

**ANALISIS PERFORMANSI ONOS, POX, RYU CONTROLLER PADA  
PROTOKOL ROUTING OSPF DAN BGP DI SOFTWARE DEFINED  
NETWORK BERBASIS RASPBERRY-PI**

*Performance Analysis Of ONOS, POX, RYU Cotroller In OSPF And BGP Routing  
Protocols In Software Defined Network Based On Raspberry-Pi*

**PROPOSAL PROYEK AKHIR**

**Diajukan sebagai syarat untuk mengambil Mata Kuliah Proyek Akhir**

**oleh :**

**USU GUNAWAN**

**6705180021**



**D3 TEKNOLOGI TELEKOMUNIKASI  
FAKULTAS ILMU TERAPAN  
UNIVERSITAS TELKOM  
2021**

## LEMBAR PENGESAHAN

Proposal Proyek Akhir dengan judul :

ANALISIS PERFORMANSI ONOS, POX, RYU *CONTROLLER* PADA PROTOKOL  
*ROUTING* OSPF DAN BGP DI *SOFTWARE DEFINED NETWORK* BERBASIS  
RASPBERRY-PI

*Performance Analysis Of ONOS, POX, RYU Cotroller In OSPF And BGP Routing  
Protocols In Software Defined Network Based On Raspberry-Pi*

oleh :

USU GUNAWAN

6705180021

Telah diperiksa dan disetujui untuk diajukan sebagai syarat mengambil  
Mata Kuliah Proyek Akhir  
pada Program Studi D3 Teknologi telekomunikasi Universitas Telkom

Bandung, Januari 2020

Menyetujui,

Pembimbing I

22-Jan-21

Untuk Proposal PA  
Usu Gunawan

Rohmat Tulloh, S.T., M.T.

NIP. 06830002

Pembimbing II

Dr. Indrarini Dyah Irawati, S.T., M.T.

NIP. 07780053

## ABSTRAK

Perkembangan teknologi jaringan semakin berkembang dan terus meningkat, untuk memenuhi kebutuhan dan menciptakan jaringan yang handal, SDN (*Software Defined Network*) merupakan salah satunya. SDN merupakan suatu arsitektur jaringan yang kendali jaringan dipusatkan pada kontroller yang diprogram mampu untuk mengatur proses jalannya data ke pusat dan mengirim ke client. Oleh Karena itu kontroler pada SDN merupakan elemen penting dalam performansi jaringan. SDN memiliki banyak kontroler seperti, NOX, POX, ONOS, Beacon, Floodlight, MuL, Maestro, Ryu dan lain-lain. Setiap kontroler memiliki perbedaan dalam kelebihan dan kekurangan sehingga mempengaruhi performansi jaringan.

Pada penelitian proyek akhir ini teknologi SDN yang akan digunakan adalah ONOS, POX dan RYU sebagai *controller* yang akan dipasang pada perangkat Raspberry-Pi untuk meminimalisir penggunaan perangkat PC pada jaringan SDN. Analisis yang akan dilakukan adalah dengan membandingkan QoS (*Quality of Service*) jaringan yang dibangun menggunakan kontroler ONOS, POX dan RYU dengan *routing* protokol BGP. Selain itu akan diteliti juga QoS jaringan yang dibangun menggunakan kontroler POX dan RYU dengan *routing* protokol OSPF.

Dengan dilakukannya analisis perbandingan ini diharapkan dapat mengetahui cara kerja ONOS, POX dan RYU *Controller* pada SDN, Serta mengetahui performa kontroler mana yang paling baik ketika menggunakan routing OSPF dan BGP yang akan ditampilkan melalui hasil analisa berupa QoS.

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
ABSTRAK .....	ii
DAFTAR ISI .....	iii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan dan Manfaat .....	2
1.3 Rumusan Masalah .....	2
1.4 Batasan Masalah .....	3
1.5 Metodologi .....	3
BAB II DASAR TEORI .....	4
2.1 <i>Software Defined Network</i> .....	4
2.2 <i>Openflow</i> .....	5
2.3 <i>Open Network Operating System (ONOS)</i> .....	5
2.4 POX .....	6
2.5 RYU .....	7
2.6 Protokol Routing OSPF ( <i>Open Shortest Path First</i> ) .....	8
2.7 <i>Border Gateway Protocol (BGP)</i> .....	8
2.8 Raspberry-Pi .....	9
2.9 Quality of Service (QoS) .....	10
2.10 Parameter Quality of Services .....	10
2.11 Quagga .....	12
BAB III MODEL SISTEM .....	14
3.1 Blok Diagram Sistem .....	14
3.2 Tahapan Perancangan .....	15
3.3 Perancangan .....	16
BAB IV BENTUK KELUARAN YANG DIHARAPKAN .....	18
4.1 Keluaran yang Diharapkan .....	18
4.2 Jadwal Pelaksanaan .....	19
DAFTAR PUSTAKA .....	21

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Seiring berkembangnya zaman, perkembangan teknologi khususnya teknologi jaringan menyebabkan banyaknya penelitian dan percobaan pada platform *software defined network* dengan tujuan memperbaiki kondisi yang lebih baik dari pada jaringan yang digunakan sekarang yaitu jaringan konvensional. Jaringan *software defined network* merupakan teknologi baru yang memisahkan fungsi data plane dari control plane yang memungkinkan dapat memprogram perangkat seperti switch dan router sesuai dengan yang diinginkan secara terpusat dan membuat arsitektur jaringan lebih fleksibel hemat biaya dan lebih efisien dalam mengkonfigurasi suatu jaringan. SDN mampu membuat jaringan baik berskala kecil maupun berskala besar mampu terkontrol menggunakan 1 controller yang terpusat. Beberapa kontroler yang ada pada SDN diantaranya adalah POX, RYU, OpenDaylight, dan ONOS. Untuk mendukung jaringan dengan skala besar, *Network Operating System* harus memenuhi persyaratan yang menuntut *Scalability*, *Performance*, dan *Availability*.

Pada penelitian [4] , telah dilakukan analisis perbandingan beberapa *controller* SDN seperti POX, RYU, Maestro, Floodlight, dan ONOS. Perbandingan beberapa *controller* tersebut dilakukan dengan memvariasikan jumlah *switch* untuk melihat controller mana yang memiliki kemampuan terbaik dalam menangani aliran data dalam jumlah besar. Pada penelitian [11], telah dilakukan perancangan dan implementasi *routing* EBGp menggunakan ONOS *controller* dan dilakukan pengukuran QoS menggunakan standard ITU-T G.1010. Pada penelitian [13], telah dilakukan analisis perbandingan nilai QoS jaringan yang dibangun menggunakan kontroler RYU dan POX menggunakan topologi *full mesh*. Pada penelitian [5] dilakukan simulasi *routing* OSPF dengan menggunakan kontroler RYU dengan simulasi 8 buah *switch openflow*.

Dengan perbandingan penelitian sebelumnya pada penelitian proyek akhir ini teknologi SDN yang digunakan adalah ONOS, POX dan RYU yang akan dipasang pada perangkat Raspberry-Pi untuk meminimalisir penggunaan perangkat

PC pada jaringan SDN. Analisis yang akan dilakukan adalah dengan membandingkan QoS (*Quality of Service*) jaringan yang dibangun menggunakan kontroler ONOS, POX dan RYU dengan *routing* protokol BGP. Selain itu akan diteliti juga QoS jaringan yang dibangun menggunakan kontroler POX dan RYU dengan *routing* protokol OSPF.

Dengan dilakukannya analisis perbandingan ini diharapkan dapat mengetahui cara kerja ONOS *Controller*, POX *Controller* dan RYU *Controller* pada SDN, Serta mengetahui performa kontroler mana yang paling baik ketika menggunakan routing OSPF dan BGP yang akan ditampilkan melalui hasil analisa berupa QoS.

## **1.2 Tujuan dan Manfaat**

Adapun tujuan dari Proyek tingkat ini, sebagai berikut:

1. Dapat mengimplementasikan jaringan *Software Defined Network* (SDN) dengan menggunakan ONOS , POX dan RYU *Controller* berbasis Raspberry-Pi .
2. Menganalisis penggunaan kontroler Ryu dan POX yang menggunakan *routing* protokol OSPF dengan membandingkan nilai *Quality of Service* yang dihasilkan dengan nilai standarisasi ITU-T
3. Menganalisis penggunaan kontroler ONOS, POX dan RYU yang menggunakan *routing* protokol BGP dengan membandingkan nilai *Quality of Service* yang dihasilkan dengan nilai standarisasi ITU-T

## **1.3 Rumusan Masalah**

Adapun rumusan masalah dari Proyek tingkat ini, sebagai berikut:

1. Bagaimana cara mengimplementasikan jaringan *Software Defined Network* (SDN) dengan menggunakan ONOS , POX dan Ryu *Controller* berbasis Raspberry-Pi ?
2. Bagaimana penggunaan kontroler Ryu dan POX yang menggunakan *routing* protokol OSPF dengan membandingkan nilai *Quality of Service* yang dihasilkan dengan nilai standarisasi ITU-T ?

3. Bagaimana penggunaan kontroler ONOS, POX dan RYU yang menggunakan *routing* protokol BGP dengan membandingkan nilai *Quality of Service* yang dihasilkan dengan nilai standarisasi ITU-T ?

#### 1.4 Batasan Masalah

Dalam Proyek tingkat ini, dilakukan pembatasan masalah sebagai berikut:

1. ONOS, POX dan RYU *controller* sebagai *control plane*
2. Menggunakan Raspberry-Pi sebagai *controller* dengan sistem operasi Raspbian.
3. Parameter dalam analisis QoS dalam menggunakan standarisasi ITU-T.

#### 1.5 Metodologi

Metodologi pada penelitian ini, sebagai berikut:

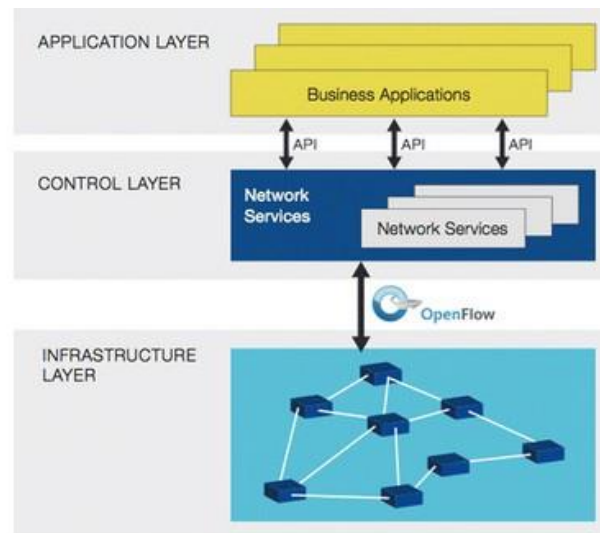
1. Studi Literatur  
Pencarian informasi, pendalaman materi serta pemahaman konsep dan pengumpulan data dan informasi yang berkaitan dengan kontroler SDN, algoritma routing, parameter kinerja melalui berbagai sumber seperti buku, jurnal, dan internet.
2. Perancangan Sistem  
Melakukan perancangan dan permodelan topologi sistem, serta cara kerja sistem sesuai dengan model sistem yang diinginkan.
3. Implementasi Perancangan  
Mengimplementasi perancangan topologi sistem yang telah dibuat sesuai dengan perancangan jaringan yang diinginkan.
4. Pengujian  
Menentukan parameter-parameter yang akan diuji lalu melakukan pengujian terhadap sistem yang telah dirancang berdasarkan parameter-parameter tersebut.
5. Analisa sistem  
Mengamati dan menguji hasil dari sistem yang dirancang serta menyimpulkan dan menganalisa dengan menentukan dan membandingkan apakah sesuai dan memenuhi kriteria yang diharapkan
6. Kesimpulan  
Menyimpulkan hasil dari seluruh tahapan yang telah dilakukan

## BAB II

### DASAR TEORI

#### 2.1 *Software Defined Network*

*Software Defined Network* (SDN) adalah konsep baru dalam mengontrol, mengimplementasi serta mengelola suatu jaringan yang mendukung kebutuhan dan inovasi di bidang telekomunikasi yang semakin lama semakin berkembang dan kompleks. Konsep dasar dari SDN yaitu dengan memisahkan antara *control plane* dan *data plane*. Gambar arsitektur SDN bisa dilihat pada Gambar 2.1



Gambar 2.1 Arsitektur *Software Defined Network*

Arsitektur SDN terdiri dari 3 layer :

- Data Plane* : terdiri dari elemen jaringan yang dapat mengatur SDN *Datapath* sesuai dengan instruksi yang diberikan melalui *Control-Data Plane Interface* (CDPI).
- Control Plane* : entitas kontrol (SDN *Controller*) mentranslasikan kebutuhan aplikasi dengan infrastruktur dengan memberikan instruksi yang sesuai untuk SDN *Datapath* serta memberikan informasi yang relevan dan dibutuhkan oleh SDN *Application*.
- Application* : berada pada lapis teratas, berkomunikasi dengan sistem via *North Bound Interface* (NBI).



*Software Defined Networking* (SDN) memperkenalkan strategi dan protokol kontrol jaringan inovatif baru berdasarkan pada logika kontrol pusat. arsitektur yang muncul yang dinamis, mudah dikelola, hemat biaya, dan mudah beradaptasi, menjadikannya ideal untuk bandwidth tinggi, sifat dinamis dari aplikasi saat ini. Arsitektur ini memisahkan fungsi kontrol jaringan dan penerusan yang memungkinkan kontrol jaringan menjadi secara langsung dapat diprogram dan infrastruktur yang mendasarinya akan diabstraksi untuk aplikasi dan layanan jaringan.

## 2.2 *Openflow*

*Openflow* adalah suatu protokol dalam Jaringan SDN yang berfungsi sebagai penghubung antara *control layer* dan infrastruktur layer pada arsitektur jaringan SDN. Dengan menggunakan *Openflow* pada jaringan SDN, perangkat antara *controller* SDN dan perangkat *Data Plane* dapat terhubung. *Openflow* mendefinisikan infrastruktur *flow-based forwarding* dan *Application Programmatic Interface* (API) standart yang memungkinkan controller untuk mengarahkan fungsi dari *Switch* melalui saluran yang aman (*secure channel*).

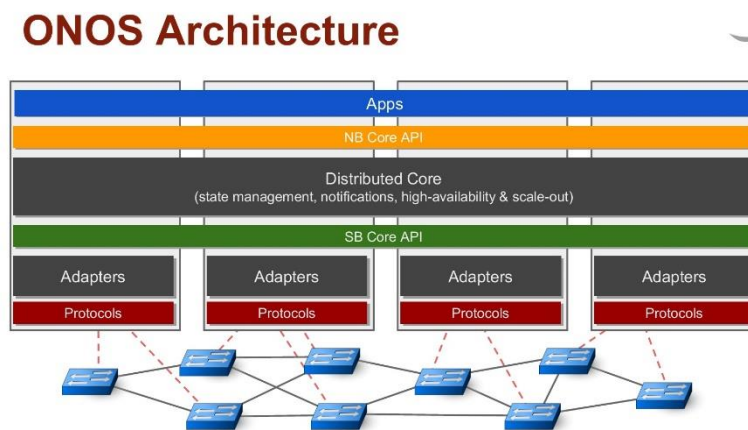


Gambar 2.2 *Openflow*

## 2.3 *Open Network Operating System (ONOS)*

ONOS adalah sistem operasi jaringan SDN open source terkemuka untuk membangun solusi SDN generasi selanjutnya yang ditargetkan secara khusus di *Service Provider* dan *Mission Critical Network*. ONOS dibangun untuk memberikan ketersediaan tinggi (*High Availability*), *scale-out*, dan performansi kinerja jaringan yang dibutuhkan. *Controller* ini berbasis java. Dengan pengontrol *cloud* ONOS, pengguna dapat dengan mudah membuat aplikasi jaringan baru tanpa perlu mengubah sistem *data plane*. ONOS mendukung konfigurasi dan kontrol

jaringan secara *real-time*, dan menghilangkan kebutuhan untuk menjalankan *routing* serta mengalihkan protokol kontrol di dalam jaringan. ONOS adalah proyek *open source* yang didistribusikan di bawah lisensi Apache 2.0 menggunakan bahasa pemrograman berbasis Java dan menggunakan Apache Karav untuk menghidupkan setiap fitur yang ada pada ONOS.



Gambar 2.3 Arsitektur ONOS

## 2.4 POX

POX adalah *platform* pengembangan sumber terbuka untuk aplikasi SDN dengan bahasa pemrograman berbasis Python, seperti kontroler *Openflow* SDN. POX, memungkinkan pengembang dan pembuatan *prototype* 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 *controller* 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. Sementara komponen ini mempelajari alamat L2, arus yang dipasangnya persis sama di banyak bidang sebanyak mungkin. Sebagai contoh, koneksi TCP yang berbeda akan menghasilkan arus yang berbeda yang diinstal.



Gambar 2.4 Logo POX

## 2.5 RYU

Ryu merupakan salah satu *controller* dalam *software defined network* yang dirancang untuk meningkatkan kemampuan dalam jaringan yang bermanfaat untuk mempermudah dan mengatur. Secara umum *controller* merupakan fungsi otak dari *software defined network*. RYU merupakan *controller open source* yang dikembangkan oleh NTT. Dalam RYU *application program interface* (API) sudah didefinisikan dengan sangat baik yang berarti dapat melakukan pengembangan dengan mudah untuk membuat suatu network management yang baru. RYU merupakan *controller* yang menggunakan bahasa pemrograman python yang mudah dalam pemakaiannya serta memiliki dokumentasi yang banyak sehingga lebih mudah menemukan solusi jika terdapat permasalahan. *Controller* RYU ini mendukung beberapa *protocol* dalam *software defined network* diantaranya *Openflow*, *Netconf*, *OF-config* serta lainnya.



Gambar 2.5 Logo RYU

## 2.6 Protokol Routing OSPF (*Open Shortest Path First*)

Protokol OSPF adalah salah satu protokol *ip routing* dan merupakan *Protocol Gateway Interior* (IGP) untuk internet yang digunakan untuk mendistribusikan informasi *routing* keseluruhan jaringan yang saling terhubung. OSPF mengumpulkan informasi *link state* dari *router* yang ada dan membangun sebuah grafik topologi dari jaringan. Untuk menentukan jalur terpendek, OSPF membutuhkan pemberian bobot setiap link di jaringan. Dan memang kebanyakan fitur ini digunakan untuk *management* dalam skala jaringan yang sangat besar. Oleh karena itu untuk mempermudah penambahan informasi *routing* dan meminimalisir kesalahan distribusi informasi *routing*, maka OSPF bisa menjadi sebuah solusi.

OSPF menggunakan protokol *routing interior* dengan algoritma *link state*, OSPF memiliki karakteristik sebagai berikut :

- a. Melakukan *update routing* apabila terjadi perubahan topologi jaringan.
- b. Menggunakan Algoritma SPF untuk menghitung *cost* terendah.
- c. Protokol *routing link state*

## 2.7 Border Gateway Protocol (BGP)

*Border Gateway Protocol* adalah protokol *routing* standar yang bertujuan untuk memilih jalur – jalur interdomain. Fungsi utama dari BGP untuk mempertukarkan *network reachability information* antar BGP *router*. BGP dirancang untuk menyediakan *link bebas loop* antar penyedia layanan. BGP menggunakan protokol TCP (*port 179*) sebagai protokol *transport* karena TCP adalah protokol *connection-oriented*. Cisco adalah salah satu penyedia perangkat keras dan lunak yang mendukung BGP versi 4 dan ini adalah versi yang telah digunakan oleh penyedia layanan internet untuk membangun internet. BGP mempunyai 2 standarisasi dari *Internet Engineering Task Force* (IETF) yaitu :

- A. RFC 1771 : memiliki sejumlah fitur baru yang memungkinkan protokol untuk penggunaan skala internet.
- B. RFC 2858 : merupakan *multiprotocol* BGP yang memungkinkan BGP membawa informasi *routing* untuk *multicast* IP dan beberapa alamat

protokol *layer 3*, termasuk IPv4, IPv6 dan *Connectionless-mode Network Service* (CLNS)

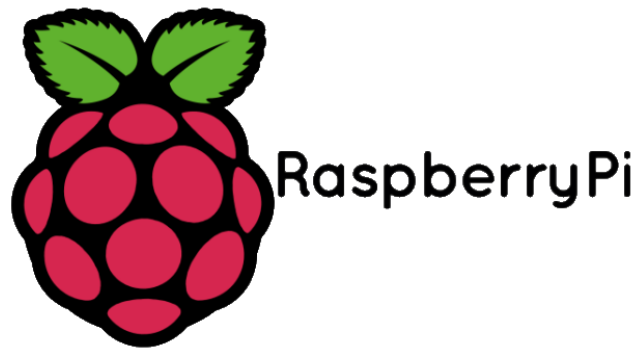
BGP biasanya digunakan untuk menghubungkan jaringan lokal ke jaringan eksternal (internet) agar dapat mengakses jaringan yang tersebar di internet. Saat tersambung ke penyedia jasa layanan luar, sesi *external BGP* (eBGP) dibuat. Meskipun BGP disebut sebagai *exterior gateway protocol* (EGP), banyak jaringan dalam suatu penyedia layanan menjadi kompleks sehingga BGP dapat digunakan untuk menyederhanakan jaringan internal yang bisa digunakan untuk bertukar informasi sesama penyedia layanan. Rekan BGP dalam penyedia layanan yang sama dapat berkomunikasi dengan menggunakan protokol *internal BGP* (iBGP)[13]. Dalam protokol BGP terdapat sebuah *Autonomous Systems* (AS), AS merupakan sebuah *group router* yang saling berbagi aturan kebijakan *routing* yang sama dan beroperasi dalam satu *domain administrative*, satu penyedia layanan biasanya hanya mempunyai satu AS . Nomor AS yang tersedia antara 1 sampai 65.535, AS yang dapat digunakan jaringan publik adalah 1 sampai 64.512 untuk nomor 64.512 sampai 65.535 digunakan untuk keperluan pribadi (seperti *IP Private*)[14]. Jika AS terhubung ke jaringan publik (internet) menggunakan protokol eBGP, maka AS harus diberi nomor AS unik yang dikelola oleh *Internet Assigned Number Authority* (IANA). IANA bertanggung jawab untuk mengalokasikan nomor AS melalui lima *Regional Internet Registries* (RIRs). RIRs adalah perusahaan *non-profit* yang didirikan untuk tujuan mengelola administrasi dan pendaftaran alokasi alamat IP dan nomor AS di lokasi geografis utama.

## 2.8 Raspberry-Pi

Raspberry-Pi adalah sebuah *mini computer*. Raspberry-Pi telah memiliki berbagai fungsi layaknya sebuah komputer lengkap, menggunakan *System-on-a-chip* (SoC) ARM yang dikemas dan diintegrasikan diatas PCB. Perangkat ini menggunakan kartu SD untuk *booting*.

Pada Raspberry-Pi terdapat dua jenis model yaitu model A dan model B. Secara umum Raspberry-Pi Model B, 512MB RAM. Perbedaan model A dan B terletak pada memori yang digunakan, Model A menggunakan memory 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapi dengan *ethernet port* (kartu jaringan) yang tidak terdapat di model A.. Penyimpanan data pada

Raspberry-Pi tidak menggunakan hardisk atau *solid-state drive*, melainkan mengandalkan kartu SD (SD memory card) untuk *booting* dan penyimpanan jangka panjang. *Hardware* Raspberry-Pi tidak memiliki *real-time clock*, sehingga OS harus memanfaatkan timer jaringan *5 server* sebagai pengganti. Raspberry-Pi bisa dimodifikasi sesuai kebutuhan penggunaanya.



Gambar 2.6 Logo Raspberry Pi

## 2.9 Quality of Service (QoS)

*Quality of Services* adalah suatu cara untuk melakukan pengukuran kualitas suatu jaringan seperti *delay*, *jitter*, *packet loss*, *throughput* dan *bandwidth*. Tujuan dari pengukuran QoS adalah supaya mengetahui nilai yang terdapat pada parameter QoS (*delay*, *jitter*, *packet loss*, dan *throughput*) sesuai dengan nilai standart yang dikeluarkan oleh suatau lembaga seperti ITU-T.

Parameter QoS tercipta supaya *end user (client)* mendapatkan performasi yang lebih maksimal dan membantu *client* menjadi lebih produktif. QoS menunjuk pada kemampuan jaringan untuk dapat menyediakan suatu layanan yang lebih baik pada jaringan trafik tertentu dengan teknologi yang berbeda-beda. QoS adalah suatu parameter yang sangat diperlukan oleh suatu jaringan supaya suatu *client* dalam jaringan mendapatkan kualitas dan layanan yang lebih baik.

## 2.10 Parameter Quality of Services

### a. *Delay*

Delay adalah waktu tunda kedatangan paket yang disebabkan oleh proses transmisi data dari pengirim ke penerima. Standar *delay* menurut TIPHON ditunjukkan pada Tabel 2.1 berikut:

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

Tabel 2.1 Standar *Delay* TIPHON

b. *Packet loss*

*Packet loss* merupakan parameter QoS yang menunjukkan jumlah total paket yang hilang yang dikirim dari pengirim ke penerima. Paket yang hilang ini dapat disebabkan oleh adanya collision dan congestion pada jaringan. *Packet Loss* dihitung berdasarkan persentase paket yang berhasil dikirim.

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

Tabel 2.2 Standar *Packet Loss* TIPHON

c. *Throughput*

*Throughput* adalah kecepatan sebenarnya jaringan dalam melakukan pengiriman data. *Throughput* adalah kecepatan transfer data efektif yang diukur dengan satuan *bit per second* (bps).

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 2.3 Standar *Throughput* TIPHON

d. *Jitter*

*Jitter* adalah variasi *delay*. *Jitter* juga didefinisikan sebagai gangguan pada komunikasi digital maupun analog yang disebabkan oleh perubahan sinyal karena referensi posisi waktu. Adanya *jitter* ini dapat mengakibatkan hilangnya data, terutama pada pengiriman data dengan kecepatan tinggi. Standar *jitter* menurut TIPHON ditunjukkan pada Tabel 2.4 berikut :

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

Tabel 2.4 Standar *Jitter* TIPHON

## 2.11 Quagga

Quagga adalah suatu *routing engine* yang dapat menjalankan protokol *routing* konvensional seperti RIP, OSPF, BGP, dan lain-lain. Quagga yang tujuannya secara umum implementasi *routing* bersifat *open source* yang cocok untuk digunakan di SDN *enviromtment*. Penerapan quagga dapat digunakan untuk Unix *platform* khususnya FreeBSD, Linux, Solaris, dan NetBSD. Arsitektur Quagga terdiri dari *core daemon* , *zebra*, yang bertindak sebagai *layer* abstraksi untuk Unix kernel yang mendasari dan menyajikan Zserv API diatas unix atau TCP stream untuk



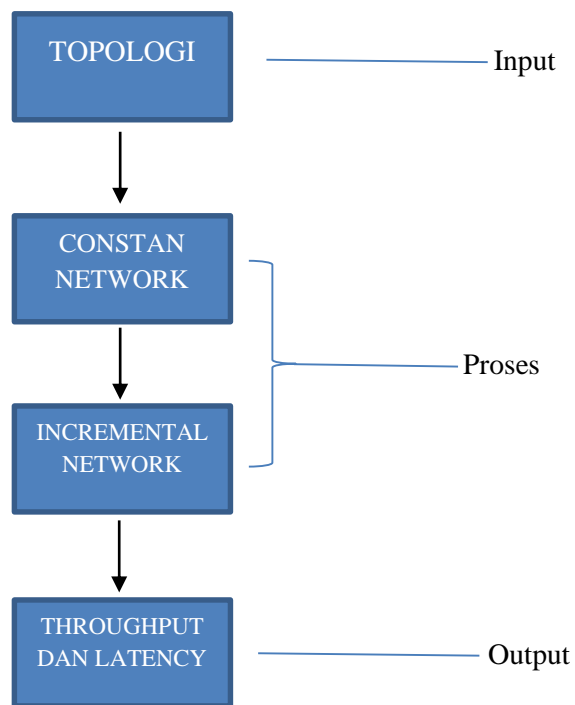
Quagga *client*. Pada Zserv *client* ini biasanya menerapkan protokol *routing* dan menyampaikan *update routing* ke *zebra daemon*.

## BAB III

### MODEL SISTEM

#### 3.1 Blok Diagram Sistem

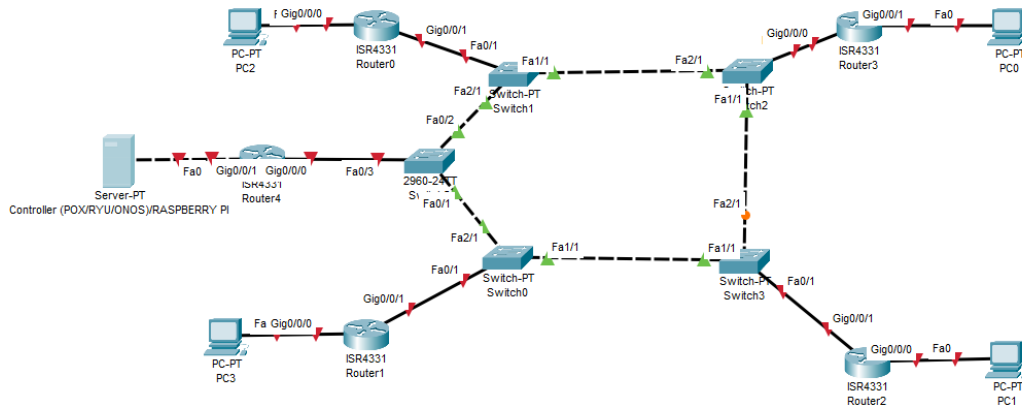
Pada bab ini akan dijelaskan mengenai perancangan dari analisis perbandingan performansi ONOS, POX, dan RYU *Controller* pada protokol routing OSPF dan BGP di SDN berbasis raspberry pi. Dimulai dengan merancang jaringan topologi yang sudah ditentukan dan diakhiri atau ouput berupa QoS pada masing-masing kontroler yaitu ONOS, POX dan RYU dengan *routing* BGP dan POX dan RYU untuk *routing* OSPF. Adapun model sistem yang telah dibuat dapat dilihat pada Gambar 3.1 dibawah ini.



Gambar 3.1 Blok Diagram Sistem

Merupakan gambaran umum perancangan proyek akhir ini. Kontroler akan digunakan dalam scenario jaringan menggunakan topologi yang sesuai. Dalam simulasi ini akan dilakukan pengukuran QoS pada masing-masing jaringan kontroler dan *routing protocol*.

Pada perancangan topologi komponen yang digunakan yaitu 4 PC *host*, 5 *switch*, dan 5 *router* serta raspberry-pi. berikut topologi jaringannya :



Gambar 3.2 Topologi Jaringan

### 3.2 Tahapan Perancangan

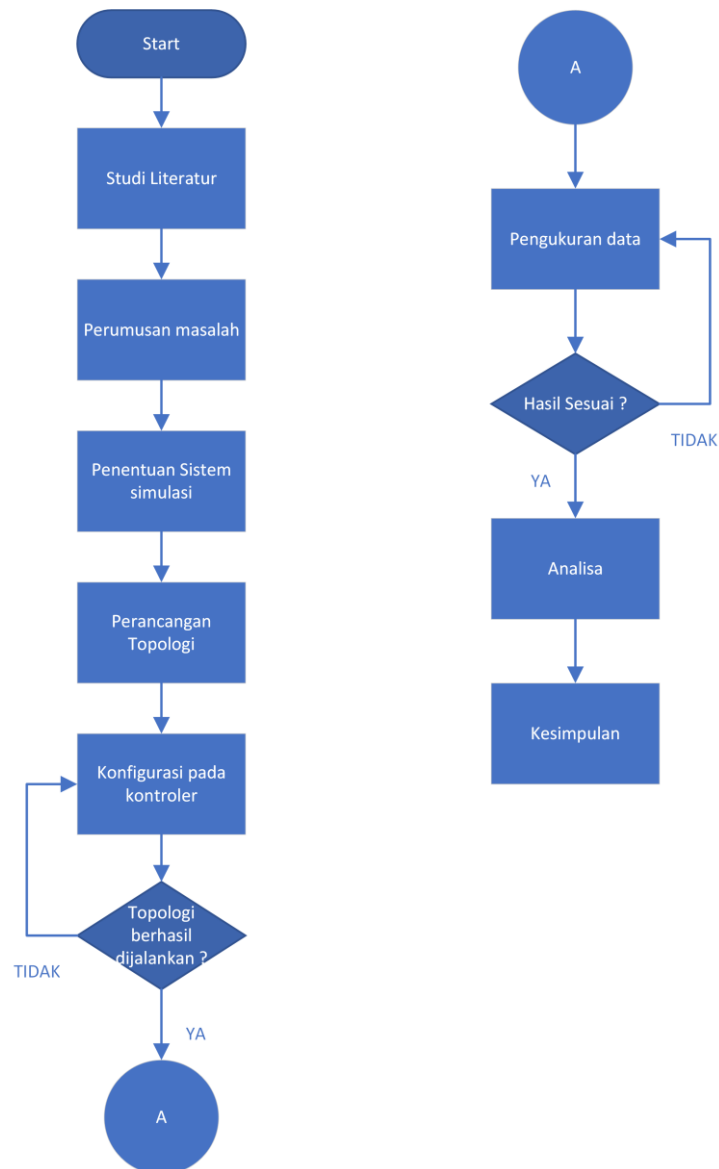
Proses perancangan dilakukan dengan metode eksperimental, tahapan pembuatannya adalah sebagai berikut :

1. Penentuan spesifikasi

Langkah awal dalam pembuatan dari proyek akhir ini adalah dengan menentukan rancangan topologi jaringan untuk mengintegrasikan semua komponen agar dapat bekerja dengan di atur oleh *Raspberry pi*.

2. Penyusunan Komponen

Semua komponen yaitu pc/laptop dan switch akan di hubungkan dengan *Raspberry pi* sebagai kontrolernya, untuk tahapan penyusunan komponennya dapat dibuat *flowchart* sebagai berikut:

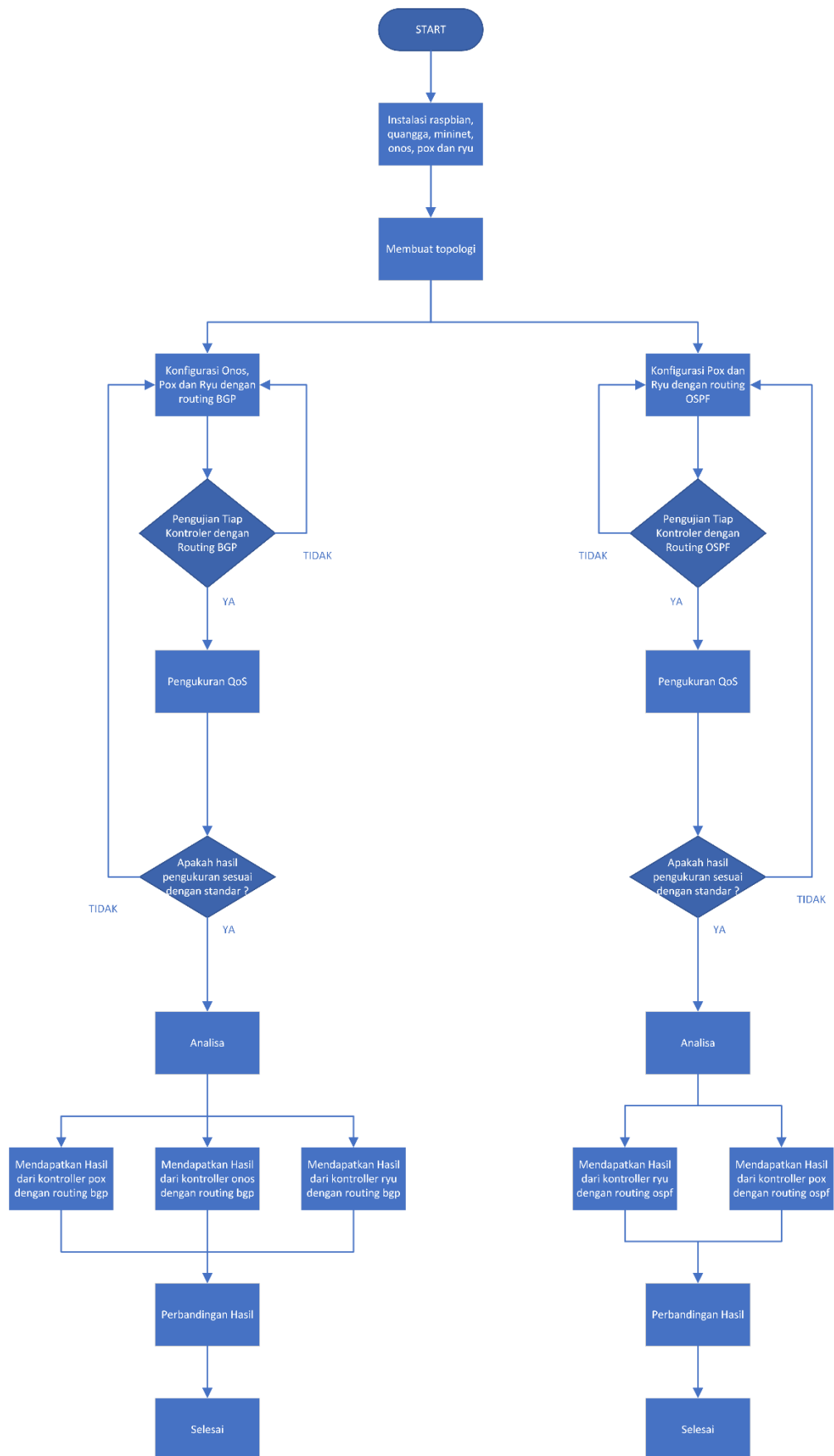


Gambar 3.2 Diagram Alir Rancangan

### 3.3 Perancangan

Pada Proyek tingkat ini akan dirancang sebuah jaringan yaitu berupa membuat topologi yang akan diimplementasikan. Komponen yang dibutuhkan yaitu 5 buah *switch openflow* sebagai *data plane*, kemudian 5 buah *router*, dan 4 PC/Laptop, pada implementasi ini penulis menggunakan quagga sebagai aplikasi *routing engine*. Dan berikut adalah rangkuman perangkat yang digunakan :

- a. Jumlah *swicth Openflow* : 5 buah
- b. Jumlah *Router* : 5 buah
- c. Jumlah *Host* : 4 buah
- d. *Tool* pengukuran QoS : Wireshark



Gambar 3.3 *Flowchart* umum sistem

## BAB IV

### BENTUK KELUARAN YANG DIHARAPKAN

#### 4.1 Keluaran yang Diharapkan

Perancangan pada Proyek tingkat akan dibuat dengan spesifikasi sebagai berikut :

- a) Spesifikasi laptop sebagai *host*

Tabel 4.1 Spesifikasi laptop

Processor	AMD Quad Core A10-9620p 2.4GHz up to 3.4Ghz
RAM	8 GB
VGA	AMD Radeon™ R5 Graphics + Radeon™ R6 M435DX Dual Graphics with 2GB DDR3 VRAM
Sistem Operasi	Windows 10

- b) Spesifikasi Raspberry-Pi Sebagai *Controller*

Tabel 4.2 Spesifikasi raspberry pi

CPU	4x ARM Cortex-A53, 1.2GHz
RAM	1GB LPDDR2 (900 MHz)
Network	10/100 Ethernet, 2.4GHz 802.11n Wireless
Sistem Operasi	Raspbian 4.19

- c) Spesifikasi Mikrotik Sebagai *Switch*

Tabel 4.2 Spesifikasi *switch*

LAN Port	4
RAM	32MB
Routeboard	RB-941-2nD (hAP-Lite)
Sistem Operasi	RouterOS

d) Spesifikasi *Router* TP-LINK WR1043N

Tabel 4.4 Spesifikasi *router*

Hardware Features	
Interface	4 10/100/1000Mbps LAN Ports
	1 10/100/1000Mbps WAN Port
Button	WPS/Reset Button
	Power Button
	Wi-Fi Button
Antenna	5dBi*3 Omni Directional
External Power Supply	9V / 0.85A
Wireless Standards	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Dimensions ( W x D x H )	8.9 x 5.8 x 1.3in. (225 x 148 x 33mm)
Wireless Features	
Frequency	2.4-2.4835GHz
Reception Sensitivity	270M: -68dBm@10% PER
	130M: -68dBm@10% PER
	108M: -68dBm@10% PER
	54M: -68dBm@10% PER
	11M: -85dBm@8% PER
	6M: -88dBm@10% PER
	1M: -90dBm@8% PER
Transmit Power	CE:
	<20dBm(2.4GHz)
	FCC:
	<30dBm
Wireless Function	Enable/Disable Wireless Radio, WDS Bridge, WMM,
	Wireless Statistics
Wireless Security	64/128/152-bit WEP / WPA / WPA2,WPA-PSK / WPA2-PSK

## 4.2 Jadwal Pelaksanaan

Adapun jadwal pengerjaan Proyek tingkat bisa dilihat pada tabel **Error! Reference source not found.** sebagai berikut :

Tabel 4.5 Jadwal Pelaksanaan

Judul Kegiatan	Waktu							
	Des	Jan	Feb	Mar	Apr	Mei	Jun	Jul
Studi Literatur								
Perancangan dan Simulasi								
Implementasi								
Pengujian								
Analisa								
Pembuatan Laporan								



## DAFTAR PUSTAKA

- [1] F. Adnantlya, S. N. Hertiana, L. Vidya, and Y. St, “Simulasi Dan Analisis Performansi Protokol Ruting Ebgp Pada Sdn ( Software Defined Network ) Simulation and Perfomance Analysis of Ebgp Routing Protocol on Sdn ( Software Defined Network ),” e-Proceeding Eng., vol. 2, no. 2, pp. 2346–2353, 2015.
- [2] A. Kodar, “Analisa Dan Uji Kinerja PC Router Yang Menjalankan Protokol Routing Border Gateway Protocol ( BGP ) Menggunakan Zebra / Quagga,” Semin. Nas. Pengaplikasian Telemat. SINAPTIKA 2010, vol. 1, pp. 162–167, 2010.
- [3] T. Ernawati and J. Endrawan, “Peningkatan Kinerja Jaringan Komputer dengan Border Gateway Protocol (BGP) dan Dynamic Routing,” J. Ilmu Komput. dan Inform., vol. 4, no. 1, p. 35, 2018
- [4] 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). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(10), 3779-3887.
- [5] Simarmata, R. F., Tulloh, R., & Haryani, Y. S. (2018). Simulasi Jaringan Software Defined Network Menggunakan Protokol Routing OSPF Dan Ryu Controller. 4(3), 2887.
- [6] Anam, K., & Adria, R. (2017). Analisis Performa Jaringan Software Defined Network Berdasarkan Penggunaan Cost Pada Protokol Ruting Open Shortest Path First.
- [7] Anggara, S. M. (2015). Pengujian Performa Kontroler Software-defined Network (SDN): POX dan Floodlight. STEI ITB.
- [8] Kurniawan, R. C., Tulloh, R., & Irawati, I. D. (2020). Implementasi VPLS Pada Jaringan Software Defined Network (SDN) dengan menggunakan ONOS Controller Berbasis Raspberry-Pi 3.
- [9] Tulloh. Rohmat, Negara M. Ridha, Hidayat. Arif. 2015. Simulasi Virtual Local Area Network (VLAN) Berbasis Software Defined Network (SDN) Menggunakan POX

Controller, Jurnal Infotel.

- [10] Santoso, K. A. (t.thn.). Konfigurasi dan Analisis Performasi Routing OSPF Pada Jaringan LAN dengan Simulasi Cisco Packet Tracer Versi 6.2. Jurnal Kajian Teknik Elektro Vol.1 No.1 Universitas 17 Agustus 1945 Jakarta , 70.
- [11] M. N. Yaqin, R. Tulloh, I. D. Irawati, “Perancangan dan Implementasi Protocol Routing EBGp Pada Software Defined Network Menggunakan ONOS Controller,” 2019.
- [12] github.com. "Opennetworkinglab/Onos/Protocols" Diakses pada 12 januari 2021 , <https://github.com/opennetworkinglab/onos/tree/master/protocols>
- [13] Afan Romi, Virgono Agus, M. Rumani , " ANALISIS EFEK PENGGUNAAN KONTROLER RYU DAN POX PADA PERFORMANSI JARINGAN SDN," 2018



**UNIVERSITAS TELKOM**  
**FAKULTAS ILMU TERAPAN**  
**KARTU KONSULTASI**  
**SEMINAR PROPOSAL PROYEK TINGKAT**

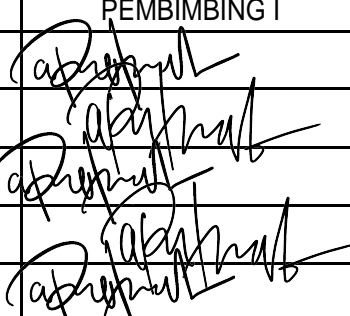
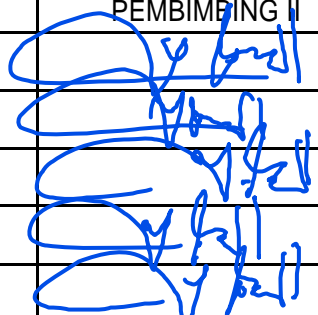
NAMA / PRODI : Usu Gunawan / D3 Teknologi Telekomunikasi NIM : 6705180021

JUDUL PROYEK TINGKAT :

ANALISIS PERBANDINGAN PERFORMANSI *ONOS*, *POX*, DAN *RYU CONTROLLER* PADA PROTOKOL ROUTING *OSPF* DAN *BGP* DI *SOFTWARE DEFINED NETWORK* BERBASIS *RASPBERRY PI*

CALON PEMBIMBING : I. Rohmat Tulloh,S.T.,M.T.

II. Dr. Indrarini Dyah Irawari,S.T.,M.T.

NO	TANGGAL	CATATAN HASIL KONSULTASI	TANDA TANGAN CALON PEMBIMBING I
1	22 Januari 2021	BAB 1 (SELESAI)	
2	22 Januari 2021	BAB 2 (SELESAI)	
3	22 Januari 2021	BAB 3 (SELESAI)	
4	22 Januari 2021	BAB 4 (SELESAI)	
5	22 Januari 2021	FINALISASI PROPOSAL	
6			
7			
8			
9			
10			
NO	TANGGAL	CATATAN HASIL KONSULTASI	TANDA TANGAN CALON PEMBIMBING II
1	20 Januari 2021	BAB 1 (SELESAI)	
2	20 Januari 2021	BAB 2 (SELESAI)	
3	20 Januari 2021	BAB 3 (SELESAI)	
4	20 Januari 2021	BAB 4 (SELESAI)	
5	20 Januari 2021	FINALISASI PROPOSAL	
6			
7			
8			
9			
10			