

**PENGUJIAN PERFORMA DAN ANALISIS PERBANDINGAN *CONTROLLER POX* DAN *RYU* PADA
JARINGAN *SOFTWARE DEFINED NETWORK***

***PERFORMANCE TESTING AND ANALYSIS OF CONTROLLER POX AND RYU COMPARISON ON
DEFINED NETWORK SOFTWARE NETWORK***

Kevin Saputro¹, Andi Suprianto²

Program Studi Teknik Informatika, Fakultas Sains dan Teknologi Informasi
Institut Sains dan Teknologi Nasional
Jl. Moh. Kahfi II, Bhumi Srengseng Indah, Jagakarsa, Jakarta Selatan 12640
¹saputro.kevin@gmail.com, ²andi.suprianto@istn.ac.id

ABSTRAKSI

Infrastruktur jaringan utama yang hampir seluruhnya di kelola oleh vendor menyebabkan semakin kompleksnya jaringan yang di bangun, sehingga di butuhkan pendekatan untuk mengelola dan mengimplementasikan jaringan agar mendukung kebutuhan yang beragam, yaitu dengan menggunakan *Software Defined Networking (SDN)*. *Software-Defined Networking (SDN)* adalah sebuah konsep pendekatan baru untuk mendesain, membangun dan mengelola jaringan komputer dengan memisahkan *control plane* dan *data plane*. Komponen utama dari *Software Defined Network* adalah *controller* yang secara langsung melakukan *control* terhadap *datapath* dari perangkat. Ada sepuluh hal yang penting dalam pemilihan *controller* menurut Asthon, M.(2013) salah satunya adalah performa dari *controller*. Beberapa *controller* yang ada diantaranya adalah *POX* dan *RYU controller*, untuk mengetahui manakah *controller* yang memiliki performa yang baik, maka di lakukanlah analisis perbandingan performa kedua *controller* tersebut. Pengujian di lakukan menggunakan *Mininet* sebagai emulator untuk merancang topologi pengujian sedangkan *QoS* menggunakan *iperf* dan *wireshark*. Hasil pengujian menunjukkan konsep jaringan *SDN* berjalan dengan baik, pada analisis ini menggunakan parameter *Delay*, *Jitter*, *Packetloss* dan *Throughput*. Dari data yang di dapat *controller RYU* mendapatkan nilai *delay* dan *Jitter* yang baik dari dua pengujian 2 dan 8 *switch* yaitu di bawah 0,0003s dan 0,01ms. Sedangkan untuk *Packetloss* kedua *controller* cukup berimbang dengan selisih hanya 0,1% dan pada *Throughput* pada *RYU 8 switch* mengalami penurunan yang signifikan menjadi 836mbit/sec.

Kata Kunci : *Software Defined Network, Ryu, Pox, Mininet, Quality of Service*

ABSTRACT

The main network infrastructure that is almost entirely managed by the vendor causes increasingly complex networks that are built, so that an approach is needed to manage and implement the network to support diverse needs, namely by using Software-Defined Networking (SDN). Software-Defined Networking (SDN) is a new conceptual approach to design, build and manage computer networks by separating the control plane and the data plane. The main component of Software Defined Network is the controller that directly controls the data from the device. There are ten things that are important in the selection of controllers according to Asthon, M. (2013) only one is the performance of the controller. Some controllers include POX and RYU controller, to find out which controller has good performance, then do a comparative analysis of the performance of the two controllers. Testing is done using Mininet as an emulator to design the test topology while QoS uses iperf and wireshark. The test results show the concept of the SDN network running well, in this analysis using the parameters Delay, Jitter, Packetloss and Throughput. From the data obtained, the RYU controller gets a good delay and Jitter value from the two 2 and 8 switch tests, which are below 0,0003s and 0.01ms. Whereas for Packetloss the two controllers are quite balanced with a difference of only 0.1% and the Throughput on RYU 8 switches has decreased significantly to 836mbit / sec.

Keywords : Software Defined Network, Ryu, Pox, Mininet, Quality of Service

1. PENDAHULUAN

Pada era sekarang ini perkembangan di bidang jaringan komputer sudah sangat pesat, dalam perkembangannya jaringan internet

merupakan salah satu teknologi yang sangat di butuhkan oleh masyarakat, yang dimana hampir bahkan seluruh aktivitas yang di lakukan tidak lepas dari jaringan internet. Tetapi infrastruktur jaringan utama yang tidak

berubah menyebabkan semakin lama jaringan yang di bangun semakin kompleks, dan masalah lainnya *router* dan *switch* yang di gunakan saat ini di distribusi secara khusus, tertutup dan merupakan hak milik perusahaan. Dengan *administrator* jaringan yang secara khusus mengkonfigurasi perangkat jaringan menggunakan antarmuka konfigurasi *vendor based*. Berdasarkan masalah insfrastruktur yang terjadi di perlukan pendekatan untuk memudahkan pengelolaan terhadap kondisi jaringan yang semakin beragam.

Software-defined Network (SDN) adalah sebuah konsep pendekatan jaringan komputer dimana sistem pengontrol dari arus data dipisahkan dari perangkat kerasnya. Dengan kata lain SDN juga dapat disebut sebagai sebuah paradigma baru di dunia *networking*, sebuah pendekatan dalam membangun, mendesain serta mengelola jaringan komputer. *Controller* menggunakan *protocol openflow* untuk melakukan konfigurasi terhadap perangkat jaringan dan memilih jalur trafik data yang optimal. Ada banyak jenis *controler* yang tersedia dalam *Software defined network* Contoh-contoh *controler* seperti NOX, Pox, Ryu, Trema, Jaxon, Beacon, Maestro, ONOS, Floodlight, and Open-daylight. Beberapa *controller* memiliki GUI sebagai fasilitas antar-muka mereka sementara beberapa *controller* tidak menggunakannya.

Dalam penelitian ini akan dipilih pengujian performa terhadap kontroler Pox dan Ryu. Perbedaan pada kontroler ini dapat menentukan kegunaan masing-masing pada kontroler tersebut dan pengaruhnya terhadap performansi jaringan. Oleh karena itu pada tugas akhir ini akan melakukan analisis perbandingan terhadap dua kontroler yakni, Ryu dan Pox menggunakan virtualisasi jaringan pada Mininet. Ryu dan Pox yang merupakan kontroler yang berbasis *python*. Analisis ini berdasarkan performansi untuk menentukan *controller* yang lebih baik dan akan dibandingkan kembali ke *controller* lainnya.

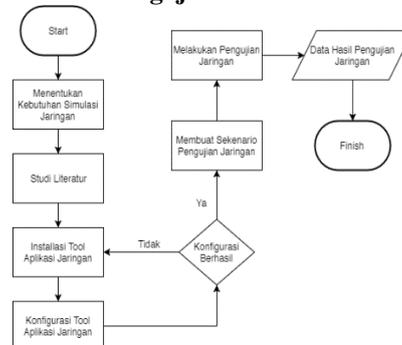
Kedua kontroler akan ditesting dengan alat uji performa yang sering digunakan dalam penelitian yaitu menggunakan *Wireshark* dan *iperf*. *Iperf* didefinisikan sebagai alat yang digunakan untuk pengukuran, analisis dan penyetelan kinerja jaringan. Ini penting sebagai alat lintas-platform yang dapat menghasilkan pengukuran kinerja standar untuk jaringan apa pun. Dengan menggunakan *iperf* dan *Wireshark* kita bisa melihat nilai *Throughput*, *Jitter*, *Packet Loss* dan *Delay* dari hasil pengujian masing-masing *controller* yang memungkinkan untuk mengetahui nilai performa dari kinerja *controller* tersebut.

2. METODOLOGI PENELITIAN

Objek penelitian

Pengujian jaringan *software defined network* menggunakan dua *controller* yaitu *controller* Pox dan Ryu menggunakan ubuntu dan *tools* pendukung seperti *wireshark* dan nano editor pada ubuntu. Pengujian hanya akan mencari data nilai QoS dari kedua *controller* yang di ujikan. Terdapat referensi jurnal perbandingan dari pengujian ini yaitu [Abdillah. D, Yuliant. S, Izzatul. U. 2016] dalam Perancangan dan Analisis jaringan virtual berbasis *software defined network*. Jurnal tersebut membahas masalah pengujian QoS pada jaringan *software defined network* menggunakan *controller* Pox dengan sekenario 2, 4, 8, dan 16 *switch*. Pada penulisan ini menggunakan *controller* Pox dan Ryu dengan sekenario 2 dan 8 *switch* sebagai variasi pengujian yang di lakukan, penggunaan sekenario *switch* 2 dan 8 *switch* di lakukan karena selain adanya peningkatan jumlah *switch* yang signifikan walaupun 4 ke 16 atau 8 ke 16 jauh lebih tinggi, ini nantinya agar pada saat konfigurasi tidak terlalu banyak menggunakan *switch* selain mempersingkat konfigurasi juga nantinya pengujian akan fokus menguji antara *controller* Pox dan Ryu yang akan di uji.

Metode Pengujian



Gambar flowchart Metode penelitian

Konsep komunikasi jaringan *software defined network* sangat digunakan dalam proses pengujian/pengukuran jaringan. Untuk menjalankan skenario/proses pengujian kinerja jaringan, digunakan perintah *ping* dan *iperf* dan *wireshark*. Pengujian jaringan nantinya akan menguji berapa nilai *delay*, *jitter*, *packet loss* dan *throughput* yang di dihasilkan dari topologi yang akan di buat pada *Controller* POX dan RYU.

Proses yang pertama di lakukan dalam melakukan pengujian jaringan yaitu dengan menghubungkan antara *controller* dengan topologi jaringan yang sudah di buat dengan mengaktifkan *controller* terlebih dahulu kemudian di ikuti dengan mengaktifkan

topologi jaringan yang sudah di buat, setelah itu aktifkan *wireshark* pada terminal ubuntu untuk melakukan uji *delay* jaringan, setelah *delay* lanjutkan dengan pengujian *throughput*, *jitter* dan *packetloss*. Pada pengujian *throughput* menggunakan tools *xterm* yang di buka di *mininet* di ikuti dengan dua host untuk pengujian *server* dan *client*, kemudian uji jaringan dengan perintah *iperf* pada tiap *node* yang sudah di buka. Pengujian sendiri dilakukan dengan menguji paket yang lewat pada port TCP dan UDP, untuk hasil *jitter* dan *packetloss* terdapat hasil yang muncul pada pengujian UDP *throughput*.

Dari pengujian jaringan ini ada parameter tersendiri yang akan di gunakan sebagai acuan untuk pengukuran nilai jaringan itu sendiri, ini berbeda dengan parameter jurnal hasil yang sudah di tentukan di awal. para meter tersendiri akan menggunakan setandar dari TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*), berikut parameternya :

Tabel Standar TIPHON untuk Throughput

Kategori Throughput	Throughput	Indeks
Bad	0 – 338 kbps	0
Poor	338 – 700 kbps	1
Fair	700 – 1200 kbps	2
Good	1200 kbps – 2,1 Mbps	3
Excelent	>2,1 Mbps	4

Tabel Standar TIPHON untuk delay

Kategori Latency	Latency	Indeks
Poor	> 450 s	1
Medium	300 – 450 s	2
Good	150 – 300 s	3
Perfect	< 150 s	4

Tabel Standar TIPHON untuk PaketLoss

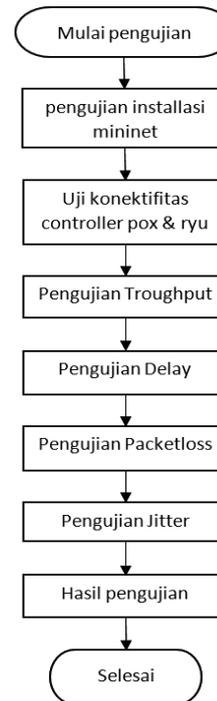
Kategori Packet Loss	Packet Loss	Indeks
Poor	>25%	1
Medium	12 – 24%	2
Good	3 – 14%	3
Perfect	0 – 2%	4

Tabel 4. Standar TIPHON untuk jitter

Kategori Jitter	Jitter	Indeks
Poor	125 – 225 ms	1
Medium	75 – 125 ms	2
Good	0 – 75 ms	3
Perfect	0 ms	4

3. HASIL DAN PEMBAHASAN

A. Rancangan pengujian



Data hasil dari pengujian yang di lakukan di sajikan dalam tabel dan grafik agar lebih mudah dalam menganalisis hasil dari perhitungan jaringan *Software Defined Network*, hasil tersebut berupa *delay*, *jitter*, *packetloss*, dan *throughput*.

Tabel Nilai pengujian ryu dan pox 2 switch

Jumlah dan pengukuran	Delay (ms)	Jitter (ms)	Packet loss %	Throughput	
				TCP	UDP
2 Switch POX	0,0038	0,13	0,82	925	940
2 Switch RYU	0,0001	0,01	1,1	926	940

Pada tabel di atas menunjukkan semua data nilai dari *delay*, *jitter*, *packet loss* dan *throughput* pada sekenario 2 switch. Data di ambil berdasarkan masing masing sekenario topologi yg di bangun. Pada data di atas niai *delay* pada topologi 2 switch controller Ryu dan Pox menunjukkan hasil yang cukup berbeda dan perbandingannya cukup signifikan, pada Ryu jelas terlihat nilai *delay* yang sangat kecil di dibandingkan dengan Pox, untuk *jitter* sendiri pun pada Ryu juga cukup baik mendapatkan selisih 0,11ms di banding dengan Pox. Sedangkan pada *packetloss* controller Pox lebih baik dari Ryu dan pada *throughput* kecepatan akses rata-rata cukup sama baik TCP maupun UDP.

Tabel Nilai pengujian ryu dan pox 8 switch

Jumlah dan pengukuran	Delay (ms)	Jitter (ms)	Packet loss %	Troughput	
				TCP	UDP
8 Switch POX	0,0042	0,04	2,3	916	951
8 Switch RYU	0,0003	0	2,4	921	836

Sedangkan pada sekenario 8 switch nilai delay juga mengalami kesamaan untuk perbandingan controller, yaitu nilai Ryu lebih sedikit nilai delay di banding dengan controller Pox, begitu juga dengan nilai jitter pada topologi 8 switch cenderung pada controller Ryu mendapatkan nilai 0ms yang dimana ini merupakan nilai yang sangat baik, kemudian pada paketloss controller Ryu cenderung mengalami loss data yang lebih banyak dari controller Pox sama halnya seperti sekenario 2 switch sebelumnya, dan untuk nilai throughput data pada pengujian controller 8 switch ini cukup beragam, pada tetapi ada cukup banyak penurunan pada UDP Ryu dengan 836Mbps/sec dimana ini bisa terjadi karena pengiriman datagram pada saat pengiriman data akses internet mengalami penurunan bandwith maupun karena adanya loss data pada jaringan. Tetapi dari semua data yang terdapat di atas yang sudah di uji pada kedua controller cenderung menyimpulkan keduanya cukup baik dari segi standar TIPHON yang di gunakan.

Delay



Gambar Nilai delay

Pada gambar di atas menunjukan nilai dari delay software defined network untuk beberapa sekenario pengujian dua controller, pada data di atas menunjkan bahwa nilai delay keseluruhan tidak tetap dan lebih ke mengalami kenaikan yang tidak terlalu tinggi pada controller POX 2 switch terlihat hasil delay yang masih dalam batas yang baik mengalami kenaikan pada pengujian POX dengan 8 switch, hal ini dapat kita lihat bahwa penambahan switch atau device jaringan pada controller pox dapat berpengaruh pada nilai delay walaupun masih dalam batas yang relative baik. Sama halnya dengan controller

RYU yang mengalami kenaikan delay pada pengujian ke dua dengan 8 switch, hal ini terjadi karena pada saat penentuan data flow mengalami penambahan deskripsi seperti perangkat jaringan yang dituju dan port-port yang tersedia pada suatu switch pada jaringan tersebut sehingga menyebabkan waktu pengecekan alamat fisik dari perangkat jaringan dan port-port yang tersedia menjadi bertambah. Berdasarkan rekomendasi standard delay dari TIPHON jaringan software defined network dengan pengujian ini masih tergolong sangat baik karena memiliki nilai delay yang masih di bawah dari rekomendasi sangat baik yaitu 0,003ms dan 0,042ms masih jauh lebih kecil dari 150ms dan belum sampai ke 1 ms.

Jitter



Gambar 4. Nilai Jitter

Pada grafik nilai jitter diatas menunjukkan bahwa nilai jitter cenderung menurun, dapat dilihat antara 2 switch dengan 8 switch pada controller POX yang mengalami penurunan yang cukup signifikan dan antara 2 switch dengan 8 switch dengan controller RYU juga mengalami penurunan namun sangat sedikit. Hal ini menunjukkan kenaikan jumlah switch tidak berbanding lurus dengan kenaikan nilai jitter. Namun suatu penurunan yang terjadi karena pengaruh dari waktu komunikasi (nilai delay) itu sendiri. Sedangkan hanya pada POX 2 switch lah yang mempunyai nilai jitter paling tinggi, hal itu menunjukkan waktu delay yang diperlukan untuk tiap-tiap host sangat bervariasi. Namun nilai-nilai jitter tersebut masih memenuhi standard rekomendasi dari TIPHON yang cukup baik pada dari 2 dan 8 switch dan bahkan masih belum sampai ke nilai 1 ms.

Troughput



Gambar Nilai troughput

Dari data nilai *throughput* di atas dapat di jelaskan bahwa nilai TCP *throughput* relative lebih setabil cenderung mengalami naik turun yang masih dalam batas standar, terdapat kenaikan pada *throughput* TCP *controller* POX 8 *switch* dimana berarti *transfer* data pada jaringan ini cukup baik tetapi selain itu pada UDP POX terjadi penurunan yang dimana terjadi karena penambahan jalur dan node yg signifikan namun hasil nya masih lebih dari cukup, selain itu pada UDP *controller* RYU baik 2 *switch* maupun 8 *switch* semuanya mengalami penurunan, bahkan yang lebih tinggi ada pada RYU 8 *switch* TCP, namun secara umum pada masing-masing *controller* pada 2 *switch* bias di sebut nilai *throughput*nya masih yang paling baik. Namun bila melihat dari standar TIPHON sebagai acuan maka kecepatan antara TCP dan UDP cukup baik karena lebih dari 1000kbps terutama pada topologi 2 *switch* tiap *controller* relative sama. Walau kalau di lihat dapat di artikan port TCP lebih cepat untuk transfer data di banding dengan UDP pada jaringan *Software defined network*.

Paketloss



Gambar Nilai paketloss

Yang terakhir pada perhitungan dan perbandingan *packetloss* dapat di lihat pada masing -masing *controller* dan tiap penambahan *switch* mengalami kenaikan loss packet yang terjadi, pada *controller* POX 2 *switch* ke 8 *switch* sebesar 1,48ms dan pada RYU 2 *switch* ke 8 *switch* juga mengalami kenaikan yaitu sekitar 1,3ms. Dari data *packet loss* di atas dapat di simpulkan bahwa secara umum kenaikan terjadi akibat penambahan *device* pada setiap *controller* yang ini berpengaruh pada hilangnya data pada saat pengiriman packet yang di lakukan karena data harus melewati beberapa *switch* dan ini terjadi juga karena pembatasan *background traffic* sebesar 1000mbps yang di lakukan pada saat pengujian. Namun dari data *packet loss* yang di hasilkan mendapatkan hasil yang cukup baik bila kita melihat dari setandar TYPHON yang di gunakan yaitu sebesar 3% data yang hilang.

4. SIMPULAN

Setelah melakukan perbandingan pengujian performasi kinerja jaringan dengan *controller* POX dan RYU dengan parameter dapat di lihat dalam pengujian dan pengukuran QoS jaringan. Hasil analisis dari *controller* POX dan RYU terhadap performasi jaringan adalah sebagai berikut :

1. Software Defined Network sangat kompatibel bila di buat dan di konfigurasi menggunakan emulator mininet, selain into mininet juga sangat muda di install pada system operasi ubuntu yang di gunakan.
2. Hasil dari pengujian QoS kontroler POX dan RYU pada topologi 2 *switch*, dan 8 *switch* yang telah dirancang menunjukkan nilai delay, jitter, packetloss dan *throughput* yang didapatkan yaitu rata-rata delay pada POX antara 2 *switch* dan 8 *switch* terendah yaitu pada 2 *switch* sebesar 0.0038ms. sedangkan delay dari RYU antara 2 dan 8 *switch* yaitu 0.0001ms pada 2 *switch*. Jitter dengan rata-rata terendah antara 2 dan 8 *switch* pada POX yaitu 0.04, sedangkan pada RYU terdapat nilai 0 pada jitter 8 *switch*. Pada paketloss nilai rata-rata tertinggi pada *controller* RYU dan POX ada pada topologi 8 *switch* yaitu 2,3% dan 2,4%.
3. Berdasarkan data pengujian yang sudah di dapatkan, bahwa pengujian *controller* POX pada penelitian kali ini di banding dengan penelitian dari jurnal terkait sebelumnya mendapatkan hasil yang lebih baik yaitu pada nilai *delay* pada penelitian ini masih di bawah 0,0050ms sedangkan sebelumnya sudah di atas 0,100ms, pada nilai *jitter* pada penelitian sebelumnya sudah di atas 0,013ms pada penelitian sebelumnya dan untuk *throughput* nilai kecepatan pada penelitian ini cukup jauh lebih tinggi di banding dengan data sebelumnya yaitu pada 8 *switch* saja data terkecil pada penelitian ini sebesar 916Mbits/sec sudah jauh di bandingkan data terbesar 9,537Bits/sec pada 8 *switch* penelitian sebelumnya.
4. Dari semua pengujian yang di lakukan dapat di simpulkan bahwa *controller* POX maupun RYU masih dalam range yang baik bahkan sangat baik bila di lihat dari parameter TIPHON, tetapi bila hanya melihat dari performa yang ada, *controller* RYU yang lebih unggul dari POX, melihat dari sisi pengukuran *delay*, *jitter* bahkan *throughput* yang cukup baik, tetapi itu semua dapat berubah

tergantung bagaimana konfigurasi yang di lakukan dan seberapa besar bandwidth internet yang kita gunakan.

Saran

Pada perkembangan selanjutnya, ada beberapa saran untuk memperbaiki tugas akhir ini yaitu :

1. Pengujian ini di lakukan menggunakan POX dan RYU, perlu di coba juga dengan tambahan controller yang lain dengan Bahasa pemrograman yang berbeda maka mungkin hasil yang berbeda dan lebih baik dapat terjadi di bandingkan dengan penelitian ini.
2. Dengan menambahkan variasi background trafik pada masing-masing topologi setiap controller yang di uji agar mendapat nilai yang kemungkinan lebih bervariasi

5. DAFTAR PUSTAKA

- [1] Abdillah. D, Yuliant. S, Izzatul. U. (2016). Perancangan dan analisis jaringan virtual berbasis software defined networking (sdn). Ilmu Komputasi. Fakultas Informatika. Universitas Telkom.
- [2] Rikie. K. (2014). Prototipe infrastruktur software-defined network dengan protokol openflow menggunakan ubuntu sebagai kontroler. Jurnal Dasi. STMIK AMIKOM Yogyakarta.
- [3] Anggara, S.M. (2015). Pengujian Performa Kontroler SDN: POX dan Floodlight. Institut Teknologi Bandung: Bandung.
- [4] Dewaweb team (2018, November 6). Pengertian jaringan komputer dan jenis-jenisnya. Diakses, November 26, 2019. Dari <https://www.dewaweb.com/blog/>
- [5] Niko (2014). Diakses, November 26, 2019. Dari <http://www.pintarkomputer.com/pengertian-kelebihan-kekurangan-tipe-jaringan-komputer-client-server-dan-peer-to-peer/>
- [6] Internet. (2019, Desember 19). Di *Wikipedia, Ensiklopedia Bebas*. Diakses pada, Desember 19, 2019, dari <https://id.wikipedia.org/w/index.php?title=Internet&oldid=16335247>
- [7] Protokol Internet. (2019, Desember 11). Di *Wikipedia, Ensiklopedia Bebas*. Diakses pada, Desember 11, 2019, dari <https://id.wikipedia.org/>
- [8] Aldwin, N. (2019, Oktober 14) ip address. dari <https://www.niagahoster.co.id/blog/>
- [9] Chung Yik, E.E. (2012). "Implementation Of An Open Flow Switch On Netfpga",Universiti Teknologi Malaysia 2012,. Dari http://portal.fke.utm.my/fklibrary/files/e-echungyik/2012/15_EECHUNGYIK2012.pdf
- [10] Naufal H. (Oktober 31, 2018). Diakses, November 10, 2019.
- [11] Dari <https://medium.com/core-network-laboratory-tech-page/mengenal-salah-satu-controller-dalam-sdn-706ed4624660>
- [12] (Pengertian wireshark. n.d.) <http://www.sibro21.org/2016/03/>
- [13] Aris C, Arif M & Mulyana E. (2014). *Pengantar software defined network*. Diakses September 5, 2019. <https://eueung.gitbooks.io/buku-komunitas-sdn-rg/content/>
- [14] Pemberton., D., Linton., A., Russell., S. (2014). RYU Openflow Cotroller. Diakses Oktober 10, 2019. <https://nsrc.org/workshops/2014/nznog-sdn/raw-attachment/wiki/WikiStart/Ryu.pdf>.
- [15] Wikipedia contributors. (2019, October 2). Iperf. In *Wikipedia, The Free Encyclopedia*. Retrieved, December 9, 2019, from <https://en.wikipedia.org/w/index.php?title=Iperf&oldid=919256624>
- [16] ("Overview Mininet," n.d.) from <http://mininet.org/overview/> Joesman, "TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks)," 2008.
- [17] Slepase. (2017, agustus 16). Pemahaman delay, jitter, packetloss dan throughput. <https://steemit.com/tutorial/>
- [18] Cendikia. (2019, maret 13). Pengertian Bandwith dan throughput. <https://www.selamatpagi.id/bandwidth-dan-throughput/>