ANALISIS PERFORMANSI *SOFTWARE DEFINED NETWORK* (SDN) CONTROLLER FLOODLIGHT, POX, RYU, DAN ODL PADA TOPOLOGI JARINGAN UNIVERSITAS ISLAM RIAU

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik pada Fakultas Teknik
Universitas Islam Riau

OLEH:
YOGA SAPUTRA
163510608

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS ISLAM RIAU
PEKANBARU
2021

KATA PENGANTAR

Alhamdulillahirabbil'alamin, segala puji bagi Allah Swt yang telah melimpahkan rahmat dan hidayah-Nya dan junjungan Nabi Muhammad SAW sehingga penulis dapat menyelesaikan Sekripsi yang berjudul ANALISIS PERBANDINGAN PERFORMANSI *SOFTWARE DEFINED NETWORK* (SDN) CONTROLLER FLOODLIGHT, POX, RYU, DAN ODL PADA TOPOLOGI JARINGAN UNIVERSITAS ISLAM RIAU. Yang merupakan salah satu syarat untuk pengajuan judul skripsi pada program Strata-1 di Jurusan Jaringan, Teknik Informatika, Fakultas Teknik, Universitas Islam Riau, Pekanbaru.

Penulis menyadari dalam penyusunan skripsi ini tidak akan selesai tanpa bantuan dari berbagai pihak. Karena itu pada kesempatan ini kami ingin mengucapkan terimakasih kepada:

1. Bapak Dr.Apri Siswanto, S.Kom., M.Kom., Selaku Ketua Program Studi Teknik Informatika Universitas Islam Riau.
2. Bapak Dr.Apri Siswanto, S.Kom., M.Kom., selaku dosen Jurusan Teknik Informatika yang telah memberikan saran untuk meneliti judul ini.
3. Segenap Dosen Program Studi Teknik Informatika Universitas Islam Riau Pekanbaru yang telah memberikan ilmunya kepada penulis.
4. Orang tua, saudara-saudara kami, dan Teman-teman seangkatan 2016 Kelas B Kakak-Kakak Senior Teknik Informatika, Fakultas Teknik yang bimbingan dan mensupport saya untuk dapata membuat proposal ini.

Kami menyadari proposal skripsi ini tidak luput dari berbagai kekurangan. Penulis mengharapkan saran dan kritik demi kesempurnaan dan perbaikannya sehingga akhirnya laporan proposal ini dapat memberikan manfaat bagi bidan pendidikan dan penerepan dilapangan serata bisa dikembangankan lagi lebih lanjut Amin.

Pekanbaru, 08 Desember 2021

Yoga Saputra

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi kini kian berkembang dengan pesat dan banyak hal yang baru didunia teknologi, tidak terkecuali pada jaringan komputer. Pada kemajuan dan perkembangan teknologi ini muncul lah ide baru atau konsep baru yaitu SDN. *Software Define Networking* adalah konsep pendekatan baru untuk melakukan perancangan, membangun dan mengelola jaringan komputer dengan memisahkan *control plane* dan *data plane*. Konsep utama pada *Software Define Networking* adalah sentralisasi kendali jaringan dengan semua pengaturan berada pada *control plane. Software Define Networking* juga mampu memberikan solusi untuk permasalahan-permasalahan jaringan yang ada sekarang ini, seperti sulitnya mengintegrasikan teknologi baru karena masalah perbedaan platform perangkat keras, kinerja yang buruk karena ada beberapa operasi yang berlebihan pada protokol layer dan sulitnya menyediakan layanan-layanan baru.

Jaringan statis konvensional sudah mulai digantikan dengan jaringan dinamis yang bersifat programmable. Software Defined Network (SDN) merupakan salah satu contoh dari perkembangan jaringan dinamis yang bersifat programmable dengan paradigma baru untuk memberikan kemudahan kepada pengguna dalam mendesain, membangun dan mengelola jaringan komputer. Software defined network menyediakan pendistribusian jaringan yang dikelola

secara terpusat (centralized) untuk memudahkan layanan jaringan agar lebih efisien, otomatis dan cepat.

Pada jaringan SDN, melakukan penyeimbang beban, arsitektur dan karakteristik dari SDN dengan protokol OpenFlow dan kontroler *floodlight*. Penerapan penyeimbang beban pada jaringan SDN melakukan pendekatan dimana pada setiap switch yang ada pada jaringan membagi trafik sesuai dengan aturan penanganan paket yang diberikan kontroler.

Openflow adalah arsitektur *software defined network* yang paling umum dingunakan. Beberapa sakelar jaringan dapat di kontorler oleh satu pengontrol *openflow* pengontrol terpusat. Pengontrol o*penflow* berbasis *PytHon* dan *Java* yang berbeda dan tersedia untuk jaringan yang ditentukan perangkat lunak. *OpenFlow* merupakan teknologi utama yang ada didalam *software defined network*, *OpenFlow* ini digunakan untuk menyediakan komunikasi diantara pengontrol SDN dan lapisan penerus lainnya, ia mempertahankan aliran lalu lintas pada jaringan. Dengan menggunakan Mininet Simulator itu dapat membangun jaringan yang luas dengan kumpulan elemen jaringan seperti *host* akhir, *router* berdasarkan kernel linux.

Software Define Networking adalah suatu metode atau paradigma yang memisahkan *forwarding plane* dan *control plane. Control plane* merupakan pengaturan atau kontorler pada perangkat jaringan. *Forwarding plane* merupakan media pengiriman paket ketujuan sesuai dengan tujuan pengirimanan. *Software Define Networking* merupakan konsep pendekatan baru untuk merancang, membagun dan mengelola jaringan komputer dengan memisahkan *control plane* dan *data plane.* Konsep yang ada pada *Software Define Networking* merupakan

sentralisasi jaringan dengan semua pengaturan berbeda pada *control plane*. Konsep ini sangat memudahkan operator jaringan dalam mengelola jaringan. *Software Define Networking* juga memberikan solusi pada permasalahan-permasalahan jaringan yang sekarang sulit mengintegrasikan teknologi baru. Pengontrol SDN mengoptimalkan aliran lalu lintas jaringan dengan membuat keputusan pada jalur tautan.

Beberapa Kontorler yang telah berkembang diantaranya *OpenDayLight*, POX, NOX, *Beacon*, Ryu, ONOS, *Floodlight, Maestro* dan masih ada beberapa lainnya yang terus dikembangkan. Dalam penelitian ini diambil dua contoh kontroler yaitu *OpenDayLight* yang merupakan salah satu kontorler enterprise dengan penerapannya menggunakan bahasa pemrograman Java, dan Ryu yang merupakan kontorler SDN yang bersifat terbuka (*open source*) dan penerapannya menggunakan bahasa pemrograman *Phyton*.

POX merupakan kontroler *platform* berbasis bahasa pemrograman *Python* yang akan digunakan untuk mengembangkan perangkat lunak yang ada didalam *Software Define Networking* kontorler. POX memiliki beberapa komponen yang bisa digunakan untuk membuat *Software Define Networking* kontorler sesuai dengan kebutuhan pengguna. Hal ini memungkinkan untuk pembuatan perangkat lunak SDN kontorler yang cocok untuk *load balancer* sebuah jaringan.

Floodlight merupakan SDN kontorler yang berkerja pada switch fisik dan virtual yang menggunakan *protocol OpenFlow*. *Floodlight* adalah *OpenFlow* kontorler berbasis *java* kelas *enterprise*, berlisensi *Apache*. Ini didukung oleh

komunitas pengembang termasuk sejumlah insinyur dari *Big Switch Networks*. *OpenFlow* adalah standar terbuka yang dikelola oleh *open networking foundation*. RYU merupakan kontorler yang ada di *Software-Define Networking* yang menyediakan komponen perangkat lunak denagan API. RYU juga mendukung berbagai protocol untuk mengelola perangkat jaringan, seperti *OpenFlow, Netconf, OF-config.*

OpenDayLight (ODL) adalah protokol SDN terbentuk atas konsorsium Linux *Foundation.* Kontorler ini didukung oleh standar *Northbound* API, *OpenFlow*, I2RS. Bebrapa fitur kontorler ini adalah Java-based (OSGI), *modular*, *pluggable* dan juga didukung oleh banyak protocol *sounthbound*. Tugas akhir ini akan menggunakan aplikasi emulator virtual jaringan mininet yang akan dijalankan di sistem operasi Ubuntu. Aplikasi ini berfungsi unutk membangun topologi jaringan untutk mensimulasikan topologi yang telah di rancang dan di aplikasi menggunakan aplikasi emulator mininet virtual. Dan kontorler *Software Define Networking* dapat di oprasikan ke operasi sistem Ubuntu dimana tiap-tiap kontorler *Software Define Networking Floodlight*, POX, RYU, DAN *OpenDayLight* (ODL) akan dibandingkan setiap kontorler menggunakan operasi sistem Ubuntu.

Tools yang sangat tepat digunakan untuk dalam mengawasi jaringan adalah *tools wireshark* dimana *tools* ini sangat berguna untuk *administrator* dalam memonitoring jaringan, *tools wireshark* mampu menangkap paket- paket data atau informasi yang berjalan dalam jaringan. Iperf sendiri bisa digunakan untuk mengukur performance link dari sisi TCP maupun UDP dan tool iperf untuk mengukur troughput bandwidth dalam sebuah link network.

1.2 Identifikasi Masalah

Berdasarkan permasalahan yang telah diuraikan diatas, maka rumusan masalah dalam pembuatan sistem ini adalah sebagai berikut :

1. Perancangan topologi jaringan mengunakan aplikasi emulator Mininet.
2. Membandingkan kualitas performa kontorler menggunakan *Floodlight*, POX, RYU, dan *OpenDayLight* (ODL).

1.3 Rumusan Masalah

Rumusan Masalah yang akan dibahas dibwah ini melitputi beberapa hal pokok yaitu :

1. Bagaiman merancang topologi menggunakan emulator Mininet?
2. Bagaimana cara membangun kontorler pada *Software Define Networking* POX, *Floodlight*, RYU, *OpenDayLight* (ODL) menggunakan emulator Mininet?
3. Bagaimana menguji kinerja SDN kontorler pada POX, *Floodlight*, RYU, *OpenDayLight* (ODL)?

1.4 Batasan Masalah

Batasan masalah yang akan dibahas dibawah ini meliput beberapa hal pokok yaitu :

1. Sistem operasi yang digunakan adalah Ubuntu Server 16.04 LTS
2. Pengujian kontorler Pox, *Floodlight*, RYU, *Open Day Light* dilakukan pada jaringan Teknik Informatika Universitas Islam Riau.

3. Parameter pengujian yang digunakan menggunakan *Throughput*, *Delay*, *Packet Loss*, dan *Jitter*.
4. Pengujian jaringan di lakukan dengan melakukan simulasi menggunakan mininet pada fakultas yang memeiliki *Access Point* yaitu Fakultas Teknik, Fakultas Pertanian.
5. Penelitian dan analisa memggunakan *wireshark* yang memonitoring jaringan dan dilakukan pengirim file data sebesar 1.21Gb kesetiap *host* menggunakan *tools iperf* yang akan dikirim dari beberapa *client* atau *host.*
6. Jumlah host yang dapat penulis terapkan didalam penelitian ini merupakan 22 host jika lebih dari 30 atau 50 host maka kontroler akan down dikarenakan spesifikasi laptop penulis yang digunakan masih standar.
7. *Tool* pengujian kinerja menggunakan aplikasi pengujian yaitu *iperf* dan *wireshak*.

Commented [y1]: Ditambahkan jumlah mahasiswa yang aktif di f.teknik dan pertanian
(Pak Yudhi)

1.5 Tujuan Penelitian

Penelitian ini bertujuan untuk mengukur kinerja dan menganalisa dari setiap kontorler POX, *Floodlight*, RYU, *Open Day Light* agar mendapakan hasil perandingan dari kontorler SDN untuk menjadi ucuan dalam penelitian.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat menjadi referensi atau panduan mengenai perbandingan kinerja kontorler POX, *Floodlight*, RYU, *OpenDayLight*. Dengan mengetahui kinerja kontorler POX, *Floodlight*, RYU, *OpenDayLight* diharapkan para pengelola server dapat menentukan kontorler yang sesuai kebutuhan.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penyusunan skripsi ini penulis juga melakukan studi kepustakaan yang merujuk kepada penelitian-penelitian sebelumnya yang berkaitan dengan penelitian yang penulis buat. Studi kepustakaan ini dilakukan sebagai bahan perbandingan dan referensi bagi penulis.

Dalam penelitian Hanifa, (2018), mengguji performa kontorler *software define network floodlight* vs onos. Penelitian ini menggunakan kontorler untuk analisis seberapa baiknya performa dari sebuah kontorler mengingat pentingnya fungsi dari kontorler. Pada uji *latency* dan *throughput* akan memperoleh informasi mengenai kemampuan mengeanai kontorler kemampuan kontorler tersebut. Hasil dari pengujian yang dilakukan pada kontorler *Floodlight* dan ONOS telah menunjukan hasil bahwasanaya kontorler memberikan nilai *latency* yang baik pada jumlah *switch* dibawah 60 *switch.*

Abdillah, Sibaroni, dan Ummah, (2016), mereka merancang dan menganalisis jaringan virtual berbasis *software define networking*. Penelitian ini menggunakan protokol *OpenFlow* yang dimana sebuah protokol atau standar komunikasi antarmuka yang berada antara *control* dan *forwarding layer*. Pada penelitian ini akan mensimulasikan jaringan SDN pada jaringan virtual. Simulasi jaringan virtual SDN ini menggunakan sebuah tool atau aplikasi yaitu Mininet. Mininet merupakan aplikasi yang berbasis light-weight virtualization yang dapat

menciptakan jaringan virtual yang realistik, menjalankan real kernel, switch dan kode aplikasi. Hasil penelitian menunjukkan konsep jaringan SDN berjalan, mengukur kinerja dari jaringan SDN seperti delay, jitter dan throughput dengan beberapa skenario topologi yaitu 2 switch, 4 switch, 8 switch dan 16 switch.

Krisandi, Virgono. Rumani (2018), meneliti efek penggunaan kontorler *floodlight* dan opendaylight pada performansi jaringan SDN. Penelitian ini membahas tentang *floodlight* dan *opendayligh* yang dimana penelitian ini menggunakan kontorler dengan menggunakan bahasa pemrograman java pada kontorler *Floodlight* dan *opendaylight* hasil penelitian ini pada *Floodlight* dengan algoritma Johnson dan OpenDaylight. Ketiga model kontorler ini akan diuji performanya menggunakan parameter QoS dengan standarisasi ITU-T G.1010. didapat kontorler *Floodlight* tanpa algoritma Johnson lebih unggul dibandingkan kedua kontorler lainnya dari untuk layanan data dan VoIP. Namun untuk pengiriman layanan video, kontorler *OpenDaylight* lebih baik dari yang lainnya. Karena hanya kontorler *OpenDaylight* yang bisa mengirimkan paket video. Untuk pengujian *resource utilization*, konsumsi memori tertinggi dimiliki oleh kontorler *OpenDaylight*. Berdasarkan standarisasi ITU-T G.1010 kontorler *Floodlight* tanpa algoritma Johnson, *Floodlight* menggunakan algoritma Johnson dan *OpenDaylight* hanya memenuhi standarisasi delay untuk layanan data.

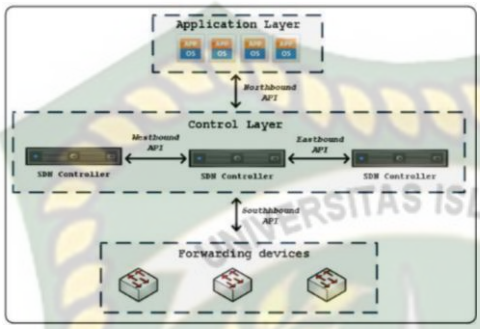
2.2 Dasar Teori

Berikut ini adalah beberapa dasar teori yang berkenaan dalam penelitian tugas akhir ini:

2.2.1 Software Defined Networking

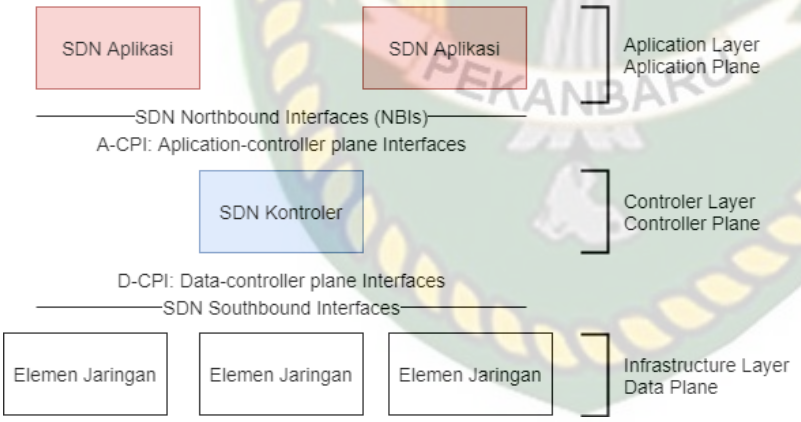
Software Defined Networking sebuah konsep pendekatan baru untuk mendesain, membangun dan mengelola jaringan komputer dengan memisahkan *control plane* dan *data plane*. Kontorler SDN yang bersifat *programmable* memungkinkan untuk mengimplementasikan suatu routing pada SDN. Routing merupakan penentuan rute terbaik yang akan dilalui informasi yang dikirim dari pengirim menuju penerima. Dalam *network, routing* biasanya menggunakan suatu algoritma pencarian jalur terpendek (Kurnia, 2017).

Software Defined Networking (SDN) merupakan sebuah paradigma baru dalam dunia jaringan yang memisahkan *control plane* dan *data plane*, baik secara logika ataupun secara fisik. *Control plane* merupakan bagian yang berfungsi sebagai "otak" dari sebuah jaringan, melakukan proses kalkulasi untuk fungsi seperti *routing, load balancer, intrusion detection system, topology learning,* dan lain sebagainya (Dewanto, 2018).



Gambar 2. 1 Arsitektur Jaringan SDN.

Pada gambar 2.1 terdapat arsitektur jaringan SDN, untuk menerapkan jaringan SDN tersebut dibutuhakn emulator mininet yang berfungsi sebagai forwarding plane dan kontorler yang berfungsi sebagai Control Plane yang dilengkapi oleh protokol openflow.



Gambar 2. 2 SDN Komponen.

Gambar 2.2 menunjukan arsitektur SDN yang terdapat tiga lapisan komponen, yaitu:

1. Infrastruktur: Terdiri dari elemen jaringan yang dapat mengatur SDN Datapath sesuai instruksi yang diberikan CDPI.
2. Kontorler: Entitas kontorler mentranslasikan kebutuhan aplikasi dengan infrastruktur dengan memberikan intruksi yang sesuai untuk SDN Datapath serta memberikan informasi yang relevan.
3. Aplikasi: Berada pada lapis teratas, berkomunikasi dengan sistem melalui north bounth interface (NBI).

2.2.2 POX

POX merupakan platform pengembangan open source untuk aplikasi *software defined network* (SDN) yang berdasarkan pada bahasa pemrograman *Python* dan merupakan kontorler *Openflow*. POX memungkinkan proses perancangan dan pembangunan jaringan yang lebih cepat, serta menjadi lebih umum digunakan daripada pendahulunya NOX (Putra, 2018).

POX adalah platform pengembangan sumber terbuka untuk aplikasi SDN dengan bahasa pemograman berbasis *Python*, seperti kontorler *OpenFlow* SDN. POX, memungkinkan pengembang dan pembuatan prototipe yang cepat, menjadi lebih umum digunakan daripada NOX, proyek sejenis. POX dapat menjalankan aplikasi yang berbeda seperti hub, switch, load balancer, dan firewall. Alat capture paket Tcpdump bisa digunakan untuk menangkap dan melihat paket yang mengalir

di antaranya POX kontorler dan perangkat *OpenFlow*. POX mempunyai komponen forwarding.l2_learning yang membuat switch OpenFlow bertindak sebagai jenis learning switch L2. Yang satu ini beroperasi seperti contoh "pyswitch" NOX, meskipun implementasinya sangat berbeda (Edgar, 2019).

2.2.3 *Floodlight*

Floodlight merupakan kontorler yang ada pada SDN yang memiliki bahasa pemograman berbasis java yang dikembangkan oleh *Big Swith Network floodlight* didukung oleh feature *OpenFlow* yang membuatnya bisa mengatur aliran data pada lingkungan SDN. *Floodlight* merupakan salah satu kontorler enterprise terbuka berlisensi *Apache* yang dikembangkan oleh komunitas pengembang di *Big Switch Network*. Pada penerapannya *floodlight* menggunakan bahasa pemrograman Java (Putra, 2018).

Kontorler *Floodlight* merupakan kontorler dengan kelas enterprise, memiliki lisensi *Apache*, dan memilik Bahasa pemrograman berbasis java yang dikembangkan oleh *Big Switch Network*. *Floodlight* didukung oleh feature *OpenFlow* yang membuatnya bisa mengatur aliran data pada lingkungan SDN (Krisandi, 2018).

2.2.4 RYU

Ryu adalah salah satu komponen yang ada di SDN. Ryu menyediakan perangkat lunak dengan API terderfinisi secara baik yang memudahkan pengembang untuk membuat manajemen jaringan baru dan aplikasi kontorler. Ryu juga mendukung beberapa protokol seperti *OpenFlow, Netconf, Of-config* (Edgar, 2019).

RYU sering disebut sebagai komponen dasar, perangkat lunak *open source* pada kerangka kerja jaringan (*networking framework*). RYU di implementasikan sepenuhnya dengan *Phyton*, dan didukung oleh laboratorium NTT. Seperti kontorler SDN lainnya, RYU juga menyediakan komponen perangkat lunak dengan API yang tak tersembunyi untuk memungkinkan para pengembang untuk membuat sebuah pengelolaan jaringan baru dan mengontrol aplikasi (Putra, 2018).

RYU kontorler merupakan salah satu jenis kontorler SDN yang menggunakan bahasa pemrograman *python* sebagai dasar pengembangannya. Ryu dapat berkomunikasi dengan *switch-switch* yang terhubung dengan menggunakan protokol *NETCONF*, *OF-Config*, *OpenFlow* 1.0,1.1,1.3, 1.4, dan 1.5. Jika menggunakan protokol *OpenFlow*, kontorler Ryu memiliki beberapa fungsi utama seperti:

1. *Packet-In Handler*: Digunakan untuk menerima *packet-in* dari setiap *switch*, untuk kemudian di ekstraksi datanya (contoh: ip *address* asal dan ip *address* tujuan) dan dilakukan kalkulasi yang dibutuhkan.
2. *Datapath.send_msg*: Digunakan untuk mengirimkan instruksi terhadap *switch* melalui protokol *OpenFlow*. Dua instruksi yang sering adalah

OFPTrafikMod untuk memperbaharui trafik tabel pada *switch*, dan *OFPTrafikStatsRequest* untuk meminta kondisi jaringan dari setiap *switch*.

3. *EventSwitchEnter dan EventLinkAdd*: Digunakan untuk mengetahui identitas *switch* dan pemetaan *link* antar *switch* pada saat *switch* pertama kali terhubung dengan kontorler.
4. *TrafikStatsReply Handler*: Digunakan untuk menampung laporan kondisi jaringan setelah mengirim *TrafikStatsRequest*.

2.2.5 *Opendaylight*

OpenDayLight adalah pengontrol SDN yang dikembang oleh komunitas pengembang terbuka LINUX. Kontorler ini didukung oleh standar *northbound* API, jadi tidak hanya didukung dengan protocol *southbound* API seperti *OpenFlow*, I2RS dan NETCONF yang bisa diprogram. Beberapa fitur kontorler ini adalah Java-Based (OSGI), modular, pluggable dan juga didukung oleh banyak protokol *southbound. OpenDayLight* dimanfaatkan dalam jenis jaringan bersekala besar (Edgar, 2019).

OpenDaylight terbentuk atas konsorsium Linux *Foundation*. Kontorler ini didukung oleh standar *northbound* API, jadi tidak hanya didukung dengan protokol *southbound* API seperti *OpenFlow*, I2RS dan NETCONF yang bisa diprogram. Beberapa fitur kontorler ini adalah *Java-based* (OSGI), modular, *pluggable* dan juga didukung oleh banyak protokol *southbound*. *OpenDaylight* dimanfaatkan dalam jenis jaringan berskala besar (Krisandi, 2018).

2.2.6 Parameter Pengujian

Melakukan pengujian dan analisis terhadap performansi *througput* dan *latency* dari masing-masing kontorler yang diuji. Pada penelitan ini terdapat beberapa scenario untuk menguji performansi kontorler dan menganalisis hasil yang diperoleh dari simulasi dengan parameter yang meliputi rata-rata *througput* dan *latency* (Putra, 2018).

Wireshark adalah penganalisis paket gratis dan sumber terbuka. Perangkat ini digunakan untuk pemecahan masalah jaringan, analisis, perangkat lunak dan pengembangan protokol komunikasi, dan pendidikan (Wicaksono, 2009).

Untuk mengetauhi kinerja kontroler dari sistem SDN yang telah dirancang maka akan dilakukan pengujian dengan parameter-parameter yaitu QoS dan resource utilization. Untuk pengujian oleh QoS menggunakan topologi yang sudah disediakan. Hal ini dilakukan untuk mengetahui kinerja kontorler, pengujian QoS ini juga melibatkan parameter *delay, jitter, throughput,* and *packet loss*. Percobaan akan dilakukan dengan trafik UDP yang kita ambil datanya. Data yang diambil berupa data dari D-ITG. Untuk pengujian *resource utilization* mengetahui komsumsi memori kontorler selama kontorler dijalankan. Untuk mengetahui komsumsi memori dari kontorler dapat diambil dari sistem ubuntu itu sendiri. Pada pengujian kali ini akan banyak variasi pengujian dengan *background traffic* dengan bantuan ipref. Terdapat jenis 5 background *traffic* yang digunakan yaitu 0 Mbps, 50Mbps, 100 Mbps, 150Mbps dan 200 Mbps. Nilai dari parameter QoS yang didapat akan dibandingkan dengan nilai yang telah di tetapkan oleh badan

standarisasi ITU- T. sehingga dapat diketahui nilai dari parameter QoS tersebut sudah memenuhi standarisasi atau belum (Edgar, 2019).

Ada beberapa parameter yang dapat diukur dalam sebuah pengujian yaitu:

1. *Throughput*, adalah kecepatan transfer sebenarnya antara *server* ke *client* dan *client* ke *server*, misalnya seseorang mendownload dari sebuah *server*, dan kecepatan internetnya 100 kbps, kecepatan *server* 1 mbps maka kecepatan *troughput* sebenarnya adalah 100 kbps. Jika *bandwidth* yang dihasilkan lebih baik maka *server* tersebut mampu menampung banyak data.

$$Throughput = \frac{Jumlah\ Data\ Yang\ Diterima}{Waktu\ Pengiriman\ Data} \quad (2.\ 1)$$

2. *Delay*

   Delay adalah waktu tunda sebuah proses pengiriman data dari satu titik sumber ke titik tujuan. Pengiriman paket yang satu dengan yang lain memiliki waktu *delay* yang berbeda, sehingga waktu *delay* tidak dapat diprediksi dengan pasti. Olehkarena itu waktu *delay* dapat dipandang sebagai variabel random. *Delay* dalam sebuah proses transmisi paket dalam sebuah jaringan komputer disebabkan karena adanya antrian yang panjang, atau mengambil rute lain untuk menghindari kemacetan pada routing.

$$Rata - rata\ Delay = \frac{Total\ Delay}{Total\ Paket\ Yang\ Diterima} \quad (2.\ 2)$$

3. *Paclet Loss*

Packet loss merupakan besarnya kemungkinan jumlah paket data yang hilang pada saat transmisi pada dalam jaringan. Hal ini disebabkan oleh banyak faktor seperti penurunan sinyal dalam media jaringan, kesalahan perangkat keras jaringan, atau juga radiasi dari lingkungan sekitar. Pada beberapa network transfer protocol seperti TCP yang bersifat connection oriented, menyediakan pengiriman kembali (restransmission) atau pengiriman secara otomatis (resends) paket yang hilang selama proses transmisi walau segmen telah tidak diakui.

$$Packet\ Loss = \frac{(Paket\ Data\ Dikirim - Paket\ Data\ Diterima - 1)}{Paket\ Data\ Yang\ Dikirim} x100\% \qquad (2.\ 3)$$

4. *Jitter*

Jitter adalah variasi dari *delay* atau selisih antara *delay* pertama dengan *delay* selanjutnya. Jika variasi *delay* dalam transmisi terlalu lebar, maka akan mempengaruhi kualitas data yang ditransmisikan.. Jika *buffer jitter* tersedia lebih banyak, maka jaringan dapat mereduksi efek dari *jitter*. Contoh dari jitter, misalnya hasil ping menunjukkan *delay* dengan rentang 2ms, 4ms, 7ms. Maka *jitter* dapat dihitung dengan mengurangi *delay* akhir dengan *delay* sebelumnya, seperti contoh *jitter* adalah 7ms-4ms=3ms. Untuk mengukur *jitter* dapat kita gunakan fasilitas UDP test pada perangkat lunak *iperf*.

$$Jitter = \frac{Total\ variasi\ delay}{Total\ Paket\ yang\ diterima - 1} \qquad (2.\ 4)$$

BAB III

METODOLOGI PENELITIAN

3.1 Alat dan Bahan Penelitian

Adapun spesifikasi perangkat keras (*hardware*) yang digunakan untuk melakukan pengujian dan spesifikasi perangkat lunak (*software*) yang dibutuhkan untuk sistem yang akan dibangun adalah sebagai berikut:

3.1.1 Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras (*hardware*) yang dibutuhkan sebagai *server* yang akan digunakan dalam penelitian ini adalah sebagai berikut:

A. *Personal Computer*

a. Prosessor setara AMD A10.

b. RAM 4 GB.

c. *Harddisk* minimal 500 GB.

d. *Type System* 64-bit *Operating System.*

B. Perangkat Jaringan

Kabel UTP (*Unshielded Twisted Pair*) untuk menghubungkan antar PC (*Personal Computer*) *client* dan PC (*Personal Computer*) server.

3.1.2 Perangkat Lunak (*Software*)

Perangkat lunak (*Software*) yang digunakan dalam pembuatan pengujian kinerja ini sebagai berikut:

1. Sistem Operasi: Ubuntu server 16.04.
2. Aplikasi untuk mengukur kinerja jaringan pada kontrol SDN menggunakan *Iperf*.
3. Aplikasi Mininet digunakan untuk mengoprasikan topology jaringan yang akan dibangun.

3.1.3 Perangkat

Kabel UTP (*Unshielded Twisted Pair*) untuk menghubungkan antar PC (*Personal Computer*) *client* dan PC (*Personal Computer*) server.

3.1.4 Teknik Pengumpulan Data

Pengumpulan data merupakan langkah yang penting untuk mendapatkan data yang benar dan menyakinkan agar hasil yang didapat tidak menyimpang dari tujuan yang diharapkan sebelumnya, maka dari itu penulis melakukan langkah-langkah penelitian sebagai berikut.

1. Analisis.

Didalam tahapan ini akan dirancang sebuah proses pengiriman data pada simulator mininet dari server menuju *clinet*, sehingga dapat diketahui parameter-parameter antara lain *throughput, delay, packet loss,* dan *jitter* sehingga kesimpulan akan di ambil dari hasil perbandinggan yang sudah diperoleh.

2. Perancangan

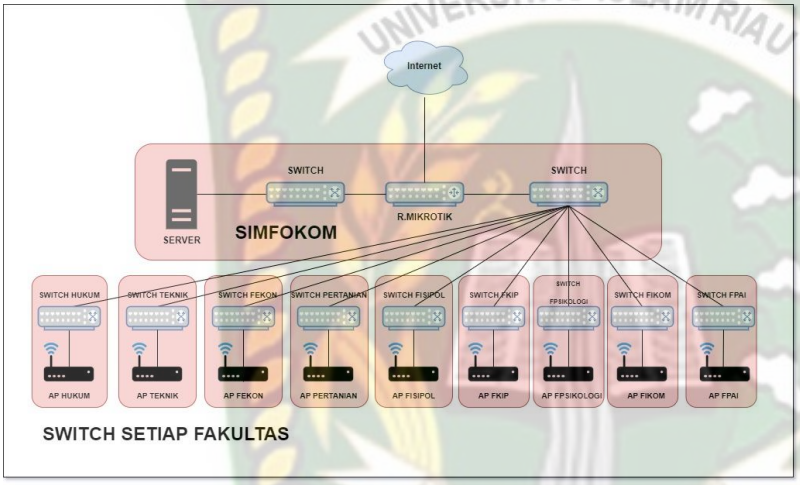
Tahap ini akan merancang analisa spesifikasi kebutuhan yang telah di dapat ke dalam bentuk arsitektural perangkat lunak, untuk diimplementasikan kepada kontorler SDN yang akan dibangun.Pengujian Dalam tahap ini penulis melakukan pengujian kinerja kontorler Pox, *Floodlight*, RYU, dan *OpenDayLight* (ODL) pada topologi jaringan yang akan dibangun melalui virtual menggunakan aplikasi mininet yang akan di oprasikan di Ubuntu Server untuk mendapatkan hasik dari pengujian yang sedang berjalan. Namun unutk melakukan pengujian kontorler SDN penulis menggunakan *tool iperf* untuk melihat *throughput*, *delay*, *packet loss*, dan *jitter* pada kontorler SDN.

3. Dokumentasi

Pada peroses dokumentasi, penulis juga melakukan studi pustaka, membaca dan mempelajari dokumen-dokumen, buku-buku acuan, serta sumber lainnya yang berkaitan dengan penelitian untuk dijadikan referensi.

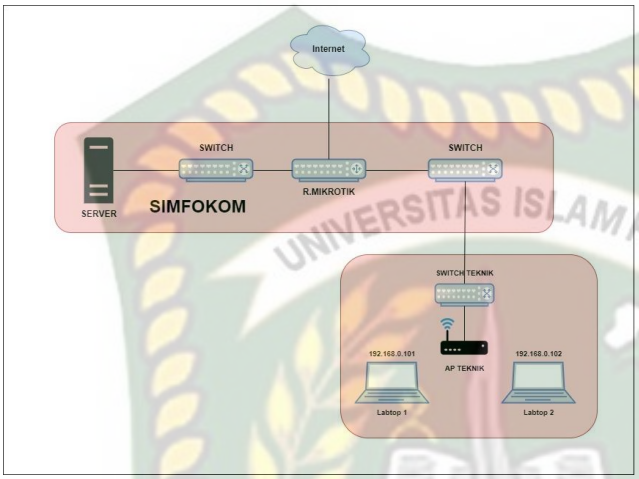
3.2 Pengembangan dan Perancanggan Sistem

Pada peroses pengujian akan menggunakan jaringan LAN (*Local Area Network*) yang sederhana untuk mensimulasikan komunikasi antar *server* dan *client*.



Gambar 3. 1 Topologi Star UIR.

Dari gambar 3.1 diatas, merupakan topologi UIR dengan menggunakan jenis jaringan topologi *star*, dimana pada gambar 3.1 jaringan topologi *star* diatas terdiri dari *Switch* dari fakultas teknik, bait, fakultas ekonomi, dan fakultas keguruan dan ilmu pendidikan dan ditangkap oleh *Acces Point* yang akan disebarkan ke setiap fakultas.



Gambar 3. 2 Topologi Logic.

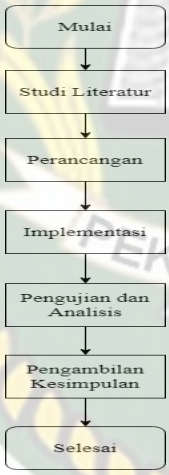
Topologi *logic* merupakan topologi yang akan menggambarkan hubunggan jaringan secara logika yang akan terjadi didalam masing-masing komputer. Pada gambar 3.2 diatas, merupakan logika jaringan yang akan terjadi pada jaringan UIR yang akan menghubungkan antara masing-masing komputer. Dalam pengujian yang akan dilakukan dengan mengkoneksikan setiap komputer dengan menggunakan virtual pada mininet.

3.3 Skenario Pengujian

Aanalisis perbandingan kontroler *Software Define Networking* (SDN) menggunkan *software* pengujian *Iperf* ini melalui beberapa tahapan penelitian yang akan dijadikan prosuder penelitian. Tahapan prosedur yang akan digambarkan dapat dilihat pada beberapa tahapan dibawah ini:

1. Metodologi Penelitian.

Metodologi penelitian pada perancanggan penelitian untuk mendapatkan data-data pada pengujian yang akan dilakukan dengan menggunakan kontorler pada *Software Define Networking* (SDN).



Gambar 3. 3 Metodologi Penelitian.

Berdasarkan gambar 3.2 diatas mengambarkan dalam proses penelitian ini dilakukan pengujian dengan tahapan-tahapan penelitian yang terkait serta relevan terhadap penelitian yang akan dilakukan penulis.

Berikut ini adalah penjelasan lebih lengkap mengenai metodologi penelitian sesuai dengan gambar 3.2:

1) Tahapan Literatur

Dibahas mengenai dasar teori dan penelitian-penelitian yang terkait dalm penelitian penulis.

2) Tahapan Perancanggan

Pada tahap ini penulis melakukan perancangan yang akan dilakukan dalam penelitian yang mana perancang tersebut sebagai acuan dalam melakukan penelitian.

3) Tahapan Implementasi

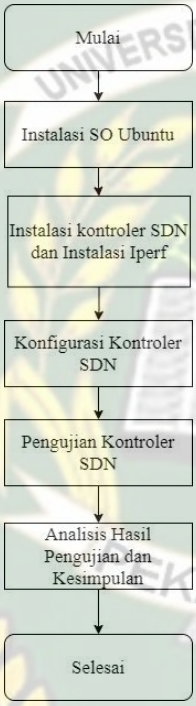
Setelah tahapan perancangan dilalui maka penulis melanjutkan pada tahapan implementasi. Dalam tahapan ini penulis mengimplementasikan tahapan perancangan seperti membangun topologi, melakukan konfigurasi pada *Software Define Networking* (SDN) dan lain sebagainya.

4) Tahapan Pengujian / Analisis

Tahapan terakhir penulis melakukan pengujian untuk mengetahui hasil kinerja dari setiap Kontorler yang diterapkan pada *Software Define Networking* (SDN). Hasil pengujian maka diambil kesimpulan.

2. Skema Rancangan Alur Pengujian Kontroler SDN.

Skema rancangan alur pengujian yang akan diterapkan pada penelitian ini dengan rancanggan alur yang akan diterapkan pada penelitian.



Gambar 3. 4 Rancangan Alur Pengujian.

Skema rancangan alur pengujian diatas mengambarkan alur perancangan dan pengujian yang akan dilakukan mulai dari instalasi sistem operasi ubuntu akan diteruskan ke-instalasi kontroler *Software Define Networking* (SDN) dan sistem pengguji Hittper, akan diteruskan melakukan konfigurasi Kontorler *Software Define Networking* (SDN), setelah melakukan tahapan-tahapan diatas maka akan dilakukan pengujian terhadap setiap kontroler akan mendapatkan hasil dari pengujian pada kontroler maka dapatlah kesimpulan dari hasil penelitian.

3.4 Parameter Pengujian

Para meter pengujian yang digunakan meliputi *Throughput, Delay, Packet Loss,* dan *Jitter*. Berikut beberapa penjelasan tentan parameter tersebut:

1. *Throughput*

Throughput adalah bandwith yang sebenarny pada proses transaksi dalam jaringan, kecepatan (rute) transfer data efektif yang diukur dalam bps. Semakin besar nilai *throughput* dalam jaringan, maka semakin bagus kinerja dari suatu *server*.

2. *Delay*

Delay adalah waktu tunda sebuah proses pengiriman data dari satu titik sumber ke titik tujuan. Pengiriman paket yang satu dengan yang lain memiliki waktu *delay* yang berbeda, sehingga waktu *delay* tidak dapat diprediksi dengan pasti. Olehkarena itu waktu *delay* dapat dipandang sebagai variabel random.

3. *Packet Loss*

Packet loss merupakan besarnya kemungkinan jumlah paket data yang hilang pada saat transmisi pada dalam jaringan. Pada beberapa network transfer protocol seperti TCP yang bersifat connection oriented, menyediakan pengiriman kembali (restransmission) atau pengiriman secara otomatis (resends) paket yang hilang selama proses transmisi walau segmen telah tidak diakui.

4. *Jitter*

Jitter adalah variasi dari *delay* atau selisih antara *delay* pertama dengan *delay* selanjutnya. Jika variasi *delay* dalam transmisi terlalu lebar, maka akan mempengaruhi kualitas data yang ditransmisikan.

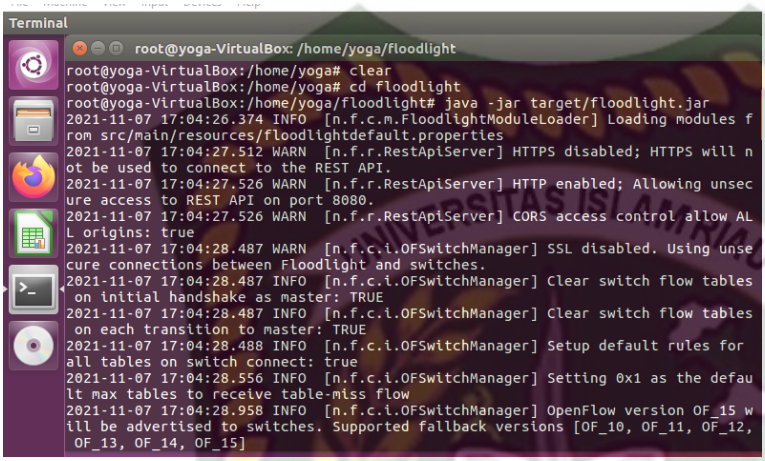
BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

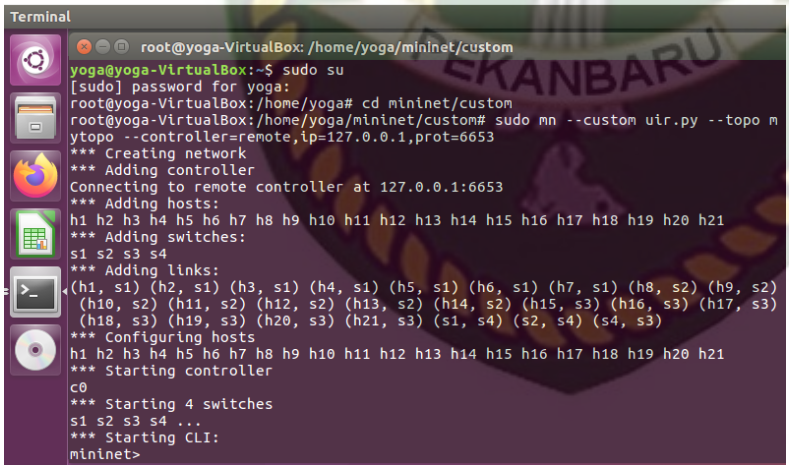
Berdasarkan analisa dan rancangan yang telah dilakukan pada bab sebelumnya. Maka perlu dilakukan berbagai pengujian untuk mengetahui hasil perbandingan kinerja kontroler *floodlight*, RYU, POX, dan *opendaylight* untuk mengetahui antar kontroler yang mana lebih unggul dalam melakukan pengiriman data. Pengujian dilakukan menggunakan simulator *mininet* untuk merancang topologi dan melakukan pengujian menggunakan iperf. Tahapan pengujian dengan melakukan testing ke semua perangkat yang dibutuhkan sesuai topologi yang dirancang dengan menggunakan mininet.

Parameter pengujian yang dilakukan pada pengujian kontroler SDN melakukan analisa menggunakan *wireshark* di antaranya adalah *Throughput, PacketLoss, Delay, Jitter*. Pengujian dilakukan dengan menggunakan simulator *mininet* antara kontroler dengan melakukan pengiriman data dari *clinet* ke *clinet* dengan menggunakan *tools iperf* dengan waktu pengiriman yang dilkaukan dari 0.0-101.3 sec di antara 1 atau 2 menit dalam peroses pengiriman data sebesar 10 .GB dengan bandwidth diantaranya 833 Mbits/sec, 808 Mbits/sec, 872 Mbits/sec, dan 1.11 Gbits/sec.



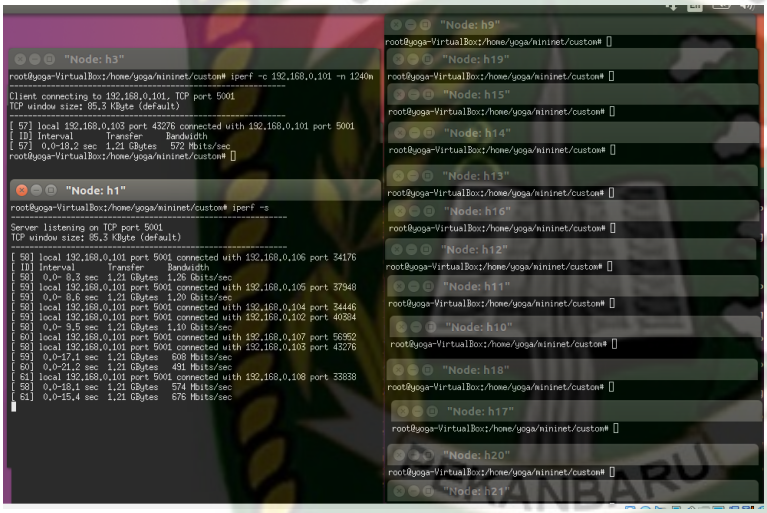
Gambar 4. 1 Menjalankan Kontroler

Pada gambar 4.1 merupakan menjalankan kontroler yang akan dihubungkan ke simulator mininet untuk melakukan pengujian pada kontroler.



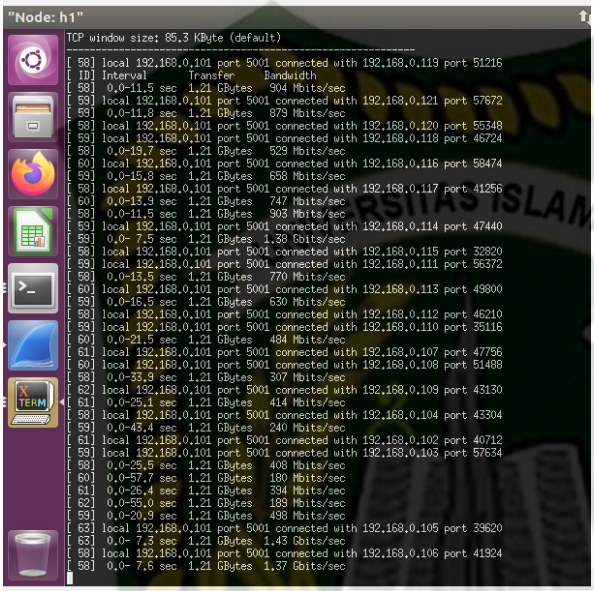
Gambar 4. 2 Rancangan Topologi *Mininet*.

Pada gambar diatas merupakan rancangan topologi dengan menggunakan mininet yang telah dihubungkan ke kontroler yang akan penulis uji dan melakukan testing perangkat pada rencangan topologi di simulator mininet.



Gambar 4. 3 Pengujian Menggunakan Iperf.

Gambar 4.2 merupakan pengujian kontroler yang memiliki 22 host yang diantaranya menjadi host server yaitu *host* 1. Saat melakukan pengujian pengiriman paket data sebesar 1,21 GB yang dikirim dari dari *host* 2 sampai dengan *host* 22 ke *host* 1 yang menjadi server dalam pengujian kontroler untuk mendapatkan hasil dari setiap pengiriman maka akan mendapatkan bandwitch dari setiap *host* dalam pengiriman paket data.



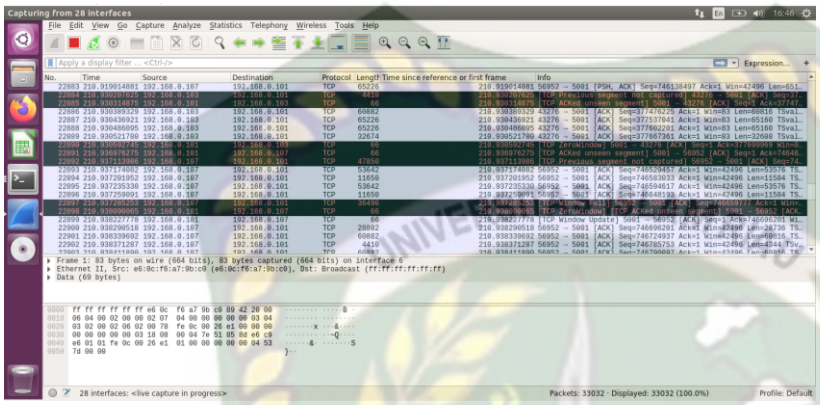
Gambar 4. 4 Hasil Pengiriman Paket data.

Dari gambar 4.4 merupakan hasil dari pengiriman paket data sebesar 1.21GB dari *host* 2 sampai dengan *host* 22 yang dikirim ke *host* 1 sebagai *host* server maka hasil pengiriman dari setiap kontroler akan mendaptakn *interveal*, *transfer* dan *bandwidth*.

Berikut tabel hasil *bandwitch* dari proses pengiriman data menggunakan kontroler *floodlight, opendaylight*, RYU, dan POX :

Tabel 4. 1 Tabel Bandwitch.

Floodlight	POX	Open Day light	RYU
621 Mbits/sec	634 Mbits/sec	1.68 Gbits/sec	966 Mbits/sec
121 Mbits/sec	362 Mbits/sec	1.47 Gbits/sec	1.68 Gbits/sec
237 Mbits/sec	559 Mbits/sec	1.36 Gbits/sec	1.16 Gbits/sec
125 Mbits/sec	1.14 Gbits/sec	1.26 Gbits/sec	633 Mbits/sec
145 Mbits/sec	1.56 Gbits/sec	908 Mbits/sec	1.10 Gbits/sec
94.7 Mbits/sec	1.68 Gbits/sec	591 Mbits/sec	859 Mbits/sec
131 Mbits/sec	1.86 Gbits/sec	768 Mbits/sec	1.14 Gbits/sec
147 Mbits/sec	1.07 Gbits/sec	1.00 Gbits/sec	2.19 Gbits/sec
75.6 Mbits/sec	485 Mbits/sec	568 Mbits/sec	887 Mbits/sec
116 Mbits/sec	460 Mbits/sec	208 Mbits/sec	1.89 Gbits/sec
68.8 Mbits/sec	393 Mbits/sec	626 Mbits/sec	1.89 Gbits/sec
87.8 Mbits/sec	237 Mbits/sec	1.18 Gbits/sec	830 Mbits/sec
119 Mbits/sec	430 Mbits/sec	858 Mbits/sec	1.63 Gbits/sec
72.3 Mbits/sec	372 Mbits/sec	631 Mbits/sec	1.68 Gbits/sec
175 Mbits/sec	214 Mbits/sec	974 Mbits/sec	1.01 Gbits/sec
150 Mbits/sec	174 Mbits/sec	1.15 Gbits/sec	977 Mbits/sec
72.5 Mbits/sec	360 Mbits/sec	1.40 Gbits/sec	1.10 Gbits/sec
86.5 Mbits/sec	723 Mbits/sec	1.04 Gbits/sec	835 Mbits/sec
97.0 Mbits/sec	151 Mbits/sec	890 Mbits/sec	1.77 Gbits/sec
108 Mbits/sec	150 Mbits/sec	1.06 Gbits/sec	1.22 Gbits/sec
115 Mbit/sec	668 Mbits/sec	1.61 Gbit/sec	1.61 Gbits/sec



Gambar 4. 5 *Tools Wireshark.*

Wireshark adalah penganalisis paket bebas dan sumber terbuka. *Wireshark* digunakan untuk memecahkan masalah jaringan, analisis, perangkat lunak dan yang lainnya. Dengan menggunakan *wireshark* dapat menganalisis data-data guna untuk menjadi perbandinggan dari antara kontroler.

Statistics

Measurement	Captured	Displayed
Packets	114270	111797 (97.8%)
Time span, s	409.579	200.997
Average pps	279.0	556.2
Average packet size, B	10481	10711
Bytes	1197662680	1197448122 (100.0%)
Average bytes/s	2.924 k	5.957 k
Average bits/s	23 M	47 M

Gambar 4. 6 Hasil *Statistic*s Pengujian Kontroler *Floodlight.*

a. Pengujian pertama *Troughput* pada kontroler *floodlight* berdasarkan *capture wireshark.*

Troughput = 1197662680 / 409.579

Troughput = 2.924,131071173083 bytes * 8

Troughput = 23.393 bps

Troughput = 23 bps

b. Pengujian pertama *Delay* pada kontroler *floodlight* berdasarkan *capture wireshark.*

Rata-rata *Delay* = 200.997 / 111797

Rata-rata *Delay* = 1,797874719357407 * 1000

Rata-rata *Delay* = 1.798 ms

Rata-rata *Delay* = 0.001798 sec

c. Pengujian pertama *Jitter* pada kontroler *floodlight* berdasarkan *capture wireshark.*

Jitter = -129,860747 /111798

Jitter = -0,00116 ms

Jitter = -0,00000116 sec

d. Pengujian pertama *Packet Loss* pada kontroler *floodlight* berdasarkan *capture wireshark.*

Packet Loss = ((114270 - 4821 = 109.449) 109.449 * 100 = 10.944.900)

Packet Loss = 10.944.900 / 114270

Packet Loss = 95,78

Packet Loss = 96 %

Statistics

Measurement	Captured	Displayed
Packets	116121	111435 (96.0%)
Time span, s	487.250	238.474
Average pps	238.3	467.3
Average packet size, B	23175	24147
Bytes	2691134575	2690790658 (100.0%)
Average bytes/s	5.523 k	11 M
Average bits/s	44 M	90 M

Gambar 4. 7 Hasil *Statistics* Pengujian Kontroler POX.

a. Pengujian pertama *Troughput* pada kontroler POX berdasarkan *capture wireshark.*

Troughput = 2691134575 / 487.250

Troughput = 5.523,108414571575 byte * 8

Troughput = 44.184bps

Troughput = 44 bps

b. Pengujian pertama *Delay* pada kontroler POX berdasarkan *capture wireshark.*

Rata-rata *Delay* = 238.474 / 111435

Rata-rata *Delay* = 2,140027818907884 * 1000

Rata-rata *Delay* = 214 ms

Rata-rata *Delay* = 0.214 sec

c. Pengujian pertama *Jitter* pada kontroler POX berdasarkan *capture wireshark.*

Jitter = -39,079422 / 111434

Jitter = -3,507 ms

Jitter = -0,003507 sec

d. Pengujian pertama *Packet Loss* pada kontroler POX berdasarkan *capture wireshark.*

Packet Loss = ((116121 - 5905 = 110.216)110.216 *100 = 11.021.600)

Packet Loss =11.021.600 / 116121

Packet Loss = 94,9

Packet Loss = 95 %

Statistics

Measurement	Captured	Displayed
Packets	92948	91706 (98.7%)
Time span, s	370.758	186.148
Average pps	250.7	492.7
Average packet size, B	34188	34650
Bytes	3177701994	3177601756 (100.0%)
Average bytes/s	8.570 k	17 M
Average bits/s	68 M	136 M

Gambar 4. 8 Hasil *Statistics* pengujian kontroler *Open Day Light.*

a. Pengujian pertama *Troughput* pada kontroler *open day light* berdasarkan *capture wireshark.*

Troughput = 3177701994 / 370.758

Troughput = 8.570,825158189439 byte * 8

Troughput = 68.566 bps

Troughput = 68 bps

b. Pengujian pertama *Delay* pada kontroler *open day light* berdasarkan *capture wireshark.*

Rata-rata *Delay* = 186.148 / 91706

Rata-rata *Delay* = 2,029834471026978 * 1000

Rata-rata *Delay* = 2.029 ms

Rata-rata *Delay* = 0,002029 sec

c. Pengujian pertama *Jitter* pada kontroler *open day light* berdasarkan *capture wireshark.*

Jitter = -0,376484 / 91705

Jitter = -4,105 ms

Jitter = -0,004105 sec

d. Pengujian pertama *Packet Loss* pada kontroler *open day light* berdasarkan *capture wireshark.*

Packet Loss = ((92948 - 7645 = 85.303) 85.303 * 100 = 8.530.300)

Packet Loss = 8.530.300 / 92948

Packet Loss = 91,77

Packet Loss = 92 %

Statistics

Measurement	Captured	Displayed
Packets	105407	101941 (96.7%)
Time span, s	596.850	403.881
Average pps	176.6	252.4
Average packet size, B	28334	29293
Bytes	2986558026	2986183622 (100.0%)
Average bytes/s	5,003 k	7,393 k
Average bits/s	40 M	59 M

Gambar 4. 9 Hasil *Statistic*s Pengujian Kontroler RYU.

a. Pengujian pertama *Troughput* pada kontroler RYU berdasarkan *capture wireshark.*

Troughput = 2986558036 / 596.850

Troughput = 5.003,867028566642 byte * 8

Troughput = 40.030 bps

Troughput = 40 bps

b. Pengujian pertama *Delay* pada kontroler RYU *capture wireshark.*

Rata-rata *Delay* = 403.881 / 101941

Rata-rata *Delay* = 3,96190933971611 * 1000

Rata-rata *Delay* = 3.962 ms

Rata-rata *Delay* = 0,003962 sec

c. Pengujian pertama *Jitter* pada kontroler RYU berdasarkan *capture wireshark.*

Jitter = -0,643127 / 101940

Jitter = -6,3088 ms

Jitter = 0,0063088 sec

d. Pengujian pertama *Packet Loss* pada kontroler RYU berdasarkan *capture wireshark.*

Packet Loss = ((105407 - 5666 = 99.741) 99.741 * 100 = 9.974.100)

Packet Loss = 9.974.100 / 105407

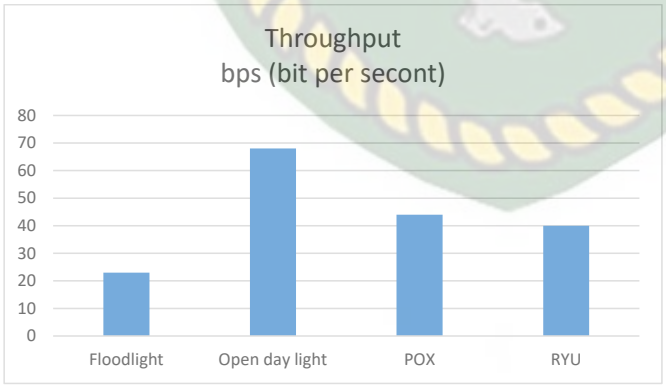
Packet Loss = 94,624

Packet Loss = 95 %

Berikut tabel hasil dari proses pengiriman data menggunakan kontroler *floodlight, opendaylight*, RYU, dan POX :

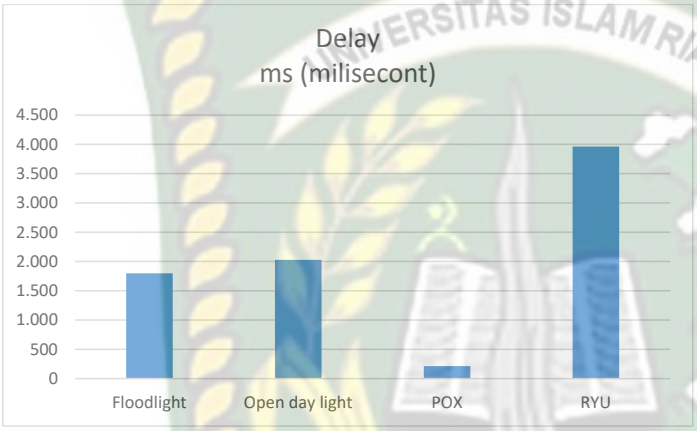
Tabel 4. 2 Tabel Hasil Pengujian.

	Throughput	*Delay*	*PacketLoss*	*Jitter*
Floodlight	23	1.798	96,00%	-0,00116
Opendaylight	68	2.029	92%	-4,105
POX	44	214	95%	-3,507
RYU	40	3.962	95%	-6,3088



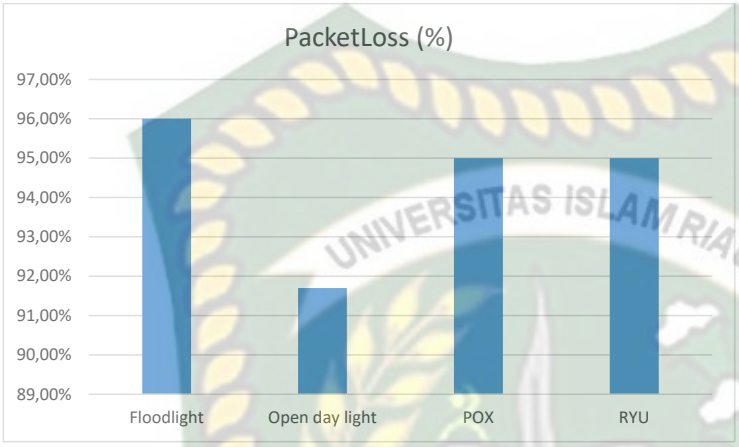
Gambar 4. 10 Grafik Pengujian *Throughput.*

Terlihat pada gambar diagram 4.4 dari hasil pengujian *throughput*. Kontroler yang unggul dalam pengujian *throughput* merupakan kontroler *open day light* dengan kecepatan pengiriman 68 bps.



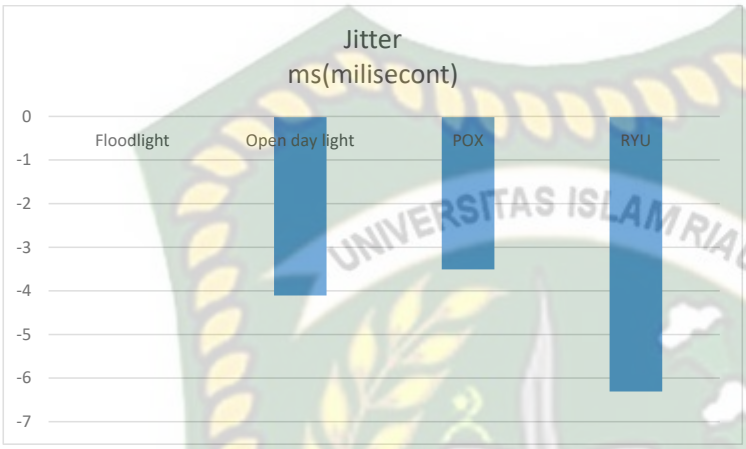
Gambar 4. 11 Grafik Pengujian *Delay*.

Terlihat pada gambar diagram 4.5 hasil pengujian *delay* pada kontroler degan waktu yang dibutuhkan data untuk menempuh jarak ke tujuan dengan rentan waktu yang cepat iyalah kontroler POX dengan kecepatan menempuh jarak pengiriman ke tujuan iyalah 214 ms.



Gambar 4. 12 Grafik Pengujian *Packet Loss.*

Terlihat pada gambar diagram 4.6 hasil pengujian *Packet loss* dengan menggunakan kontroler yang telah diuji. Dengan jumlah total paket yang hilang lebih sedikit dari ke-empat kontroler yang di uji merupakan kontroler *floodlight* dengan persenan paket data yang hilang lebih sedikit merupakan 96% paket yang hilang.



Gambar 4. 13 Grafik Pengujian *Jitter*.

Terlihat pada gambar diagram 4.7 hasil grafik pengujian *jitter*. Jitter atau variasi kedatangan paket dengan antrian pengiriman dan pengolahan data hingga penghimpunan ulang paket-peket diakhir perjalanan pengiriman yang lebih cepat pengiriman merupakan kontroler *floodlight* dengan rentang antrian pengiriman paket -0,00116 ms.

4.2. Pembahasan

Dari hasil penulusuran pengujian pengiriman dan pengunduhan data pada kontroler dapat penulis lihat hasil kinerja dari ke empat kontorler yang berbeda-beda. Dapat penulis simpulkan bahwa antara ke empat kontorler yang lebih unggul diantara *Throughput, Delay, PacketLoss,* dan *Jitter* merupakan:

1. *Throughput* : Pengujian *throughput* yang lebih unggul merupakan kontroler *open day light* dengan kecepatan pengiriman 68 bps.
2. *PacketLoss* : *Packet loss* merupakan jumlah paket yang terkirim ke tujun dan kontroler yang paling sedikit terjadi *packet loss* merupakan kontroler *floodlight* merupakan 96 % paket yang terkirim.
3. *Delay* : Rentanggan waktu pengiriman yang terendah merupakan kontroler POX dengan kecepatan menempuh jarak pengiriman ke tujuan iyalah 214 ms.
4. *Jitter* : Pengolahan data hingga penghimpunan ulang paket-peket diakhir perjalanan pengiriman yang lebih cepat pengiriman merupakan kontroler *floodlight* dengan rentang antrian pengiriman paket -0,00116 ms.

BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan hasil analisa pengujian kontorler *floodlight, opendaylight,* POX, dan RYU dapat diambil kesimpulan sebagai berikut:

1. Pada penelitian ini, analisis performansi *software defined network* kontroler *floodlight,* POX, RYU dan *open day light* telah berhasil dilakukan dan menghasilkan performa yang berbeda. Dalam hal ini kontroler *floodlight* memiliki performa yang baik dalam parameter *packet loss,* dan *jitter* sedangkan kontroler *open day light* memiliki performa yang baik dalam parameter *throughput* dan kontroler POX di parameter *delay.*

2. Pada pengujian performa pada *throughtput,* kontroler *open day light* yang memiliki nilai kecepatan rata-rata transfer yaitu 68 bps dan pada kontroler POX memiliki nilai performa *delay* lebih kecil dengan nilai rata-rata keseluruhan 214 ms. Sedangkan kontroler *floodlight* pada performa *jitter* dengan rentang antrian pengiriman paket -0,00116 ms dan *packet loss* dengan persenan paket data yang terkirim merupakan 96% merupakan paket data yang sampai ketujuan.

3. Setiap kontroler memiliki performa yang berbeda-beda sehingga dapat diterapkan dengan semestinya sehingga berkerja lebih maksimal.

Berdasarkan hasil diatas bahwa untuk melakukan pengiriman data sebaiknya untuk di *throughtput* dan *jitter* menggunakan kontroler *floodlight* dan di *delay* dan *packet loss* menggunakan kontroler *open day light* dalam penelitian ini yang tepat untuk diimplentasikan dalam penelitian ini.

5.2. Saran

Saran yang dapat diberikan dari hasil analisa perbandingan performansi *software defined network* (sdn) kontorler *floodlight*, pox, ryu, dan *open day light* adalah sebagai berikut :

1. Perlu adanya penelitian lanjutan tentang *loadblancing* antar controler agar permorma dari kontroler agar lebih setabil
2. Perlu dilakukan kajian mendalam untuk meninggkatkan performa dari kontroler-kontroler yang lainnya agar mendapakan performa yang lebih baik.
3. Jika kontroler ini ingin di implementasikan didalam server sebaiknya dilakukan secara detail dan dilengkapi dengan ovswitch agar kontroler berjalan dengan baik

DAFTAR PUSTAKA

Abdillah, D., Sibaroni, Y., & Ummah, I. (2016). *Perancangan dan analisis jaringan virtual berbasis software- define networking (sdn) design and analysis of virtual network based on software- define networking (sdn)*. *3*(1), 1247–1252.

Edgar, R., Hanuranto, A. T., & Mentari, O. (2019). *Perancangan dan analisis sistem pada kontroler pox , ryu , dan opendaylight pada software defined network design and analysis system on kontorler pox , ryu , and opendaylight on software defined network*. *6*(2), 4433–4441.

Prayoga, D., Ijtihadie, R. M., & Husni, M. (2017). *Implementasi POX pada Perangkat Lunak Software-Defined Networking Kontorler untuk Data Center Berbasis Container*. *6*(2), 352–355.

Putra, M. W., Pramukantoro, E. S., & Yahya, W. (2018). *Analisis Perbandingan Performansi Kontroler Floodlight , Maestro , RYU , POX dan ONOS dalam Arsitektur Software Defined Network (SDN)*. *2*(10), 3779–3787.

Kartadie, R., & Panggayuh, V. (2019). *Floodlight vs onos dalam unjuk kerja*. *04*, 111–121.

Krisandi, A., Ir. Agus Virgono., M. T., & Drs. Ir. R Rumani M., Bc.TT., M. S. (2018). *ANALISIS EFEK PENGGUNAAN KONTROLER FLOODLIGHT DAN OPENDAYLIGHT PADA PERFORMANSI JARINGAN SDN*. *5*(3), 6011–6024.

Susianto, Didi, and Anisa Rachmawati. 2018. “Implementasi Dan Analisis Jaringan Menggunakan Wireshark, Cain and Abels, Network Minner (Studi Kasus: AMIK Dian Cipta Cendikia).” *Jurnal Cendikia* XVI: 120–25.

UNIVERSITAS ISLAM RIAU
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA
Jln. Kaharuddin Nasution no.113, Perhentian Marpoyan, Pekanbaru-Riau 28284
Telp: 0761-674674, fax: 0761-674834

Lembaran Persetujuan Tim Penguji Ujian Seminar Hasil

Nama	: Yoga Saputra
NPM	: 163510608
Jurusan	: Teknik
Program Studi	: Teknik Informatika
Jenjang Pendidikan	: Strata Satu (S1)
Judul Skripsi	: Analisis perbandingan performansi *software defined network* (SDN) controller floodlight, pox, ryu, dan odl pada topologi jaringan universitas islam riau

Naskah skripsi ini secara keseluruhan telah diperiksa dan memenuhi ketentuan metode penelitian ilmiah, oleh karena itu Tim Penguji dan Pembimbing dapat menyetujui dan menerima untuk mengikuti **Ujian Komprehensif**.

Pekanbaru, 11 November 2021

Disetujui Oleh :

Pembimbing

Dr. Apri Siswanto, S.Kom., M.Kom

Disahkan Oleh

Penguji I	Penguji II
Dr. Evizal Abdul Kadir, ST., M.Eng	**Yudhi Arta, ST, M. Kom**

UNIVERSITAS ISLAM RIAU
FAKULTAS TEKNIK
Alamat: Jalan Kaharuddin Nasution No.113, Marpoyan, Pekanbaru, Riau, Indonesia - 28284
Telp. +62 761 674674 Email: fakultas_teknik@uir.ac.id Website: www.eng.uir.ac.id

SURAT KETERANGAN BEBAS PLAGIAT
Nomor: 458/A-UIR/5-T/2021

Operator Turnitin Fakultas Teknik Universitas Islam Riau menerangkan bahwa Mahasiswa/i dengan identitas berikut:

Nama	:	**YOGA SAPUTRA**
NPM	:	163510608
Program Studi	:	Teknik Informatika
Jenjang Pendidikan	:	Strata Satu (S1)
Judul Skripsi TA	:	ANALISIS PERFORMANSI SOFTWARE DEFINED NETWORK (SDN) CONTROLLER FLOODLIGHT, POX, RYU, DAN ODL PADA TOPOLOGI JARINGAN UNIVERSITAS ISLAM RIAU

Dinyatakan **Bebas Plagiat,** berdasarkan hasil pengecekan pada Turnitin menunjukkan angka **Similarity Index < 30%** sesuai dengan peraturan Universitas Islam Riau yang berlaku.

Demikian surat keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Mengetahui,
November 2021 M

Pekanbaru, 30

25 Robī'ust Tsānī 1443 H

Kaprodi. Teknik Informatika

Dr. Apri Siswanto, M.Kom.

Operator Turnitin F. Teknik

Ahmad Pandi, S.Kom.

SURAT KEPUTUSAN DEKAN FAKULTAS TEKNIK UNIVERSITAS ISLAM RIAU
NOMOR : 0904/KPTS/FT-UIR/2020
TENTANG PENGANGKATAN TIM PEMBIMBING PENELITIAN DAN PENYUSUNAN SKRIPSI

DEKAN FAKULTAS TEKNIK

Membaca : Surat Ketua Program Studi Teknik Informatika Nomor : 084/TA/TI/FT/2020 tentang persetujuan dan usulan pengangkatan Tim Pembimbing penelitian dan penyusunan Skripsi.

Menimbang : 1. Bahwa untuk menyelesaikan perkuliahan bagi mahsasiswa Fakultas Teknik perlu membuat Skripsi.

2. Untuk itu perlu ditunjuk Tim Pembimbing penelitian dan penyusunan Skripsi yang diangkat dengan Surat Keputusan Dekan.

Mengingat : 1. Undang - Undang Nomor 12 Tahun 2012 Tentang Pendidikan Tinggi
2. Peraturan Presiden Republik Indonesia Nomor 8 Tahun 2012 Tentang Kerangka Kualifikasi Nasional Indonesia
3. Peraturan Pemerintah Republik Indonesia Nomor 37 Tahun 2009 Tentang Dosen
4. Peraturan Pemerintah Republik Indonesia Nomor 66 Tahun 2010 Tentang Pengelolaan dan Penyelenggaraan Pendidikan
5. Peraturan Menteri Pendidikan Nasional Nomor 63 Tahun 2009 Tentang Sistem Penjaminan Mutu Pendidikan
6. Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 49 Tahun 2014 Tentang Standar Nasional Pendidikan Tinggi
7. Statuta Universitas Islam Riau Tahun 2018
8. Peraturan Universitas Islam Riau Nomor 001 Tahun 2018 Tentang Ketentuan Akademik Bidang Pendidikan Universitas Islam Riau

MEMUTUSKAN

Menetapkan : 1. Mengangkat saudara-saudara yang namanya tersebut dibawah ini sebagai Tim Pembimbing Penelitian dan penyusunan Skripsi mahasiswa Fakultas Teknik Program Studi Teknik Informatika.

No	Nama	Pangkat	Jabatan
1.	Apri Siswanto,S.Kom.,M.Kom	Lektor	Pembimbing

2. Mahasiswa yang akan dibimbing :

Nama : Yoga Saputra
NPM : 163510608
Program Studi : Informatika
Jenjang Pendidikan : Strata Satu (S1)
Judul Skripsi Baru : Analisis Perbandingan Performasi *Software Defined Network* (SDN) Controller Floodligh, POX, RYU, dan ODL pada Topologi Jaringan Universitas Islam Riau

3. Keputusan ini mulai berlaku pada tanggal ditetapkannya dengan ketentuan bila terdapat kekeliruan dikemudian hari segera ditinjau kembali.

Ditetapkan di : Pekanbaru
Pada Tanggal : 07 Muharam 1442 H
26 Agustus 2020 M

Dekan,

Dr. Eng. Muslim, S.T., M.T.
NPK : 09 11 02 374

Tembusan disampaikan :
1. Yth. Bapak Rektor UIR di Pekanbaru.
2. Yth. Sdr. Ketua Program Studi Teknik Informatika FT-UIR
3. Yang Bersangkutan
4. Arsip

LANGUAGE CERTIFICATION AND TRAINING INSTITUTE
L.A. LANGUAGE ACADEMY
Operational License (SK DIKPORA) : No. 016/LKP/P/VI/2020 NPSN : K9990439
Address : Sukun Street No. 379 Banguntapan, Bantul, Special District Of Yogyakarta 55198 Indonesia
Email : la_languageacademy@kampungbahasajogja.net Website : www.kampungbahasajogja.net

TOEFL CERTIFICATE

This is to certify that

Full Name : Yoga Saputra
Date of Birth : November 9, 1998
Certification Number : 2021718431
Test Date : December 4, 2021

Achieved the following scores on the TOEFL Prediction test

SECTIONS	SCORES
Section 1 : Listening Comprehension	39
Section 2 : Structure and Written Expression	68
Section 3 : Reading Comprehension	60
TOTAL	556

This certificate is valid for two years since the test date

بسم الله الرحمن الرحيم

UNIVERSITAS ISLAM RIAU
LEMBAGA DAKWAH ISLAM KAMPUS (LDIK)

SERTIFIKAT

Nomor Registrasi : 6482/LDIK-UIR/2020

Berdasarkan
Surat Keputusan Rektor Universitas Islam Riau Nomor : 525/UIR/KPTS/2018
tentang Kewajiban Mahasiswa Muslim Universitas Islam Riau Bisa Membaca Al-Qur'an,
Lembaga Dakwah Islam Kampus (LDIK) Universitas Islam Riau menyatakan bahwa:

YOGA SAPUTRA
Nomor Pokok Mahasiswa : 163510608

Lahir di Sei Guntung Tanggal Delapan November Tahun Seribu Sembilan Ratus Sembilan Puluh Delapan
Mahasiswa Prodi Teknik Informatika Fakultas Teknik Universitas Islam Riau
LULUS Tes Baca Al-Qur'an Dengan Predikat **Baik**

Diuji Pada : 07.01.20

Pekanbaru, 08 Januari 2020
Ketua,

Dr. H. Zulhelmy, S.E.,M. Si., Ak.C.A
NPK : 98 07 02 272

Corporate Social Responsibility

Cisco Networking Academy

Certificate of Course Completion

CCNA Routing and Switching: Introduction to Networks

The student has successfully achieved student level credential for completing CCNA Routing and Switching: Introduction to Networks course administered by the undersigned instructor. The student was able to proficiently:

- Explain network technologies.
- Explain how devices access local and remote network resources.
- Describe router hardware.
- Explain how switching operates in a small to medium-sized business network.
- Design an IP addressing scheme to provide network connectivity for a small to medium- sized business network.
- Configure initial settings on a network device.
- Implement basic network connectivity between devices.
- Configure monitoring tools available for small to medium-sized business networks.

Yoga Saputra
Student

Islamic University of Riau
Academy Name

Indonesia
Location

May 29, 2018
Date

Apri Siswanto
Instructor

Instructor Signature