SDN Controller*) yang mentranslasikan kebutuhan aplikasi dengan infrastruktur dengan memberikan instruksi yang sesuai untuk *SDN Datapath* serta memberikan informasi yang relevan dan dibutuhkan oleh *SDN Application*.

c. Lapisan Arsitektur

Terdiri dari elemen jaringan yang dapat menerima instruksi dari *Control Plane*. *Interface* antara *Control Plane* dan *Data Plane* disebut *South Bound Interface* (SBI), atau *Control-To-Data-Plane Interface* (CDPI).

C. OpenFlow

OpenFlow (OF) adalah salah satu standar SDN pertama. Semula OF menetapkan protokol komunikasi pada lingkungan SDN yang memungkinkan kontroler SDN untuk dapat langsung berinteraksi dengan *forwarding plane* dari perangkat jaringan seperti *switch* dan *router*, baik secara fisik maupun *virtual* (*hypervisor-based*), menjadikan jaringan dapat beradaptasi pada perubahan kebutuhan bisnis [7]. *Openflow* adalah protokol SDN yang paling banyak digunakan. Hal ini dikarenakan *Openflow* memungkinkan peneliti untuk mengembangkan bahkan merancang protokol yang baru [8].

D. OpenDaylight Controller

OpenDaylight dapat dijalankan diberbagai macam produksi *network environments*. *OpenDaylight* dapat mendukung *framework* kontroler modular, tetapi dapat menyediakan dukungan untuk SDN standar lain dan protokol yang akan datang. *OpenDaylight* mengekspos membuka *northbound* APIs, yang digunakan oleh aplikasi. Aplikasi ini menggunakan kontroler untuk mengumpulkan informasi mengenai jaringan, algoritma yang berjalan untuk melakukan analisis, dan kemudian menggunakan *OpenDaylight* untuk membuat *rules* baru di seluruh jaringan. *OpenDaylight* diimplementasikan hanya di *software*, dan disimpan di dalam *Java Virtual Machine* (JVM). Ini berarti *OpenDaylight* dapat dijalankan pada perangkat keras dan OS yang mendukung Java. Untuk hasil terbaik, *OpenDaylight* disarankan menggunakan *Linux distribution* terbaru dan setidaknya JVM 1.7 [9].

E. Load Balance

Load balance adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi [10].

Dalam *load balance* di MikroTik, hal-hal yang perlu diperhatikan dalam mengatur *load balance* adalah *static route*, *policy route*, *firewall mangle*, dan *firewall src-nat*. *Static route* dan *policy router* mengatur *uplink flow*, yaitu kebijakan *routing* yang akan dilalui paket yang telah

ditandai, sedangkan *firewall mangle* adalah penandaan paket sebelum masuk *routing* [11].

1. *Equal-Cost Multi-Path* (ECMP)

Equal-Cost Multi-Path dikenal sebagai metode *load balancing* yang paling sederhana dan lebih cocok digunakan pada jaringan dengan tingkat kompleksitas rendah. Pembagian beban dalam metode ini menggunakan pengaturan perbandingan langsung antara beban yang diterima oleh setiap *gateway*. Tujuan penerapan algoritma ini yaitu untuk dimungkinkannya penggunaan lebih dari satu jalur *gateway* digunakan sebagai *exit interface* dan mendapatkan pembagian beban yang merata antar setiap *gateway* dalam membagi *traffic* yang ada. Dalam penggunaannya metode ini juga mendukung penggunaan *routing* protokol baik statis maupun dinamis [12].

ECMP yang merupakan “*persisten per-connection load balancing*” atau “*per-src-dst-address combination load balancing*” hal ini membuat begitu salah satu *gateway unreachable* atau terputus, *check-gateway* akan langsung menggunakan *gateway* yang masih aktif saja, sehingga bisa mendapatkan efek *failover*. Ketika sesuatu memicu *routing table flushing* dan *cache* ECMP terhapus. Koneksi akan dikoneksikan sekali lagi ke *gateway* dan dalam kasus dua *gateway* ada 50% kemungkinan bahwa trafik akan melalui *gateway* lain. *Routing table flushing* dapat dipicu dua hal yaitu perubahan *routing table* untuk protokol *routing* dinamis dan setiap 10 menit *routing table* akan terhapus untuk alasan keamanan. Jadi meskipun tidak ada perubahan *routing table*, koneksi dapat berpindah ke *gateway* lain setiap 10 menit [13].

F. *Bandwidth*

Bandwidth adalah nilai hitung atau perhitungan konsumsi transfer data telekomunikasi yang dihitung dalam satuan *bit* per detik atau yang biasa disingkat bps yang terjadi antara komputer *server* dan komputer klien dalam waktu tertentu dalam sebuah jaringan komputer. *Bandwidth* sendiri akan dialokasikan ke komputer dalam jaringan dan akan mempengaruhi kecepatan transfer data pada jaringan komputer tersebut sehingga semakin besar *bandwidth* pada jaringan komputer makan semakin cepat pula kecepatan transfer data yang dapat dilakukan oleh klien maupun *server* [1] [1].

G. *Failover*

Failover adalah teknik untuk membagi beban trafik di jaringan yang terlalu besar dan memindahkannya secara otomatis apabila pada jaringan tersebut mengalami kendala dalam koneksi, dalam hal ini jaringan internet bisa terputus koneksi [14]. Aplikasi teknik *failover* biasanya digabungkan dengan teknik *load balancing*. Penggunaan *failover* ditujukan untuk membangun jaringan komputer yang handal [15].

III. METODE PENELITIAN

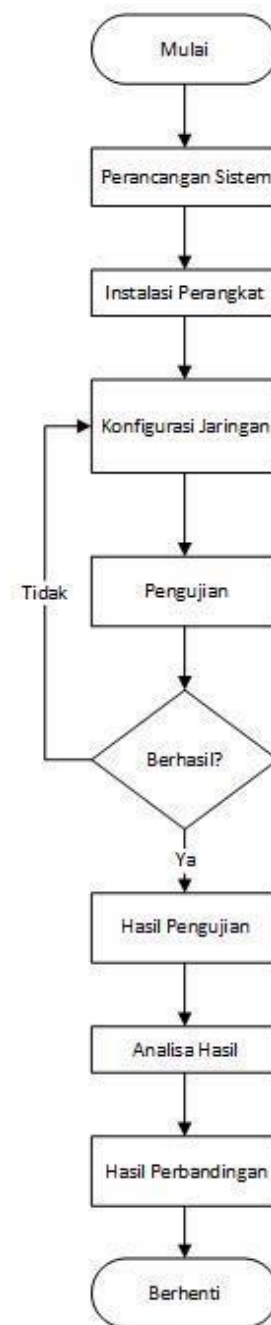
A. Alat dan Bahan

Adapun perangkat keras dan perangkat lunak yang dibutuhkan sebagai berikut.

1. Perangkat Keras:
 - 3 PC / Laptop
 - 4 Mikrotik RB951-2n
 - 7 Kabel LAN
2. Perangkat Lunak:
 - Ubuntu Desktop 14.04
 - OpenDaylight Controller
 - Winbox

B. Metode Penelitian

Metode penelitian yang dilakukan penulis dalam penelitian disajikan dalam bentuk *flow chart*. Gambar 2 menunjukkan *flow chart* penelitian.

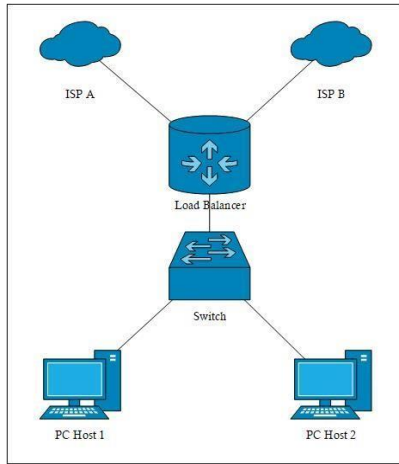


Gambar 2. *Flow Cart* Penelitian

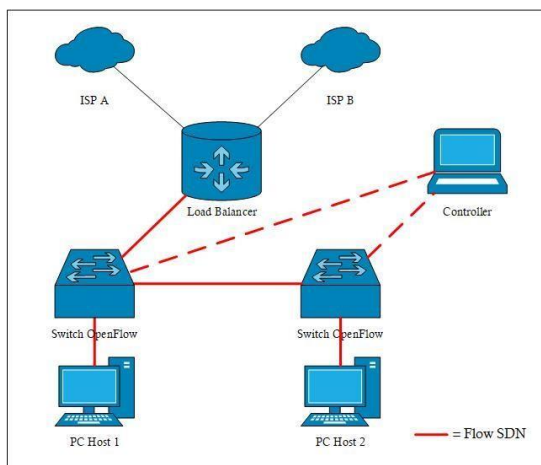
C. Implementasi Sistem Pengujian

1. Perancangan Topologi

Topologi yang digunakan pada Perbandingan dan Analisis Kinerja *Load Balancing* Metode ECMP Pada Jaringan *Software-Defined Network* sebanyak 2 skenario topologi yang semuanya menggunakan topologi *tree*.



Gambar 3. Skenario Topologi Jaringan Konvensional



Gambar 4. Skenario Topologi Jaringan SDN

2. Pengujian Sistem

Pengujian yang dilakukan untuk menganalisis perbandingan pada penelitian ini menggunakan *graphing* yang merupakan *proprietary tools* dari *router* MikroTik. *Graphing* dapat melakukan *monitoring* terhadap beberapa parameter pada *router* seperti trafik, *memory* dan CPU yang disajikan dalam bentuk grafik dan angka. Pada penelitian ini parameter yang digunakan hanya trafik saja. Grafik dari *tools graphing* dapat diakses dengan cara membuka *web browser*, dengan format alamat <http://alamat-IP-router/graphs> atau <http://192.168.1.1/graphs> pada penelitian ini. Dan untuk parameter pengujian yang digunakan yaitu berupa trafik *bandwidth* yang masuk dari *gateway* ISP-A (Internet UGM) dan ISP-B (Internet Telkomsel) ke dalam *router* untuk menguji pembagian beban serta efek *failover*. Dan efisiensi

trafik yang didistribusikan menuju jaringan Lokal yang terdapat perbedaan topologi.

Pengujian dilakukan dengan lima buah kasus yang berbeda pada masing-masing arsitektur jaringan. Host-1 dan Host-2 akan melakukan tugas yang sama pada setiap kasus dan untuk variabel pembedanya adalah *gateway* yang digunakan. Pada setiap kasus Host-1 akan melakukan pengunduhan berkas yang berukuran 1,5 GB dari situs *ubuntu.com* dan untuk Host-2 melakukan *streaming* video dari situs *youtube.com* dengan kualitas 360p.

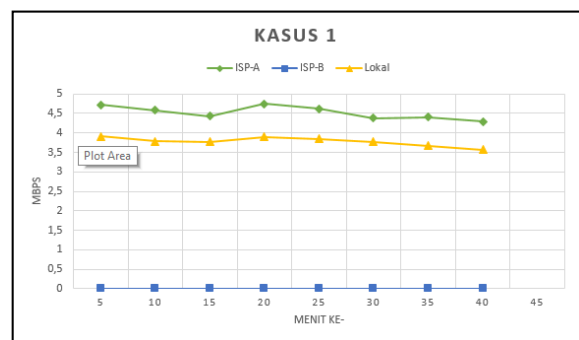
Untuk kasus pertama *gateway* yang digunakan hanya ISP-A dengan alokasi *bandwidth* 4 Mbps. Pada kasus kedua *gateway* yang digunakan hanya ISP-B dengan alokasi *bandwidth* 2 Mbps. Sedangkan pada kasus ketiga menggunakan dua *gateway* secara bersamaan yaitu ISP-A dengan alokasi *bandwidth* 4 Mbps dan ISP-B dengan alokasi *bandwidth* 2 Mbps. Pada kasus keempat alokasi *bandwidth* yang digunakan sama dengan kasus ketiga, hanya saja ditambahkan dilakukan uji *failover* dengan memutus koneksi *gateway* ISP-A dan ISP-B secara bergantian di tengah pengaksesan. Untuk kasus terakhir akan menggunakan dua jalur *gateway* dengan alokasi *bandwidth* yang sama yaitu 2 Mbps serta ditambahkan uji *failover*. Data rata-rata dari trafik *bandwidth* yang diambil dicatat menggunakan *graphing* setiap lima menit dihitung sejak setiap kasus mulai dijalankan selama 40 menit untuk masing-masing kasus.

IV. ANALISIS DAN PEMBAHASAN

Analisis data dilakukan berdasarkan kumpulan data yang telah diambil berdasar parameter pada tiap skenario pengujian yang telah ditentukan. Nilai parameter disajikan dalam bentuk grafik kemudian di analisis dan dilakukan perbandingan pada setiap skenario. Adapun analisis yang dihasilkan adalah sebagai berikut.

A. Pengujian Jaringan Konvensional

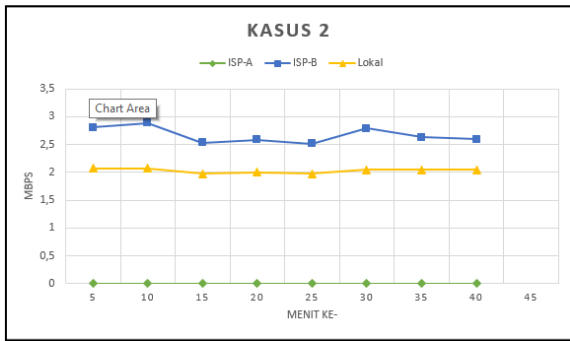
1. Pengujian Kasus 1



Gambar 5. Hasil Pengujian Trafik *Bandwidth* Pada Kasus 1 Topologi Konvensional

Dari Gambar 5 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A mengalami fluktuasi pada kisaran 4,5 Mbps. Dan trafik yang keluar menuju jaringan Lokal dibawah 4 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal tidak dapat penuh 4 Mbps seperti *bandwidth* yang disediakan oleh ISP-A.

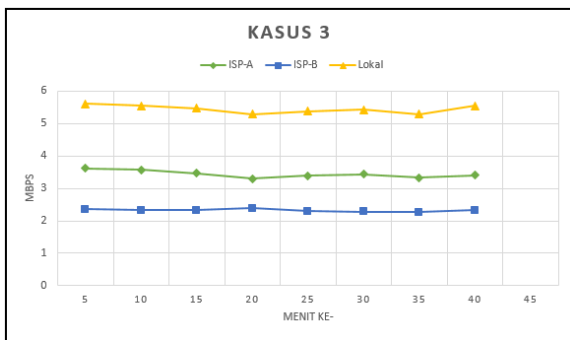
2. Pengujian Kasus 2



Gambar 6. Hasil Pengujian Trafik Bandwidth Pada Kasus 2 Topologi Konvensional

Dari Gambar 6 dapat dilihat bahwa bandwidth yang masuk dari ISP-B mengalami fluktuasi pada kisaran 2,5 hingga 3 Mbps. Dan trafik yang keluar menuju jaringan Lokal stabil pada angka 2 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal dapat 2 Mbps seperti bandwidth yang disediakan oleh ISP-B.

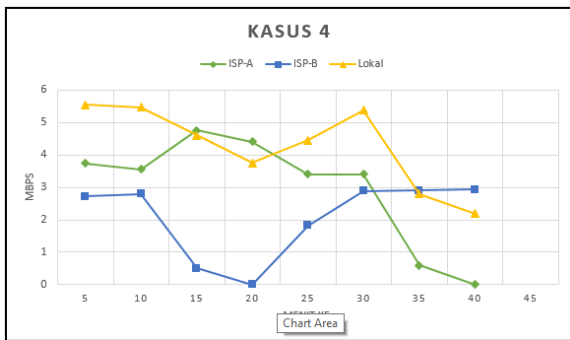
3. Pengujian Kasus 3



Gambar 7. Hasil Pengujian Trafik Bandwidth Pada Kasus 3 Topologi Konvensional

Dari Gambar 7 dapat dilihat bahwa bandwidth yang masuk dari ISP-A cukup stabil pada 3,5 Mbps dan ISP-B stabil pada 2,3 Mbps. Dan trafik yang keluar menuju jaringan Lokal stabil pada terjadi sedikit fluktuasi pada kisaran 5,5 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal sebesar 5,5 Mbps mendekati bandwidth yang digabungkan oleh ISP-A dan ISP-B yaitu 6 Mbps, hal ini dikarenakan penyeimbangan beban trafik yang dilakukan load balancer.

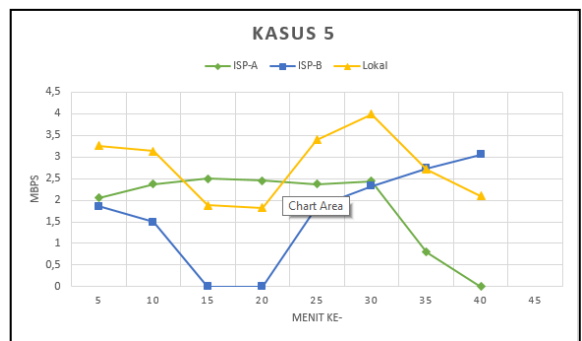
4. Pengujian Kasus 4



Gambar 8. Hasil Pengujian Trafik Bandwidth Pada Kasus 4 Topologi Konvensional

Dari Gambar 8 dapat dilihat bahwa bandwidth yang masuk dari ISP-A pada 10 menit pertama berkisar pada 3,5 Mbps dan ISP-B stabil pada 2,8 Mbps. Dan trafik yang keluar menuju jaringan Lokal pada 10 menit pertama juga stabil pada kisaran 5,5 Mbps, hal ini berarti seperti yang sudah dijelaskan pada kasus 3. Kemudian pada menit 10 hingga 20 dilakukan pemutusan koneksi pada gateway ISP-B sehingga dapat dilihat pada Gambar 4.4 terjadi penurunan drastis pada bandwidth ISP-B yang pada menit ke 20 tercatat sudah tidak ada lagi. Kebalikan dari ISP-B, pada koneksi ISP-A terjadi peningkatan secara signifikan dari kisaran 3,5 Mbps menjadi kisaran 4,5 Mbps. Pada interface Lokal trafik keluar yang awalnya kisaran 5,5 Mbps menurun pada kisaran 4,5 hingga 3,7 Mbps dikarenakan yang bekerja hanya pada gateway ISP-A yang mempunyai alokasi bandwidth 4 Mbps seperti pada kasus 1. Selanjutnya pada menit 25 hingga 30 koneksi ISP-B dinyalakan kembali sehingga trafik pada ISP-A kembali seperti 10 menit pertama pada kisaran 3,5 Mbps, pada ISP-B terjadi kenaikan bertahap hingga normal seperti pada 10 menit pertama pada menit ke 30 di kisaran 2,8 Mbps. Berikutnya pada menit 30 dan 35 dilakukan pemutusan koneksi pada ISP-A, sehingga dapat dilihat trafik ISP-A terjadi penurunan secara bertahap dari menit 35 yang masih tercatat sisa koneksi pada kisaran 0,5 Mbps hingga habis pada 5 menit terakhir. Sedangkan trafik pada ISP-B terlihat stabil pada kisaran 2,8 Mbps. Dan untuk trafik yang keluar ke jaringan Lokal terjadi penurunan menyesuaikan besaran trafik yang masuk dari ISP-A dan ISP-B.

5. Pengujian Kasus 5



Gambar 9. Hasil Pengujian Trafik Bandwidth Pada Kasus 5 Topologi Konvensional

Dari Gambar 9 dapat dilihat bahwa bandwidth yang masuk dari ISP-A pada 10 menit pertama berkisar pada 2 sampai 2,3 Mbps dan ISP-B berkisar pada 1,5 Mbps sampai 2 Mbps. Dan trafik yang keluar menuju jaringan Lokal pada 10 menit pertama stabil pada kisaran 3,2 Mbps. Kemudian mulai menit 10 hingga 20 dilakukan pemutusan koneksi pada gateway ISP-B sehingga dapat dilihat pada Gambar 4.5 terjadi penurunan drastis pada bandwidth ISP-B yang pada menit ke 15 tidak terdapat trafik. Kebalikan dari ISP-B, pada koneksi ISP-A terjadi peningkatan dari kisaran 2 Mbps menjadi kisaran 2,5 Mbps. Pada interface Lokal trafik keluar yang awalnya kisaran 3,2 Mbps menurun pada kisaran 1,8 Mbps dikarenakan

yang bekerja hanya pada *gateway* ISP-A yang mempunyai alokasi *bandwidth* 2 Mbps. Selanjutnya pada menit 20 koneksi ISP-B dinyalakan kembali sehingga trafik pada ISP-A kembali seperti 10 menit pertama pada kisaran 2,3 Mbps, pada ISP-B terjadi kenaikan bertahap di kisaran 1,8 hingga 2,3 Mbps. Berikutnya mulai menit 30 dilakukan pemutusan koneksi pada ISP-A, sehingga dapat dilihat trafik ISP-A terjadi penurunan secara bertahap dari menit 35 yang masih tercatat sisa koneksi pada kisaran 0,8 Mbps hingga habis pada 5 menit terakhir. Sedangkan trafik pada ISP-B terlihat terjadi peningkatan dari kisaran 2,3 Mbps hingga 3 Mbps. Dan untuk trafik yang keluar ke jaringan Lokal terjadi penurunan bertahap menyesuaikan besaran trafik yang masuk dari ISP-A dan ISP-B yaitu dari 3,9 Mbps menjadi 2,1 Mbps.

6. Rangkuman Data Pengujian

Tabel 1. Rangkuman Data Pengujian Jaringan Konvensional (Dalam Mbps)

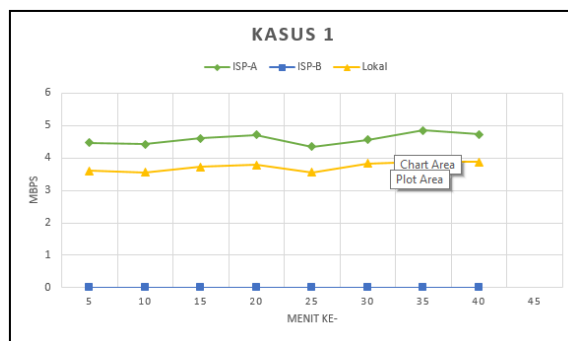
Kasus	ISP-A	ISP-B	Lokal
1	4,52	-	3,77
2	-	2,67	2,03
3	3,44	3,32	5,44
4	2,86	2	4,27
5	1,87	1,66	2,78

Tabel 1 menjelaskan pada kasus pertama terlihat bahwa trafik *bandwidth* yang masuk dari ISP-A yaitu sebesar 4,52 Mbps dan trafik yang keluar menuju jaringan Lokal sebesar 3,77 Mbps. Selanjutnya pada kasus kedua trafik yang masuk dari ISP-B sebesar 2,67 Mbps dan trafik yang keluar sebesar 2,03 Mbps. Kemudian pada kasus ketiga ketika menggunakan dua *gateway* secara bersamaan terlihat penurunan trafik pada masing-masing *gateway* di mana ISP-A menjadi 3,44 Mbps dan ISP-B menjadi 2,3 Mbps, namun trafik yang menuju jaringan Lokal menjadi sebesar 5,42 Mbps. Hal ini disebabkan pembagian beban trafik yang dilakukan oleh *load balancing* ECMP untuk menyeimbangkan beban trafik secara *equal-cost* pada kedua *gateway* dan membuat trafik yang keluar lebih besar sebanding dengan jumlah trafik yang masuk.

Pada kasus keempat, meskipun alokasi *bandwidth* yang digunakan sama dengan kasus ketiga, dapat dilihat terjadi penurunan trafik pada seluruh *interface*, hal ini disebabkan dilakukannya uji *failover* pada pertengahan pengaksesan, yang menunjukkan trafik ISP-A menjadi sebesar 2,86 Mbps, ISP-B sebesar 2 Mbps dan pada *interface* Lokal sebesar 4,27 Mbps. Untuk kasus ke lima menggunakan dua *gateway* secara bersamaan dengan alokasi *bandwidth* sama besar 2 Mbps, rata-rata trafik untuk ISP-A tercatat sebesar 1,87 Mbps dan ISP-B 1,66 Mbps sehingga menghasilkan rata-rata trafik keluar sebesar 2,78 Mbps. Meskipun pada masing-masing *gateway* trafik yang dialokasikan sebesar 2 Mbps, rata-rata trafik yang didapat dibawah 2 Mbps karena dilakukan uji *failover* seperti kasus ke 4.

B. Pengujian Jaringan SDN

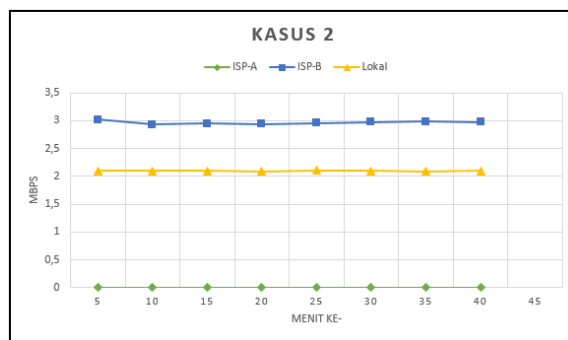
1. Pengujian Kasus 1



Gambar 10. Hasil Pengujian Trafik Bandwidth Pada Kasus 1 Topologi SDN

Dari Gambar 10 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A mengalami fluktuasi pada kisaran 4,5 Mbps. Dan trafik yang keluar menuju jaringan Lokal dibawah 4 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal tidak dapat penuh 4 Mbps seperti *bandwidth* yang disediakan oleh ISP-A.

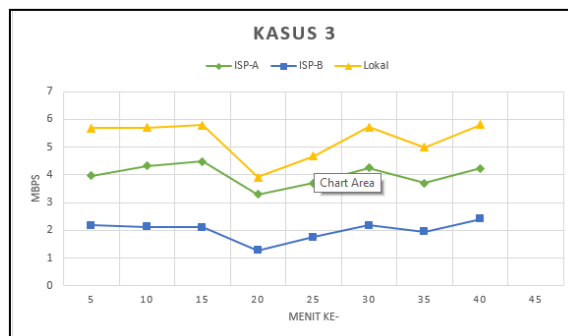
2. Pengujian Kasus 2



Gambar 11. Hasil Pengujian Trafik Bandwidth Pada Kasus 2 Topologi SDN

Dari Gambar 11 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-B terlihat stabil pada kisaran 2,9 Mbps. Dan trafik yang keluar menuju jaringan Lokal stabil pada angka 2,1 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal dapat 2 Mbps seperti *bandwidth* yang disediakan oleh ISP-B

3. Pengujian Kasus 3

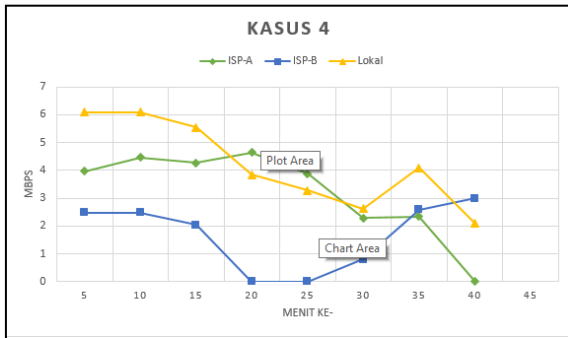


Gambar 12. Hasil Pengujian Trafik Bandwidth Pada Kasus 3 Topologi SDN

Dari Gambar 12 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A cukup stabil pada kisaran 4,2

Mbps hanya saja mengalami penurunan pada menit 20 dan menit 35 dikarenakan terjadi penurunan *request* trafik pada *host* yang berpengaruh pada kedua *gateway*. Pada ISP-B juga mengalami juga stabil pada 2,1 Mbps dan sempat mengalami penurunan trafik yang sama terjadi pada ISP-A antara menit 15 sampai 25 dan pada menit 35. Dan trafik yang keluar menuju jaringan Lokal stabil pada kisaran 5,7 Mbps, karena trafik keluar menuju jaringan Lokal menyesuaikan trafik yang masuk maka terjadi penurunan juga antara menit 15 sampai 25 dan pada menit 35

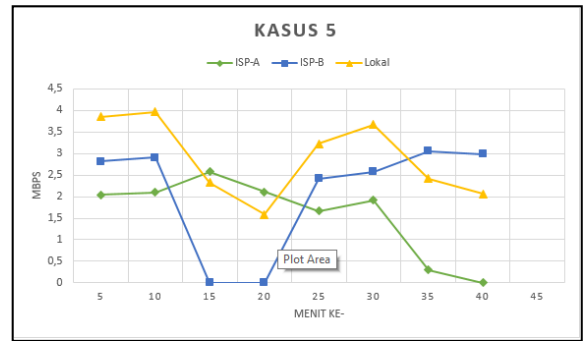
4. Pengujian Kasus 4



Gambar 13. Hasil Pengujian Trafik Bandwidth Pada Kasus 4 Topologi SDN

Dari Gambar 13 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A pada 10 menit pertama berkisar antara 4 Mbps sampai 4,3Mbps kemudian mengalami pada ISP-B stabil dikisaran 2,5 Mbps. Dan trafik yang keluar menuju jaringan Lokal pada 10 menit pertama juga stabil pada kisaran 6 Mbps sebanding dengan trafik yang masuk. Kemudian pada menit 15 hingga 25 dilakukan pemutusan koneksi pada *gateway* ISP-B sehingga dapat dilihat pada Gambar 4.8 terjadi penurunan drastis pada *bandwidth* ISP-B pada menit ke 20 hingga trafik menjadi 0 Mbps. Dan pada koneksi ISP-A terjadi peningkatan sedikit dari kisaran 4 Mbps menjadi 4,5 Mbps. Pada *interface* Lokal trafik keluar yang awalnya kisaran 6 Mbps menurun pada kisaran 3,9 Mbps dikarenakan yang bekerja hanya pada *gateway* ISP-A yang mempunyai alokasi *bandwidth* 4 Mbps seperti pada kasus 1. Selanjutnya pada menit 25 hingga 35 koneksi ISP-B dinyalakan kembali sehingga pada ISP-B terjadi kenaikan bertahap hingga normal seperti pada 10 menit pertama pada menit ke 35 di kisaran 2,7 Mbps. Berikutnya pada menit 35 dilakukan pemutusan koneksi pada ISP-A, sehingga dapat dilihat trafik ISP-A terjadi penurunan secara bertahap dari menit 35 hingga habis pada menit 40. Sedangkan trafik pada ISP-B terlihat stabil pada kisaran 2,8 Mbps. Dan untuk trafik yang keluar ke jaringan Lokal terjadi penurunan menyesuaikan besaran trafik yang masuk dari ISP-A dan ISP-B.

5. Pengujian Kasus 5



Gambar 14. Hasil Pengujian Trafik Bandwidth Pada Kasus 5 Topologi SDN

Dari Gambar 14 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A pada 10 menit pertama berkisar pada 2 Mbps dan ISP-B berkisar pada 2,8 Mbps. Dan trafik yang keluar menuju jaringan Lokal pada 10 menit pertama stabil pada kisaran 3,8 Mbps. Kemudian mulai menit 10 hingga 20 dilakukan pemutusan koneksi pada *gateway* ISP-B sehingga dapat dilihat pada Gambar 4.10 terjadi penurunan drastis pada *bandwidth* ISP-B yang pada menit ke 15 tidak terdapat trafik. Kebalikan dari ISP-B, pada koneksi ISP-A terjadi peningkatan dari kisaran 2 Mbps menjadi kisaran 2,5 Mbps. Pada *interface* Lokal trafik keluar yang awalnya kisaran 3,8 Mbps menurun pada kisaran 2,3 Mbps hingga 1,5 Mbps dikarenakan yang bekerja hanya pada *gateway* ISP-A yang mempunyai alokasi *bandwidth* 2 Mbps. Selanjutnya pada menit 20 koneksi ISP-B dinyalakan kembali sehingga trafik pada ISP-A turun lagi pada kisaran 2 Mbps, pada ISP-B terjadi kenaikan menjadi 2,5 Mbps. Berikutnya mulai menit 30 dilakukan pemutusan koneksi pada ISP-A, sehingga dapat dilihat trafik ISP-A terjadi penurunan secara bertahap dari menit 35 yang masih tercatat sisa koneksi pada kisaran 0,3 Mbps hingga habis pada 5 menit terakhir. Sedangkan trafik pada ISP-B terlihat terjadi peningkatan hingga 3 Mbps. Dan untuk trafik yang keluar ke jaringan Lokal terjadi penurunan bertahap menyesuaikan besaran trafik yang masuk dari ISP-A dan ISP-B yaitu dari 3,6 Mbps menjadi 2 Mbps

6. Rangkuman Data Pengujian

Tabel 2. Rangkuman Data Pengujian Jaringan SDN (Dalam Mbps)

Kasus	ISP-A	ISP-B	Lokal
1	4,58	-	3,72
2	-	2,96	2,09
3	4,37	2,17	5,74
4	3,14	1,62	4,2
5	1,58	1,72	2,62

Tabel 2 Menjelaskan bahwa pada kasus pertama terlihat bahwa trafik *bandwidth* yang masuk dari ISP-A yaitu sebesar 4,58 Mbps dan trafik yang keluar menuju jaringan Lokal sebesar 3,72 Mbps. Selanjutnya pada kasus kedua trafik yang masuk dari ISP-B sebesar 2,96 Mbps dan trafik yang keluar sebesar 2,09 Mbps. Kemudian pada kasus ketiga ketika menggunakan dua *gateway* secara bersamaan

terlihat penurunan trafik pada masing-masing *gateway* di mana ISP-A menjadi 4,37 Mbps dan ISP-B menjadi 2,17 Mbps, namun trafik yang menuju jaringan Lokal menjadi sebesar 5,74 Mbps.

Pada kasus keempat, meskipun alokasi *bandwidth* yang digunakan sama dengan kasus ketiga, dapat dilihat terjadi penurunan trafik pada seluruh *interface*, hal ini disebabkan dilakukannya uji *failover* pada pertengahan pengaksesan, yang menunjukkan trafik ISP-A menjadi sebesar 3,14 Mbps, ISP-B sebesar 1,62 Mbps dan pada *interface* Lokal sebesar 4,2 Mbps. Ini menunjukkan bahwa beban trafik yang sebelumnya diseimbangkan oleh *load balancing* pada dua *gateway* menjadi hilang sehingga beban trafik kembali membebani salah satu *gateway* yang masih aktif saja. Untuk kasus ke lima menggunakan dua *gateway* secara bersamaan dengan alokasi *bandwidth* sama besar 2 Mbps, rata-rata trafik untuk ISP-A tercatat sebesar 1,58 Mbps dan ISP-B 1,72 Mbps² sehingga menghasilkan rata-rata trafik keluar sebesar 2,62 Mbps. Meskipun pada masing-masing *gateway* trafik yang dialokasikan sebesar 2 Mbps, rata-rata trafik yang didapat dibawah 2 Mbps karena dilakukan uji *failover* seperti kasus ke 4.

C. Perbandingan Data Hasil Pengujian

Tabel 3. Perbandingan Data Hasil Pengujian Paringan Konvensional (K) dan SDN (S) (Dalam Mbps)

Kasus	ISP-A	ISP-B	Lokal	Efisiensi
1 – K	4,52	-	3,77	83,4%
1 – S	4,58	-	3,72	81,2%
2 – K	-	2,67	2,03	76%
2 – S	-	2,96	2,09	70,6%
3 – K	3,44	3,32	5,44	94,4%
3 – S	4,37	2,17	5,74	87,7%
4 – K	2,86	2	4,27	87,8%
4 – S	3,14	1,62	4,2	88,2%
5 – K	1,87	1,66	2,78	78,7%
5 - S	1,58	1,72	2,62	79,3%

Untuk pembandingan pengaruh *failover* dari kedua topologi dapat dilihat efisiensi trafik pada *interface* Lokal yang didistribusikan menuju jaringan lokal, sedangkan untuk *interface* ISP-A dan ISP-B di sini untuk melihat efisiensi data yang masuk dari kedua *gateway* yang kemudian didistribusikan keluar ke *interface* Lokal. Dari Tabel 4.7 dapat dilihat bahwasannya secara keseluruhan efisiensi trafik dari jaringan konvensional lebih baik dari jaringan SDN dan untuk data yang masuk pada SDN secara keseluruhan lebih besar dibanding dengan jaringan konvensional. Merujuk dari Tabel 4.7 pada kasus 4 dan 5, topologi jaringan SDN lebih baik tingkat efisiensi trafiknya, dapat disimpulkan bahwa untuk pengaruh *failover* pada jaringan SDN memiliki kemampuan yang lebih baik.

V. KESIMPULAN

Berdasarkan hasil analisis pengujian, maka dapat diambil kesimpulan bahwa Efisiensi trafik pada jaringan konvensional bila tanpa pengaruh *failover* pada jaringan konvensional menunjukkan angka yang lebih baik dari

jaringan SDN dengan selisih 2,2% untuk kasus 1, 5,4% untuk kasus 2 dan 6,7% untuk kasus 3. Kemudian untuk Efisiensi trafik dengan pengaruh *failover* menunjukkan bahwa jaringan SDN memiliki angka yang lebih baik dari jaringan konvensional dengan selisih 0,4% untuk kasus 4 dan 0,6 % untuk kasus 5. Dan terakhir bahwa Intergrasi *load balancing* metode ECMP baik pada arsitektur topologi konvensional dan arsitektur topologi SDN, kurang cocok untuk jaringan yang membutuhkan *session login*, dikarenakan pada metode ECMP setiap 10 menit melakukan *routing table flushing* seperti yang sudah dijelaskan pada Bab II, sehingga memerlukan *login* ulang atau setidaknya menunggu untuk *refresh login* yang mana akan membuat jaringan kurang stabil.

DAFTAR PUSTAKA

- [1] G. Prakosa, Analisis Perbandingan Metode Load Balancing Ecmp, Nth, Dan Pcc Menggunakan Mikrotik Cloud Hosted Router Pada Gns3, Yogyakarta: Universitas Gadjah Mada, 2016.
- [2] I. Grgurević, Z. Kavran dan A. Pušeljić, Simulation Analysis of Characteristics and Application of Software-Defined Networks, Zagreb: University of Zagreb, 2015.
- [3] R. Kartadi, Uji Performa Kontroler Floodlight Dan OpenDaylight Sebagai Komponen Utama Arsitektur Software-Defined Network, Yogyakarta: STIMIK AMIKOM Yogyakarta, 2015.
- [4] M. Didik, Modul Praktikum Jaringan Komputer, Yogyakarta: UIN Sunan Kalijaga, 2011.
- [5] B. S. D. Oetomo, Konsep dan Perancangan Jaringan Komputer, Yogyakarta: Andi, 2003.
- [6] Open Network Foundation, "Software-Defined Networking (SDN) Definition," 2018. [Online]. Available: <https://www.opennetworking.org/sdn-definition/>. [Diakses 11 Mei 2018].
- [7] SDX-Central, "What is OpenFlow? Definition and How it Relates to SDN," 2012. [Online]. [Diakses 12 Mei 2018].
- [8] W. Rahmaning, Implementasi Dan Analisis Kinerja Protokol Routing BGP (Border Gateway Protocol) Pada Jaringan SDN (Software Defined Network), Yogyakarta: UGM, 2017.
- [9] SDX-Central, "What is an OpenDaylight Controller? AKA: OpenDaylight Platform," 2012. [Online]. Available: <https://www.sdxcentral.com/sdn/definitions/sdn-controllers/openflow-controller/>. [Diakses 13 Mei 2018].
- [10] H. Nugroho dan M. R. Utama, "Simulasi Management Bandwidth dan Load Balancing Server Menggunakan Clear OS pada Virtual Box," *Jurnal ICT Penelitian dan Penerapan Teknologi*, Vol. %1 dari %2-, no. -, p. 82, 2014.
- [11] M. F. Adani, Analisis Perbandingan Metode Load Balance PCC Dengan Nth Menggunakan MikroTIK, 1st penyunt., Surabaya: Institu Bisnis & Informatika STIKOM, 2016.

- [12] R. Ramandito, S. Sumaryono dan S. Suning, "Analisis Performance Jaringan Komputer dengan Mekanisme Load Balancing-Failover," *Jurnal Penelitian Teknik Elektro*, vol. 3, p. 178, 2010.
- [13] MikroTik, "ECMP load balancing with masquerade," 2016. [Online]. Available: https://wiki.mikrotik.com/wiki/ECMP_load_balancing_with_masquerade. [Diakses 28 Juli 2018].
- [14] R. Towidjojo, Mikrotik Kung Fu Kitab 2, Jakarta: Jasakom, 2013.
- [15] K. D. T. Saputro, Membangun Server Internet dengan Mikrotik OS, Yogyakarta: Gava Media, 2008.