SKRIPSI

ANALISIS KINERJA PROTOKOL ROUTING OSPF MENGGUNAKAN CONTROLLER RYU DAN OPENDAYLIGHT PADA JARINGAN SOFTWARE DEFINED NETWORK (SDN)

Disusun dan diajukan oleh:

MUH. NAUFAL FALIQ
D121171503

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024

LEMBAR PENGESAHAN SKRIPSI

ANALISIS KINERJA PROTOKOL ROUTING OSPF MENGGUNAKAN CONTROLLER RYU DAN OPENDAYLIGHT PADA JARINGAN SOFTWARE DEFINED NETWORK (SDN)

Disusun dan diajukan oleh

Muhammad Naufal Faliq
D121171503

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Pada tanggal ..31 Juli 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,	Pembimbing Pendamping,
Dr. Eng. Muhammad Niswar, S.T., M.IT	Dr. Eng. Zulkifli Tahir, S.T., M.Sc
NIP. 19730922 199903 1 001	NIP. 19840403 201012 1 004

Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN Eng.
NIP. 19750716 200212 1 004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;
Nama : Muhammad Naufal Faliq
NIM : D121171503
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Analisis Kinerja Protokol Routing OSPF Menggunakan Controller RYU dan OpenDaylight Pada Jaringan Software Defined Network (SDN)

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitnya. Oleh karena itu, semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggung jawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keselur
uhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 31 Juli 2024

Yang Menyatakan

Muhammad Naufal Faliq
NIM. D121 17 1503

ABSTRAK

MUHAMMAD NAUFAL FALIQ. *Analisis Kinerja Protokol Routing OSPF Menggunakan Controller RYU dan OpenDaylight Pada Jaringan Software Defined Network (SDN)* (dibimbing oleh Dr. Eng. Muhammad Niswar, S.T., M.IT dan Dr. Eng. Zulkifli Tahir, S.T., M.Sc)

Jaringan *Software Defined Network* (SDN) adalah paradigma jaringan yang mengubah cara tradisional dalam mengelola dan mengontrol jaringan. Salah satu komponen kunci dalam SDN adalah penggunaan controller yang memfasilitasi pengaturan dinamis dan sentralisasi manajemen jaringan. Protokol routing OSPF (*Open Shortest Path First*) telah lama digunakan dalam jaringan tradisional untuk mengatur lalu lintas data dengan efisien. Penelitian ini bertujuan untuk menganalisis kinerja protokol OSPF dalam konteks SDN dengan menggunakan controller RYU dan OpenDaylight.

Metode penelitian yang digunakan meliputi pengaturan simulasi dengan topologi jaringan yang disimulasikan menggunakan perangkat lunak Mininet. Pengujian dilakukan dengan memvariasikan jumlah node dan karakteristik trafik untuk mengevaluasi kinerja protokol OSPF dalam hal *throughput, latency,* dan *packet loss*.

Kesimpulan dari penelitian ini adalah bahwa OSPF tetap menjadi pilihan yang solid untuk digunakan dalam lingkungan SDN, walaupun terdapat perbedan kinerja kedua controller. Dengan kemampuan adaptasi yang baik terhadap perubahan topologi jaringan yang dinamis. Saran untuk penelitian mendatang termasuk pengembangan strategi pengoptimalan OSPF yang lebih baik untuk SDN serta peningkatan keamanan dalam implementasi SDN yang melibatkan controller. Studi ini diharapkan dapat memberikan kontribusi yang berarti bagi pengembangan lebih lanjut dalam mengintegrasikan OSPF dengan SDN secara efektif.

Kata kunci : Software Defined Network (SDN), OSPF, RYU, OpenDaylight, routing, kinerja jaringan

ABSTRACT

MUHAMMAD NAUFAL FALIQ. *Performance Analysis of OSPF Routing Protocol Using RYU and OpenDaylight Controllers on Software Defined Networks (SDN)* (supervised by Dr. Eng. Muhammad Niswar, S.T., M.IT dan Dr. Eng. Zulkifli Tahir, S.T., M.Sc).

Software Defined Network (SDN) is a networking paradigm that changes the traditional way of managing and controlling networks. One of the key components in SDN is the use of controllers that facilitate dynamic settings and centralized network management. The OSPF (Open Shortest Path First) routing protocol has long been used in traditional networks to manage data traffic efficiently. This research aims to analyze the performance of the OSPF protocol in the SDN context using the RYU and OpenDaylight controllers.

The research method used includes a simulation setup with a simulated network topology using Mininet software. Testing was carried out by varying the number of nodes and traffic characteristics to evaluate the performance of the OSPF protocol in terms of throughput, latency, and packet loss.

The conclusion of this research is that OSPF remains a solid choice for use in an SDN environment, even though there are differences in the performance of the two controllers. With good adaptability to dynamic network topology changes. Suggestions for future research include developing better OSPF optimization strategies for SDN as well as improving security in SDN implementations involving controllers. This study is expected to provide a significant contribution to further development in effectively integrating OSPF with SDN.

Keywords : Software Defined Network (SDN), OSPF, RYU, OpenDaylight, routing, network performance

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
SDN	*Software Defined Network*
OSPF	*Open Shortest Path First*
QoS	*Quality of Service*
ODL	*Opendaylight*
LAN	*Local Area Network*
WAN	*Wide Area Network*
APIs	*Aplication Protocol Interfaces*
ONOS	*Open Network Operating System*
NTT	*Nippon Telegraph dan Telephone*
ICMP	*Internet Control Message Protocol*
TTL	*Time-to-Live*
IP	*Internet Protocol*
IGP	*Interior Gateway routing Potokol*
ISP	*Internet Service Provider*
LSAs	*Link-State Advertisements*
LSDB	*Link-State Database*
DoS	*Denial of service*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
HTTP	*Hypertext Transfer Protocol*
DNS	*Domain Name System*
ISP	*Internet Service Provider*
RIP	*Routing Information Protocol*
BGP	*Border Gateway Protocol*
IS-IS	*Intermediate System to Intermediate System*

DAFTAR LAMPIRAN

KATA PENGANTAR

Puji syukur kita panjatkan kepada Allah SWT yang telah memberikan limpahan kenikmatan dan kesempatan dalam menyelesaikan tugas akhir ini. Salawat serta salam kita panjatkan pada Nabi Muhammad SAW sebagai suri tauladan, pelita dalam kegelapan hingga akhir zaman sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul **"*Analisis Kinerja Protokol Routing OSPF Menggunakan Controller RYU dan OpenDaylight Pada Jaringan Software Defined Network (SDN)*"** guna memenuhi salah satu persyaratan dalam menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

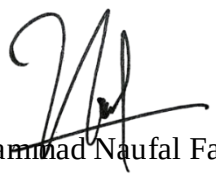
Penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari kesempurnaan karena menyadari segala keterbatasan yang ada. Dalam penulisan skripsi ini penulis menghadapi berbagai kendala dan masalah, namun itu tidak menurunkan semangat penulis dengan usaha yang maksimal dan kemampuan yang Allah berikan kepada penulis serta bantuan dan dukungan dari berbagai pihak, maka penulisan skripsi ini dapat selesai. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada :

1. Allah SWT, sebagai zat yang Maha segalanya sebagai penolong di setiap langkah hidup penulis,
2. Prof. Dr. Eng. Ir. Muhammad Isran Ramli, S.T., M.T., IPM., ASEAN Eng, selaku Dekan Fakultas Teknik Universitas Hasanuddin yang telah memberikan izin kepada penulis untuk melakukan penelitian,
3. Bapak Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng. selaku Ketua Ketua Program Studi Departemen Informatika Fakultas Teknik Universitas Hasanuddin atas bantuannya dan bimbingannya kepada penulis,
4. Bapak Dr. Eng, Muhammad Niswar S.T., M.IT. selaku pembimbing utama dan Bapak Dr. Eng. Zulkifli Tahir, S.T., M.Sc selaku pembimbing pendamping yang senantiasa menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa dalam mengarahkan penulis dalam penyusunan tugas akhir ini,

5. Bapak Robert, Bapak Zainuddin dan Ibu Yuanita serta segenap staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu kelancaran penyelesaian tugas akhir penulis,
6. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu semasa perkuliahan hingga penyelesaian tugas akhir penulis,
7. Alm Ayahanda Herman MP, Ibunda Lenni Marlina, Kakak Heli Kaishelmi Herman dan Adek Reyhan Faqih Ashuri yang tercinta atas segala bantuan, bimbingan, dorongan serta doa restu yang diberikan kepada penulis selama penyusunan skripsi.
8. Ir. H. Muhammad Ilham Musa atas segala bantuan, bimbingan, dorongan serta doa restu yang diberikan kepada penulis selama penyusunan skripsi.
9. Seluruh keluarga besar saya yang tidak bisa saya sebutkan satu persatu yang juga telah memberikan banyak sumbangsih baik secara materi maupun non-materi kepada penulis,
10. Anak Teknik, baik junior maupun senior dan terkhusus untuk teman-teman angkatan 2017 yang menemani penulis dalam mengarungi suka dan duka dalam menjalani dunia perkuliahan dan kemahasiswaan di Fakultas Teknik Universitas Hasanuddin,
11. Saudara seperjuangan penulis RECOGN17ER yang telah menemani dan mendukung perjalanan penulis sekaligus tempat berbagi keluh kesah selama menjadi mahasiswa teknik di Departemen Informatika Fakultas Teknik Universitas Hasanuddin,
12. Seluruh pihak yang tidak sempat disebutkan satu persatu yang telah banyak meluangkan tenaga, waktu, dan pikiran selama penyusunan tugas akhir ini.

Makassar, 31 Juli 2024

Penulis,



Muhammad Naufal Faliq

BAB I
PENDAHULUAN

1.1 Latar Belakang

Penggunaan jaringan komputer saat ini semakin kompleks dengan jumlah perangkat yang semakin banyak. Untuk mengatasi kompleksitasi ini, penggunaan *Software Defined Network (SDN)* menjadi solusi yang efektif. SDN memungkinkan administrator jaringan untuk mengelola jaringan dengan lebih mudah dan efisien, serta mempercepat implementasi perubahan pada jaringan. SDN merupakan sebuah konsep baru pada pendekatan dalam merancang, menyusun serta mengelola jaringan komputer. Konsep ini berkenaan dengan arsitektur perangkat jaringan seperti router, packet switch, switch LAN dan lain sebagainya. Sebuah perangkat jaringan memiliki dua bagian yaitu data plane dan control plane. Konsep dari SDN ini menerapkan pemisahan antara data plane dan control plane, yang mana data plane tetap berada pada perangkat jaringan, sedangkan control plane terdapat dalam sebuah entity terpisah yang dinamakan sebagai controller. (Dwi Rahmawan & Risqiwati, 2020)

Pada jaringan *Software Defined Network (SDN)* ada beberapa protocol, diantaranya Openflow. Openflow merupakan protocol yang memungkinkan server memberitahukan switch jaringan tempat mengirim paket yang dimana berfungsi sebagai penghubung antara controller yaitu termasuk dalam control plane dengan data plane melewati secure channel lalu ke flow tabel dan diteruskan ke user. (홍종욱, 2019)

Protokol Openflow dapat diimplementasikan di berbagai platform, salah satunya RYU dan OpenDaylight. Kedua platform ini memiliki kelebihan dan kekurangan masing – masing. Selain itu, kedua controller tersebut akan diimplementasikan pada protocol routing OSPF. Protokol OSPF merupakan protokol routing yang lebih kompleks dan dapat menghasilkan jalur terpendek pada jaringan. Penggunaan OSPF pada jaringan SDN akan memberikan keuntungan dalam mengatur jalur yang optimal pada jaringan. Teknologi SDN yang terus berkembang seiring dengan kebutuhan akan jaringan terutama dalam melakukan konfigurasi jaringan yang semakin kompleks maka performansi menjadi hal krusial

yang harus diperhatikan. Tolak ukur pengukuran performansi pada jaringan dikenal dengan *Quality of Service (QoS).* Penelitian ini dilakukan untuk mengetahui bagaimana analisi kinerja dari protocol routing OSPF menggunakan controller RYU dan OpenDaylight pada topologi jaringan tree berdasarkan parameter QoS. (Iryani et al., 2021)

Oleh karena itu, penelitian ini akan membahas tentang "Analisis Kinerja Protokol Routing OSPF Menggunakan Controller RYU dan Opendaylight Pada Jaringan Software Defined Network (SDN)". Penelitian ini diharapkan dapat memberikan informasi tentang perbedaan kinerja antara kedua platform controller pada jaringan SDN dengan menggunakan protokol OSPF. Informasi ini dapat membantu administrator jaringan dalam memilih platform controller yang sesuai untuk jaringan SDN yang sedang dikelola.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang diuraikan, maka rumusan masalah pada penelitian ini antara lain :

1. Bagaimana implementasi routing protokol OSPF pada jaringan SDN yang menggunakan controller RYU dan OpenDaylight ?
2. Bagaimana performa jaringan SDN yang menggunakan controller RYU dan OpenDaylight dengan menggunakan routing protokol OSPF ?
3. Manakah controller yang lebih optimal untuk digunakan pada jaringan SDN dengan menggunakan routing protokol OSPF, RYU atau Opendaylight ?

1.3 Tujuan Penelitian/Perancangan

Tujuan akhir dari penelitian ini yakni :

1. Mengimplementasikan routing protokol OSPF pada jaringan SDN yang menggunakan controller RYU dan OpenDaylight.
2. Menganalisis performa jaringan SDN yang menggunakan controller RYU dan OpenDaylight dengan menggunakan routing protokol OSPF.
3. Menentukan controller yang lebih optimal untuk digunakan pada jaringan SDN dengan menggunakan routing protokol OSPF, apakah RYU atau OpenDaylight.

1.4 Manfaat Penelitian/Perancangan

Manfaat yang akan didapat pada penelitian ini, yakni :

1. Memberikan pemahaman yang lebih baik tentang karakteristik controller RYU dan OpenDaylight pada jaringan SDN serta bagaimana implementasi routing protokol OSPF pada kedua controller tersebut.
2. Membantu administrator jaringan dalam memilih controller yang lebih optimal untuk digunakan pada jaringan SDN dengan menggunakan routing protokol OSPF, sehinggan dapat meningkatkan performa jaringan.
3. Memberikan informasi yang berguna bagi pengembangan teknologi jaringan SDN dan meningkatkan literatur penelitian di bidang jaringan komputer.
4. Sebagai referensi bagi peneliti lain yang ingin melakukan penelitian serupa mengenai perbandingan antara controller RYU dan OpenDaylight pada jaringan SDN dengan menggunakan routing protokol OSPF.
5. Meningkatkan kemampuan dan keterampilan penulis dalam melakukan penelitian dan mengembangkan teknologi jaringan SDN.

1.5 Ruang Lingkup/Asumsi perancangan

Berdasarkan beberapa hal yang dicantumkan pada rumusan masalah, maka batasan dari penelitian ini yakni :

1. Penelitian ini akan berfokus pada perbandingan antara controller RYU dan OpenDaylight pada jaringan SDN dengan menggunakan routing protokol OSPF.
2. Penelitian ini hanya akan membahas performa jaringan pada aspek routing protokol OSPF dan tidak membahas aspek lain seperti keamanan, manajemen, dan monitoring jaringan.
3. Menggunakan topologi jaringan tree yang terdiri dari beberapa switch, router dan host.
4. Penelitian ini tidak akan membahas mengenai implementasi jaringan SDN secara keseluruhan, namun hanya akan membahas aspek yang berkaitan dengan perbandingan controller RYU dan OpenDaylight dengan menggunakan routing protokol OSPF.
5. Penelitian ini hanya akan dilakukan di lingkungan simulasi menggunakan perangkat lunak Mininet.

BAB II
TINJAUAN PUSTAKA

2.1 Jaringan Komputer

Jaringan komputer adalah kumpulan komputer atau perangkat komunikasi yang terhubung satu sama lain untuk berbagi sumber daya, seperti data, aplikasi, dan layanan. Tujuan dari jaringan komputer adalah memungkinkan komunikasi dan pertukaran informasi antara berbagai perangkat dalam suatu sistem. Jaringan komputer dapat memiliki cakupan yang bervariasi, mulai dari jaringan lokal (*Local Area Network* atau LAN) yang terbatas pada area geografis kecil, hingga jaringan luas (*Wide Area Network* atau WAN) yang dapat mencakup area geografis yang lebih besar, bahkan mencakup wilayah global.

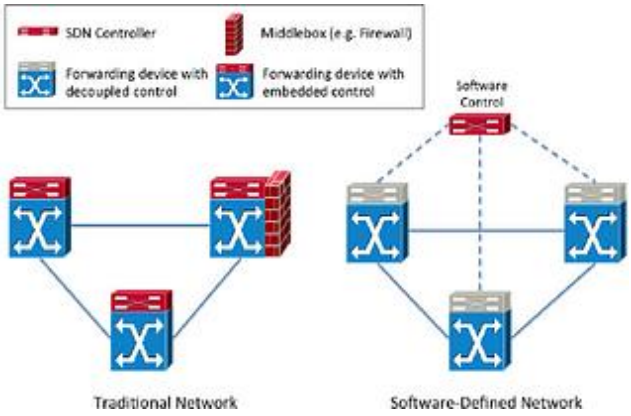
2.2 Software Defined Network (SDN)

Awal mula terciptanya teknologi *Software Defined Networking* dimulai tidak lama setelah Sun Mycrosystems merilis Java pada tahun 1995, namun pada saat itu belum cukup membangunkan para periset untuk mengembangkan teknologi tersebut. Baru pada tahun 2008 *Software Defined Network* ini dikembangkan di UC Barkeley and Stanford University. Dan kemudian teknologi tersebut mulai di promosikan oleh Open Networking Foundation yang didirikan pada tahun 2011 untuk memperkenalkan teknologi SDN dan OpenFlow. (Abidin, 2021)

2.2.1 Pengertian Software Defined Network (SDN)

Jaringan tradisional dengan arsitektur yang kaku dan kompleksitas yang tinggi sering kali tidak mampu memenuhi tuntutan modern ini. Di sinilah konsep *Software Defined Network (SDN)* hadir sebagai solusi inovatif SDN memungkinkan pemisahan antara lapisan kontrol dan lapisan data dalam jaringan, memberikan kemampuan untuk mengelola jaringan secara dinamis melalui perangkat lunak yang terpusat.

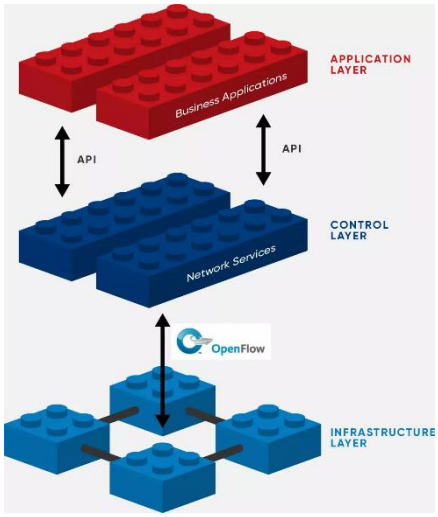
Software Defined Network (SDN) diusulkan untuk mengatasi kelemahan jaringan tradisional. Sebagai paradigma jaringan baru, SDN merevolusi teknologi jaringan dengan mematahkan ide dasar jaringan tradisional.



Gambar 2. 1 Perbedaan (a) Jaringan Tradisional dan (b) Jaringan SDN

Perbedaan mendasar antara jaringan tradisional dengan *Software Defined Network (SDN)* adalah penempatan fungsi control dan fordward. Pada jaringan tradisional fungsi control dan fordward ditempatkan pada device yang sama yaitu router. Sedangkan pada jaringan SDN fungsi kontrol ditempatkan pada software terpusat (controller) dan fungsi fordward ditempatkan pada suatu perangkat kosong berupa switch (fordwarding device). (Sulfiana, 2019)

SDN merupakan sebuah arsitektur jaringan yang sedang berkembang yang memisahkan antara data plane dan control plane dalam perangkat jaringan yang kemudian membuat control plane menjadi independent dan dapat diprogram. Pemisahan yang dilakukan antara data plane dan control plane menjadikannya abstraksi dan infrastruktur jaringan dan aplikasi yang membuat jaringan menjadi sebuah keberadaan virtual.



Gambar 2. 2 Arsitektur SDN

2.2.2 Konsep Software Defined Network (SDN)

Konsep *Software Defined Network (SDN)*, pertama kali diperkenalkan oleh Martin Casado di Universitas Stanford pada tahun 2007 dengan tulisan pada jurnalnya berjudul "Ethane : Talking Control of the Enterprise". (Thomas et al., 2018). Konsep dasar dari SDN memisahkan fungsi kontrol (kontrol plane) dan pengalihan lalu lintas (data plane). Fungsi kontrol ditempatkan di pusat, sementara perangkat keras jaringan (switches, router) bertanggung jawab untuk pengalihan lalu lintas sesuai dengan intruksi yang diberikan oleh kontroler. Kontroler pusat berperan sebagai otak jaringan. Ia mengambil keputusan terkait routing dan manajemen jaringan. Keputusan ini kemudian dikirimkan ke perangkat keras jaringan untuk diimplementasikan. Dalam SDN terdapat protocol yang menonjol yaitu Openflow. SDN menggunakan protocol komunikasi Openflow antara kontroler untuk mengirim intruksi kepada perangkat keras jaringan dan mengatur aliran lalu lintas.

Dari segi programmability, SDN memberikan kemampuan untuk mengelola jaringan secara programatik. Administrator dapat menggunakan antarmuka pemrograman untuk mengatur aturan, kebijakan, dan konfigurasi jaringan. Dengan kontrol yang terpusat dan kemampuan programatik, SDN memberikan kecepatan dan fleksibilitas yang tinggi dalam menyesuaikan jaringan dengan perubahan kebutuhan dan kondisi.

SDN memanfaatkan konsep virtualisasi untuk menciptakan jaringan virtual yang terisolasi. Hal ini memungkinkan penggunaan yang lebih efisien dari sumber daya jaringan dan mempermudah manajemen berbagai layanan. SDN mendukung orkestrasi, yang mencakup otomatisasi konfigurasi dan manajemen jaringan. Hal ini memungkinkan jaringan beradaptasi secara otomatis terhadap perubahan kondisi atau permintaan. Dalam hal ini, SDN memungkinkan pengelolaan jaringan berbasis kebijakan, dimana administrator dapat menentukan kebijakan secara terpusat dan mengimplementasikannya secara konsisten di selurh jaringan.

Oleh karena itu, konsep – konsep ini mengarah pada paradigma SDN yang memberikan pengelolaan yang lebih sentral, responsif, dan mudah diatur dalam konteks jaringan. SDN memberikan fleksibilitas dan skalabilitas yang lebih tinggi,

menghadirkan solusi yang dapat disesuaikan dengan kebutuhan bisnis yang berubah – ubah.

2.2.3 Tujuan Utama Software Defined Network (SDN)

Tujuan utama dari SDN untuk mencapai pengelolaan jaringan yang lebih baik dengan tingkatan dan kompleksitas yang besar serta memastikan bahwa semua keputusan dari sistem kontrol dibuat dari titik pusat (kontroller). SDN memperkenalkan suatu metode untuk meningkatkan tingkat abstraksi pada konfigurasi jaringan, menyediakan mekanisme yang secara otomatis bereaksi terhadap perubahan yang sering terjadi dan terus menerus untuk jaringan. SDN bertujuan untuk menyentralisasi kontrol jaringan, memungkinkan pengambilan keputusan yang lebih pintar dan efisien untuk routing dan manajemen sumber daya. SDN memberikan fleksibilitas yang tinggi dengan kemampuan penyesuaian jaringan secara dinamis, memungkinkan skalabilitas yang lebih baik sesuai dengan kebutuhan bisnis. SDN bertujuan untuk mengoptimalkan penggunaan sumber daya jaringan dan memberikan platform yang efisien untuk mengelola dan mengonfigurasi perangkat keras jaringan. SDN berupaya meningkatkan kontrol keamanan dan memberikan pengelolaan jaringan yang lebih efisien melalui perangkat lunak. SDN dirancang untuk mendukung pengembangan aplikasi dan inovasi dalam penggunaan sumber daya jaringan, sehingga merangsang perkembangan teknologi, sehinggan SDN mempermudah penerapan berbagai jenis macam aplikasi jaringan yang dapat diprogram pada controllernya, karena sifat controller SDN yang programmable tersebut. (Attamimi et al., 2017)

2.2.4 Karakteristik Software Defined Network (SDN)

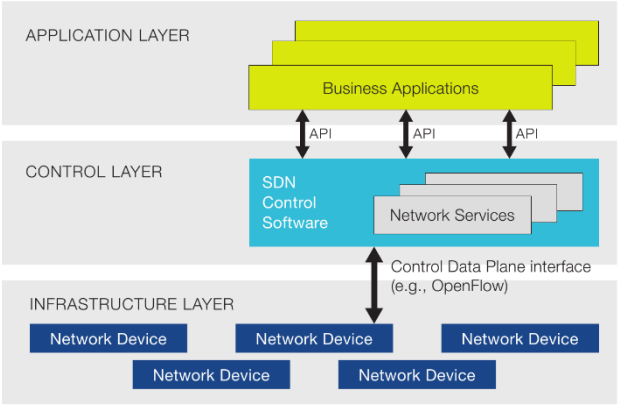
Software Defined Network (SDN) memiliki beberapa karakteristik kunci yang membedakannya dari pendekatan tradisional dalam pengelolaan jaringan, diantaranya :

1. SDN memisahkan fungsi kontrol plane, yang mengelola keputusan perutean dari data plane yang mengirimkan paket data. Ini memberikan fleksibilitas lebih besar dalam pengelolaan dan pengaturan jaringan.
2. SDN memungkinkan pengaturan dan konfigurasi jaringan secara dinamis melalui perangkat lunak.

3. SDN memberikan kontrol jaringan yang lebih sentral melalui pusat kontrol. Ini memungkinkan pengambilan keputusan jaringan yang lebih terpusat dan koheren.
4. SDN mendukung virtualisasi sumber daya jaringan, memungkinkan pembagian dan isolasi sumber daya secara efisien untuk mendukung multiple tenants atau layanan.
5. SDN sering kali didukung oleh standar terbuka dan protokol yang dapat diakses, memungkinkan interoperabilitas antara berbagai vendor dan perangkat.
6. SDN menyediakan antarmuka yang dapat diprogram (API) yang memungkinkan aplikasi dan perangkat lunak manajemen jaringan untuk berkomunikasi dengan perangkat keras jaringan.
7. SDN mendukung otomatisasi konfigurasi jaringan. Perangkat lunak kontrol dapat secara otomatis merespon perubahan kondisi jaringan dan menyesuaikan konfigurasi sesuai kebutuhan.
8. SDN memungkinkan jaringan untuk beradaptasi dengan perubahan secara dinamis, memungkinkan penyesuaian cepat terhadap permintaan dan kondisi jaringan yang berubah.
9. SDN dirancang untuk skalabilitas yang baik, memungkinkan pertumbuhan jaringan tanpa mengorbankan kinerja atau menghasilkan kompleksitas yang tidak terkelola.
10. SDN memungkinkan pengumpulan data analitik jaringan yang mendalam yang dapat digunakan untuk pemantauan, analisis dan pengambilan keputusan yang lebih terkait performa jaringan.

2.2.5 Arsitektur Software Defined Network (SDN)

Pada arsitektur *Software Defined Network (SDN)* setiap layer dapat bekerja secara independent dan berkomunikasi melalui antarmuka jaringan untuk memberikan fungsi berlapis dari perangkat fisik yang berbeda. Aspek arsitektur ini memungkinkan administrator jaringan untuk mengatasi beberapa tantangan dalam dunia jaringan komputer (Adrian, 2017).



Gambar 2. 3 Arsitektur Software Defined Network

Gambaran logis arsitektur Software Defined Network pada gambar 2.3 menunjukan terdapat 3 layer pada arsitektur *Software Defined Network (SDN)*, yaitu :

1. Aplication Layer

   Aplication layer merupakan layer yang berperan sebagai antar muka untuk memudahkan pengelola jaringan dalam melakukan fungsi konfigurasi, fungsi kontrol dan fungsi evaluasi. Pada lapisan ini terbentuk dari intergrasi dengan kontrol layer yang memberikan informasi tentang jaringan yang selanjutnya ditampilkan dalam bentuk yang dapat dipahami oleh pengelola jaringan (Irmawati et al., 2017).

2. Control Layer

   Lapisan kontrol berfungsi sebagai otak dari arsitektur SDN. Disini, perangkat lunak kontrol SDN berada bertugas mengambil keputusan terkait manajemen jaringan. Perangkat lunak kontrol inilah yang memahami topologi jaringan, kondisi, dan kebutuhan aplikasi. Contoh teknologi di lapisan kontrol termasuk controller SDN seperti OpenDaylight, ONOS, atau RYU.

3. Infrastructure Layer

   Lapisan infrastruktur atau data plane merupakan bagian fisik jaringan yang terdiri dari perangkat keras seperti switch dan router. Perangkat keras ini bertanggung jawab untuk mengirimkan paket data sesuai dengan intruksi

yang diberikan oleh lapisan kontrol. Pada lapisan ini, terdapat perangkat jaringan yang dapat diatur dan dikonfigurasi oleh perangkat lunak kontrol.

2.2.6 Keunggulan Arsitektur Software Defined Network (SDN)

Software Defined Network (SDN) memiliki beberapa keunggulan yang membuatnya menjadi pendekatan yang menarik dalam pengelolaan jaringan. Keungguan SDN antara lain :

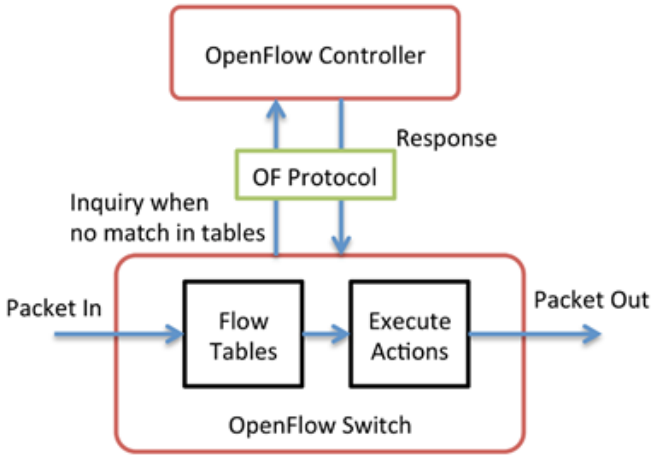
1. Dapat diprogramkan secara langsung karena kontrol panelnya terpisah dari *forwarding plane* (data plane).
2. Pemisahan antara kontrol plane dan data plane menyebabkan administrator jaringan dapat secara dinamis menyesuaikan *flow traffic network* sesuai dengan kebutuhan instusi.
3. *Network intelligence* mengelola satu gambaran umum yang utuh mengenai jaringan.
4. SDN memungkinkan jaringan administrator untuk mengelola sumber daya jaringan dengan sangat cepat.
5. Controller plane tidak bergantung pada *vendor-specifik device* atau protokol tertentu.
6. Penyajian antar muka yang lebih baik sebagai aplikasi manajemen jaringan terpusat.

2.3 Protokol Openflow

Jadi awal mula dari Openflow merupakan protokol terbilang relatif baru yang dirancang dan diimplementasikan di Stanford University pada tahun 2008. Sejak dimulainya standarisasi Protokol Openflow telah berkembang. Protokol Openflow adalah salah satu jenis dari APIs (*Aplication Protocol Interfaces*) dalam jaringan SDN yang digunakan untuk mengontrol dan mengatur *traffic flows* pada switch dalam sebuah jaringan, jadi singkatnya kontrol plane berkomunikasi dengan data plane melalui Openflow. Openflow dapat bekerja pada switch dari berbagai vendor.

Openflow adalah protokol paling utama pada SDN. Openflow terletak diantara kontrol plane dan data plane. Openflow mengatur routing dan pengiriman paket ketika melalui switch. Dalam sebuah jaringan, switch berfungsi meneruskan paket yang melalui suatu port tanpa mampu membedakan tipe protokol data yang dikirimkan. Openflow memungkinkan untuk mengakses dan mampu memanipulasi data plane secara langsung dari perangkat jaringan seperti switch dan router baik secara fisik maupun virtual (Ummah, 2016).

Agar Openflow dapat bekerja sesuai tujuan dan fungsinya, Openflow mempunyai 2 komponen penting yaitu Openflow Controller dan Openflow Switch.



Gambar 2. 4 Openflow Controller dan Openflow Switch

2.3.1 Openflow Controller

Openflow controller bertugas mengontrol path, memformulasikan flow dan mengatur kerja dari Openflow switch. Terdapat beberapa Openflow controller yang dapat digunakan seperti NOX (C base), POX (phyton base), dan Floodlight (java base).

2.3.2 Openflow Switch

Pada perangkat Openflow switch sendiri dapat dibagi menjadi tiga bagian yaitu :

1. Flow Table

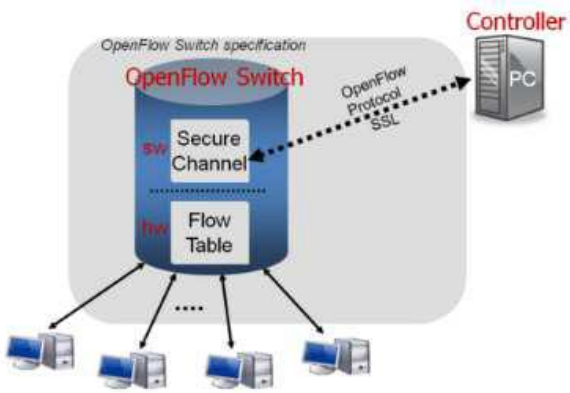
   Sebuah *flow table* yang mengindikasikan bahwa switch harus memroses flow yang ada di dalamnya. Daftar flow ini dibuat berdasarkan *actions* yang mana bersinggungan langsung dengan setiap flow.

2. Secured Chanel

   Sebuah saluran yang aman dibutuhkan untuk menghubungkan switch dengan controller. Melalui saluran ini, Openflow menyediakan jalur komunikasi antara switch dan controller melalui protokol yang disebut protocol Openflow.

3. Protokol Openflow

   Protokol ini menyediakan sebuah standar dan komunikasi terbuka antara controller dan switch. Openflow controller menentukan ke interface manakah flow akan diterapkan dari flow table (Khoerul & Ronald, 2017).



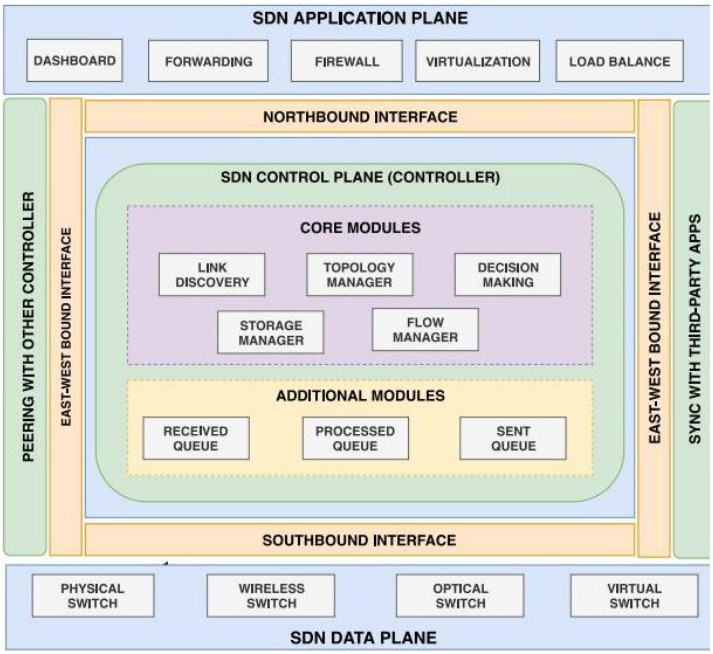
Gambar 2. 5 Mekanisme Switch Openflow

2.4 Controller Software Defined Network (SDN)

Controller SDN adalah aplikasi SDN yang mengelola flow kontrol untuk mengaktifkan *intelligence networking*. Controller SDN bekerja berdasarkan protokol seperti Openflow yang memungkinkan server memberitahu kemana paket dikirimkan. Perangkat meneruskan paket data yang diterima berdasarkan aturan yang ditetapkan dari controller. Controller menentukan apa yang harus dilakukan dengan paket dan jika perlu mengirimkan aturan baru untuk perangkat sehingga dapat menangani paket data di masa depan dengan cara yang sama (Indriani Lestariningati, 2018).

Pada saat ini, beberapa controller yang terkenal dan telah berkembang diantaranya NOX, POX, Floodlight, OpenDaylight, Open Network Operating

System (ONOS), dan RYU. Namun, sejumlah pengontrol dan varian rasa lain tersedia dalam literatur yang ada (Zhu et al., 2019). Pada penelitian ini menggunakan Controller RYU dan OpenDaylight.



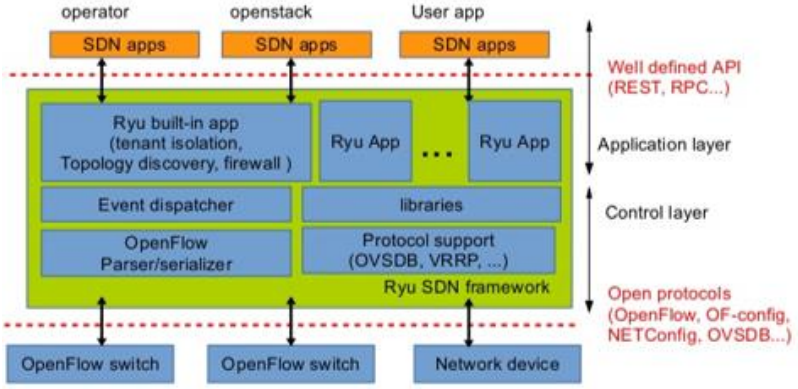
Gambar 2. 6 Arsitektur SDN Controller

2.4.1 RYU Controller

RYU adalah kerangka kerja jaringan SDN berbasis komponen sumber terbuka yang dikembangkan oleh *Nippon Telegraph dan Telephone (NTT)*. RYU adalah pengontrol SDN yang diimplementasikan sepenuhnya dengan phyton. Nama RYU berasal dari kata dalam bahasa Japan yang berarti “mengalir” yang dimana merupakan naga Japan, salah satu dewa air. Ia mengelola kontrol “aliran” untuk mengaktifkan kecerdasan jaringan. Controller RYU menyediakan komponen perangkat lunak dengan antarmuka program aplikasi (API) yang terdefinisi dengan baik sehingga dapat memudahkan pengembang untuk membuat aplikasi manajemen dan kontrol jaringan baru.

Controller RYU pada SDN memiliki tiga lapisan. Lapisan atas dikenal sebagai lapisan atas yang terdiri dari aplikasi logika bisnis dan jaringan. Lapisan tengah dikenal sebagai lapisan kontrol atau kerangka SDN yang terdiri dari layanan jaringan, sedangkan untuk lapisan bawah dikenal sebagai lapisan infrastruktur yang terdiri dari perangkat fisik dan virtual (Time & Time, 2020).

RYU mendukung berbagai protokol untuk mengelola perangkat jaringan, seperti OpenFlow, Netconf, OF-config dan lain sebagainya. Pada OpenFlow sendiri, RYU mendukung sepenuhnya 1.0, 1.2, 1.3, 1.4, 1.5 dan Nicira Extensions. Semua kode tersedia secara bebas di bawah lisensi Apache 2.0 Ryu berbasis bahasa phyton dan bersifat Open Source (Edgar et al., 2019).



Gambar 2. 7 Arsitektur Controller RYU

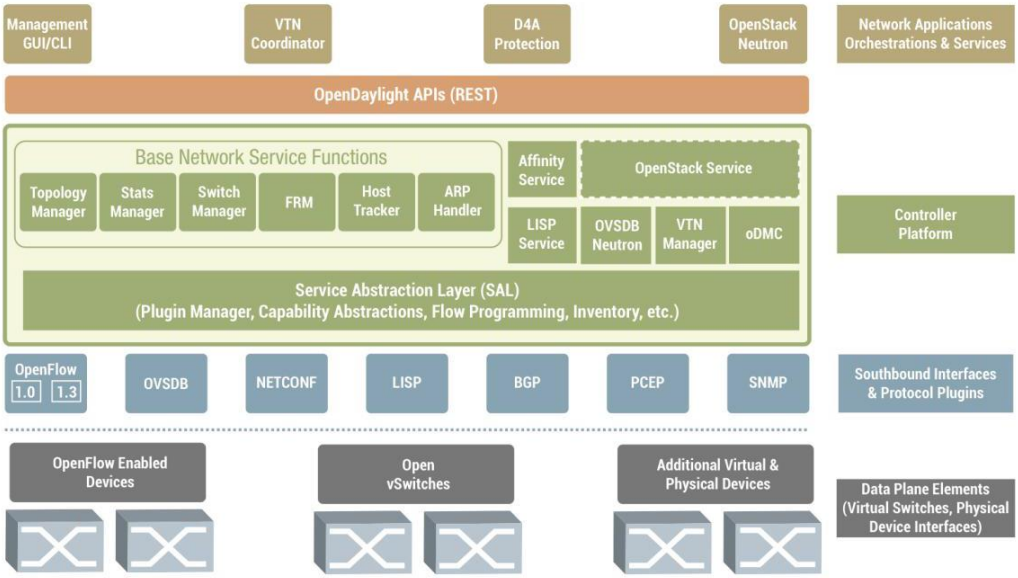
2.4.2 OpenDaylight

Dalam dunia jaringan yang terus berkembang, pengelolaan dan pengaturan jaringan yang efektif menjadi semakin penting. OpenDaylight menyediakan kerangka kerja yang fleksibel dan kuat untuk membangun solusi SDN yang *scalable* dan *interoperable*.

Dengan fitur-fiturnya yang canggih dan komunitas pengembang yang aktif, OpenDaylight tidak hanya memungkinkan pengelolaan jaringan yang lebih efisien tetapi juga mempercepat inovasi dalam teknologi jaringan. Dalam pengantar ini, kita akan mengeksplorasi konsep dasar OpenDaylight, fungsionalitasnya, dan peran pentingnya dalam merevolusi manajemen jaringan modern.

OpenDaylight merupakan Controller SDN yang dikembangkan oleh komunitas pengembang terbuka LINUX dan merupakan jawaban atas kebutuhan industri pada jaringan yang memiliki kemampuan pemrograman mandiri (Edgar et al., 2019). OpenDaylight didukung oleh IBM, Cisco, Juniper, VMWare dan beberapa vendor jaringan besar lainnya. OpenDaylight sendiri bersifat OpenSource dan diimpelemntasikan di Java.

OpenDaylight SDN Controller memiliki tiga lapisan. Lapisan atas terdiri dari aplikasi logis bisnis dan jaringan. Lapisan tengah merupakan lapisan kerangka kerja, sedangkan lapisan bawah terdiri dari perangkat fisik dan virtual. OpenDaylight bertujuan untuk mengurangi vendor, mengunci sehingga mendukung protokol selain OpenFlow dan BGP-LS sebagai plugin terpisah (Khattak et al., 2014)



Gambar 2. 8 Arsitektur Pengontrol OpenDaylight SDN

2.5 Miniet

Mininet merupakan sebuah emulator jaringan sumber terbuka yang memungkinkan pengguna untuk membuat dan menjalankan topologi jaringan virtual di dalam satu sistem komputer. Dengan kata lain, mininet memungkinkan anda membuat jaringan virtual yang terdiri dari berbagai perangkat seperti switch, router, dan host yang dapat diatur dan di uji tanpa memerlukan perangkat keras fisik.

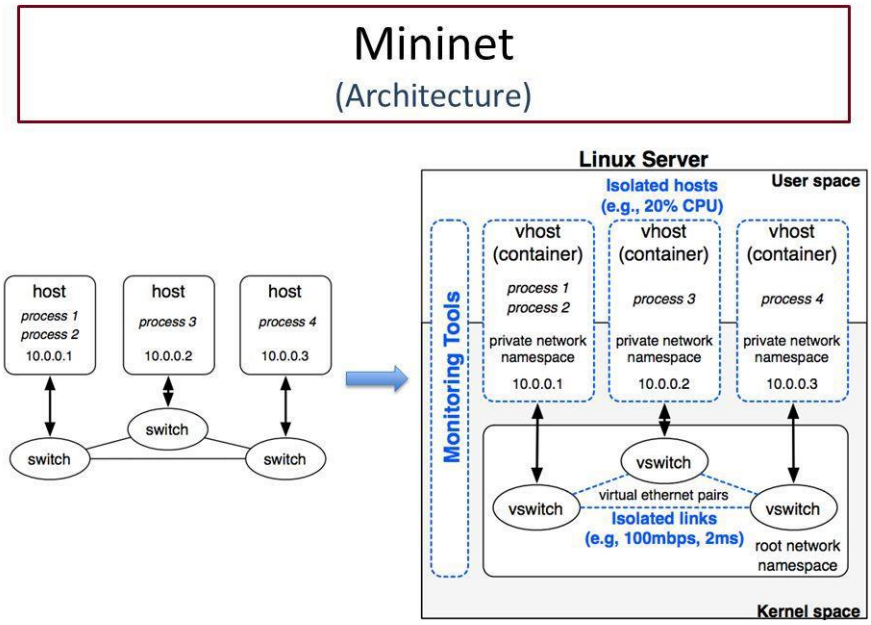
Mininet berfungsi sebagai emulator atau simulator jaringan, memungkinkan pengguna untuk membuat dan menguji topologi jaringan virtual tanpa harus memiliki perangkat keras jaringan fisik. Secara sederhana mininet ini berfungsi untuk emulasi pada bagian data path untuk mengetes konfigurasi jaringan SDN.

Sedangkan untuk melakukan testing pada mininet dapat dilakukan command “sudo mn” (Sudiyatmoko et al., 2016).

Mininet bersifat sumber terbuka, yang berarti kode sumbernya dapat dikases oleh siapa saja. Hal ini memungkinkan pengguna untuk memodifikasi dan mengadaptasi sesuai dengan kebutuhan mereka. Mininet berguna terutama untuk keperluan pengembangan dan pengujian di lingkungan pengembangan atau penelitian. Ini memberikan fleksibilitas kepada pengguna untuk menciptakan dan menguji berbagai skenario jaringan tanpa harus menghadapi kompleksitas dan biaya yang terkait dengan perangkat keras fisik.

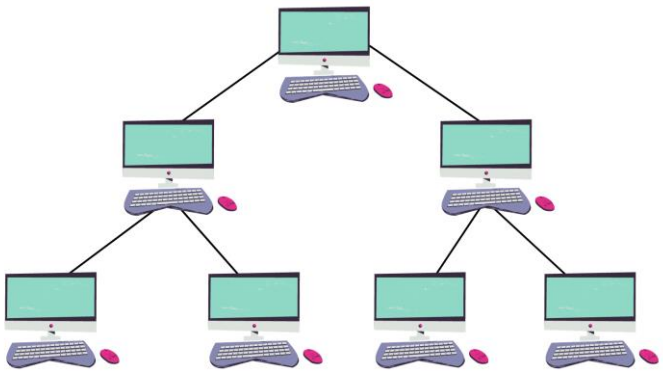
Mininet merupakan solusi yang dianggap paling unggul dalam hal kemudahan penggunaan, performansi, akurasi, dan skalabilitas. Ia mampu menyediakan lingkungan yang relastis dan nyaman dengan harga yang murah.

Mininet menggunakan pendeketan *light-wight* menggunakan virtualisasi level OS mencakup proses – proses dan namespace jaringan, sehingga memungkinkan dilakukannya simulasi jaringan dengan skala sangat besar (hingga ratusan node). Mininet dapat menciptakan jaringan virtual yang relastis, menjalankan real kernel, switch dan kode aplikasi, pada single machine baik berupak physical maupun virtual atau cloud (Kementrian Kesehatan Republik Indonesia, 2018).



Gambar 2. 9 Arsitektur Mininet

2.6 Topologi Jaringan Tree



Gambar 2. 10 Topologi Jaringan Tree

Topologi jaringan pohon atau tree network adalah suatu jenis konfigurasi jaringan komputer yang menggabungkan dua atau lebih struktur topologi berbeda. Dalam topologi ini, beberapa pohon atau cabang jaringan dihubungkan melalui satu titik pusat atau simpul utama yang disebut dengan root atau akar. Pusat ini berperan sebagai pengatur utama atau pemberi layanan kepada seluruh bagian jaringan.

Karakteristik utama dari topologi jaringan pohon melibatkan struktur hierarki, dimana setiap cabang atau pohon dari simpul utama dapat memiliki cabang – cabang lebih kecil lagi. Pusat atau simpul utama memainkan peran penting dalam pengiriman data antar bagian – bagian jaringan.

Keuntungan dari topologi pohon melibatkan kemudahan dalam manajemen dan organisasi, karena hierarki yang jelas membuatnya lebih muda untuk memahami dan mengelola bagian – bagian jaringan. Namun, kelemahannya termasuk kerentanan terhadap kegagalan dalam simpul utama. Jika simpul utama gagal, maka seluruh bagian jaringan yang terhubung dengannya dapat terpengaruh.

Topologi tree sering digunakan dalam jaringan bisnis dan organisasi besar karena kemampuannya untuk menyediakan struktur yang terorganisir dan mudah dikelola. Oleh karena itu, dalam memilih topologi jaringan, kita juga harus mempertimbangkan kebutusan spesifik dari suatu lingkungan dan potensi resiko kegagalan sistem.

2.7 Wireshark

Gambar 2. 13 Logo Wireshark

Wireshark adalah sebuah perangkat lunak open source yang berfungsi sebagai analisis paket jaringan. Ini berarti bahwa Wireshark dapat digunakan untuk memantau dan menganalisis data yang dikirim melalui jaringan komputer, termasuk data yang dikirim melalui kabel jaringan, jaringan nirkabel (Wi-Fi) dan berbagai protokol jaringan lainnya.

Dengan menggunakan Wireshark, pengguna dapat menangkap dan menganalisis paket data yang dikirim dan diterima oleh komputer atau perangkat jaringan lainnya. Hal ini memungkinkan pengguna pengguna untuk memahami interaksi antara berbagai perangkat dalam jaringan, mendeteksi masalah jaringan, memeriksa keamanan jaringan, serta memvalidasi kinerja jaringan.

Fitur utama dari Wireshark meliputi kemampuan untuk menangkap dan menyimpan paket data dalam berbagai format, menampilkan detail lengkap dari setiap paket termasuk informasi header dan payload, menyediakan filter untuk menampilkan paket yang spesifik, dan mendukung berbagai protokol jaringan seperti TCP, UDP, HTTP, DNS, dan banyak lagi.

Wireshark sering digunakan oleh administrator jaringan, insinyur jaringan, penelitian kemanan dan professional IT lainnya untuk memecahkan masalah jaringan, menganalisis kinerja, dan melakukan penelitian terkait dengan protokol jaringan. Keunggulan utama dari Wireshark adalah bahwa itu merupakan perangkat lunak sumber terbuka, sehingga dapat diunduh dan digunakan secara gratis oleh siapapun.

2.8 Internet Control Message Protokol (ICMP)

Internet Control Message Protocol (ICMP) merupakan salah satu protokol dalam suite protokol internet (TCP/IP) yang digunakan untuk mengirim pesan kontrol dan kesalahan antara perangkat dalam jaringan. ICMP bekerja diatas lapisan IP (Internet Protocol) dan digunakan untuk berbagai tujuan, termasuk :

1. Reporting Errors

   ICMP digunakan untuk melaporkan kesalahan yang terjadi selama pengiriman paket data, seperti ketika tujuan tidak dapat dicapai (host tidak dapat dijangkau), TTL (*Time-to-Live*) habis, atau ketika sebuah paket dibuang oleh router karena kelebihan beban.

2. Ping dan Traceroute

   Dua perintah yang paling umum digunakan yang berbasis ICMP adalah ping dan traceroute. Ping digunakan untuk menguji koneksi ke sebuah host dengan mengirimkan pesan ICMP Echo Request dan menerima balasan ICMP Echo Reply. Traceroute digunakan untuk melacak rute yang diambil oleh paket data melalui jaringan dengan mengirimkan serangan paket ICMP dan melacak respons dari setiap node dalam perjalanan

3. Konfigurasi da Pengujian Jaringan

   ICMP dapat digunakan untuk pengujian dan konfigurasi jaringan, seperti dengan pengujian kecepatan koneksi menggunakan perangkat lunak seperti *iperf* atau dengan pengujian ketersediaan host menggunakan alat monitoring jaringan

4. Pemberitahuan Mengenai Pemutusan Jaringan

   ICMP juga digunakan untuk memberikan pemberitahuan kepada pengguna atau perangkat lain dalam jaringan mengenai kondisi jaringan, seperti pemutusan jaringan yang tidak terduga atau keadaan darurat lainnya.

Meskipun ICMP memberikan banyak manfaat dalam manajemen dan pengoperasian jaringan, perlu diingat bahwa beberapa jenis serangan jaringan, seperti serangan ping atau serangan ICMP flood, juga dapat di eksploitasi menggunakan protokol ini. Oleh karena itu, dalam konfigurasi jaringan yang aman, ICMP mungkin perlu dikontrol dengan hati – hati.

2.9 Quality of Service (Qos)

Quality of Service (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu service. QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu service (Cintasari, 2018).

Tujuan dari *QoS (Quality of Service)* adalah untuk memastikan kualitas pengalaman pengguna dalam jaringan komputer atau system komunikasi. QoS bertujuan untuk mengatur dan mengontrol lalu lintas data dalam jaringan dengan cara memprioritaskan aplikasi atau layanan yang lebih penting, serta memastikan pengiriman yang tepat waktu, keandalan dan kinerja yang memadai. Adapun parameter yang akan digunakan untuk analisis kinerja controller yaitu *throughput, latency dan packet loss.*

2.9.1 Throughput

Throughput yaitu jumlah data yang berhasil ditransfer dari satu titik ke titik lain dalam jaringan dalam waktu tertentu. Biasanya diukur dalam bit per detik (bps), kilobit per detik (kbps) atau megabit per detik (mbps).

Adapun persamaan perhitungan *throughput* :

$$Throughput = \frac{jumlah\ data\ yang\ diterima}{lama\ pengamatan}$$

Table 2. 1 Kategori Throughput

Kategori Degredasi	Throughput	Indeks
Sangat Baik	>100 Mbps	4
Baik	50-100 Mbps	3
Cukup	10-50 Mbps	2
Buruk	1-10 Mbps	1

Sumber: TIPHON

2.9.2 Latency

Latency adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Latency* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Pada jaringan SDN, *latency* didefinisikan sebagai tanggapan yang dapat diberikan oleh controller pada setiap detiknya (Sulfiana, 2019).

Table 2. 2 Kategori Latency

Kategori Degredasi	Latency	Indeks
Sangat Baik	<50 ms	4
Baik	100-200 ms	3
Cukup	200-300 ms	2
Buruk	>300 ms	1

Sumber: TIPHON

2.9.3 Packet Loss

Packet loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena collison dan congestion pada jaringan. Hal ini berpengaruh pada semua aplikasi karena retransmisi akan mengurangi efesiensi jaringan secara keseluruhan meskipun jumlah bandwidth cukup tersedia untuk aplikasi – aplikasi tersebut.

Dalam SDN, *packet loss* merupakan jumlah paket yang hilang pada suatu jaringan yang disebabkan ketika satu atau lebih paket yang dikirim melalui jaringan IP gagal sampai ditempat tujuan (Sulfiana, 2019). Untuk mencari nilai dari *packet loss* dapat dihitung dengan persamaan :

$$Packet\ loss = \frac{(paket\ data\ dikirim - paket\ data\ diterima)}{paket\ data\ yang\ dikirim}\ x\ 100\%$$

Indeks dan kategori Packet loss dapat dilihat pada table 2.

Table 2. 3 Kategori Packet loss

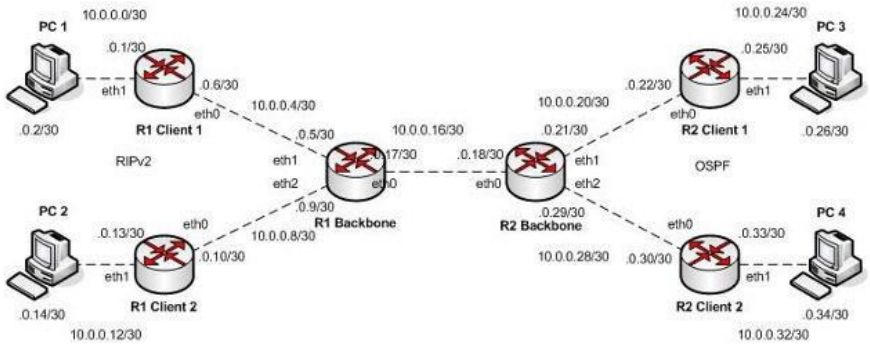
Kategori Degredasi	Packet Loss	Indeks
Sangat Baik	0 – 2%	4
Baik	3 – 14%	3
Cukup	15 – 24%	2
Buruk	>25%	1

Sumber: TIPHON

2.10 Iperf

Iperf adalah alat pengukuran bandwidth dan kinerja jaringan yang open-source dan lintas platform. Alat ini membantu dalam mengevaluasi dan menganalisis throughput jaringan TCP dan UDP. *Iperf* dirancang untuk memberikan hasil yang detail dan akurat, memungkinkan pengguna untuk memahami kemampuan jaringan mereka dengan lebih baik. Tool ini juga bersifat freeware. Tool ini hanya dapat dijalankan melalui command prompt dan tidak memiliki tampilan GUI (Stmik-amik-riau et al., 2016)

2.11 Routing Protokol OSPF (Open Shortest Path First)



Gambar 2. 14 Routing Protokol OSPF (Open Shortest Path First)

OSPF (Open Shortest Path First) merupakan sebuah routing protokol berjenis *IGP (Interior Gateway routing Potokol)* yang hanya dapat bekerja dalam jaringan internal suatu organisasi atau perusahaan. Jaringan internal maksudnya adalah jaringan dimana kita masih memeliki hak untuk menggunakan, mengatur dan memodifikasinya. Singkatnya, kita masih memiliki hak administrasi terhadap

jaringan tersebut. Selain itu, OSPF merupakan routing protokol berstandar terbuka (Ainy, 2017). Ada beberapa hal penting mengenai OSPF, diantaranya :

1. Link State Protocol

   OPSF adalah protokol routing berbasis state link, yang berarti setiap router dalam area OSPF mengumpulkan informasi tentang jaringan dan topologi yang disebarkan oleh router lain. Informasi ini digunakan untuk membangun database topo logi jaringan.
2. Perhitungan Rute Terpendek

   OSPF menggunakan algoritma djikstra untuk menghitung rute terpendek (shortest path) antara router dalam jaringan. Ini memungkinkan OSPF untuk menemukan rute yang paling efesien antara dua titik dalam jaringan.
3. Hierarchical Design

   OSPF dibagi menjadi area – area yang membentuk hierarki. Ini memungkinkan OSPF untuk skalabel dalam jaringan yang besar, dengan mengurangi overhead dalam pertukaran dalam informasi topologi.
4. Konvergensi Cepat

   OSPF dirancang untuk memiliki waktu konvergensi yang cepat, yang berarti bahwa jika terjadi perubahan dalam topologi jaringan, OSPF akan secara cepat menyesuaikan rute – rute jaringan untuk menghindari paket yang hilang atau terlewat.
5. Authentication dan Security

   OSPF mendukung berbagai mekanisme otentikasi untuk memastikan keamanan dalam pertukaran informasi routing antara router, termasuk plaintext, MD5, dan metode otentikasi kunci public.

OSPF banyak digunakan dalam jaringan besar, terutama dalam jaringan korporat, ISP (*Internet Service Provider*) dan organisasi yang memiliki kebutuhan routing yang kompleks. Dengan keandalan, kecepatan konvergensi dan kemampuan skalabilitasnya, OSPF menjadi salah satu protokol routing yang paling umum digunakan dalam pengaturan jaringan yang luas.

2.12 Quagga

Quagga adalah sebuah suite software routing yang digunakan untuk mengimplementasikan protokol routing pada sistem operasi berbasis Unix, seperti Linux dan BSD. Quagga merupakan fork dari proyek Zebra yang sudah tidak aktif, dan menawarkan fungsionalitas routing yang canggih untuk jaringan skala besar maupun kecil. Adapun fitur dari quagga :

1. **Implementasi Protokol Routing:** Quagga mendukung beberapa protokol routing standar industri, termasuk :
   - OSPF (Open Shortest Path First) versi 2 dan 3
   - RIP (Routing Information Protocol) versi 1 dan 2
   - BGP (Border Gateway Protocol) versi 4
   - IS-IS (Intermediate System to Intermediate System)
2. **Arsitektur Modular:** Quagga memiliki arsitektur modular di mana setiap protokol routing diimplementasikan sebagai daemon yang terpisah. Ini memungkinkan fleksibilitas dalam mengonfigurasi dan mengelola protokol routing sesuai kebutuhan jaringan.
3. **CLI (Command Line Interface):** Quagga menyediakan antarmuka baris perintah (CLI) yang mirip dengan perintah-perintah Cisco IOS, memudahkan administrasi dan konfigurasi bagi mereka yang sudah familiar dengan perangkat Cisco.
4. **Integrasi dengan Kernel:** Quagga berfungsi sebagai pengguna ruang (user space) daemon yang mengelola tabel routing kernel, memungkinkan pembaruan dinamis tabel routing berdasarkan informasi yang diterima dari protokol routing.

Quagga memiliki berbagai kegunaan dalam manajemen jaringan dan implementasi protokol routing, menjadikannya alat yang sangat berguna dalam berbagai skenario jaringan. Bebetapa kegunaan utama dari Quagga adalah sebagai berikut :

1. Routing Dinamis di Jaringan Enterprise

   Quagga memungkinkan administrator jaringan untuk mengimplementasikan routing dinamis menggunakan berbagai protokol routing seperti OSPF, BGP, dan RIP. Ini sangat berguna dalam jaringan

perusahaan yang kompleks di mana rute optimal mungkin perlu diperbarui secara dinamis untuk mengakomodasi perubahan jaringan, kegagalan perangkat, atau optimasi jalur.

2. Interkoneksi Jaringan yang Berbeda

   Dalam lingkungan yang menggabungkan beberapa jaringan atau subnet yang berbeda, Quagga dapat digunakan untuk menghubungkan dan mengelola rute antar jaringan tersebut. Protokol seperti OSPF dan BGP membantu memastikan bahwa data dapat mengalir dengan lancar antara jaringan yang berbeda, bahkan jika mereka menggunakan skema alamat yang berbeda.

3. Penggunaan dalam Penelitian dan Pengembangan

   Quagga sering digunakan dalam penelitian jaringan dan pengembangan teknologi baru. Karena sifatnya yang open-source dan fleksibel, Quagga memungkinkan peneliti untuk menguji dan mengembangkan algoritma routing baru, mempelajari perilaku protokol routing, atau mensimulasikan skenario jaringan yang kompleks.

4. Routing di Jaringan ISP

   ISP (Internet Service Providers) menggunakan Quagga untuk mengelola routing BGP, yang merupakan protokol routing utama yang digunakan untuk pertukaran rute antar AS (*Autonomous Systems*) di internet. Quagga membantu ISP mengelola tabel routing besar, mengoptimalkan jalur, dan menjaga keandalan serta efisiensi jaringan mereka.

5. Manajemen Routing di Datacenter

   Data center yang besar dan kompleks sering kali memerlukan routing yang canggih untuk memastikan ketersediaan dan efisiensi layanan. Quagga dapat digunakan untuk mengelola routing internal dalam data center, memastikan bahwa lalu lintas data dapat mengalir secara optimal antara server, storage, dan perangkat jaringan lainnya.

6. Penggantian Perangkat Keras Routing Mahal

   Dengan menggunakan Quagga, perusahaan dan organisasi dapat menghemat biaya dengan mengurangi ketergantungan pada perangkat keras routing yang mahal. Quagga memungkinkan penggunaan server standar

untuk menjalankan fungsi routing, yang dapat mengurangi biaya operasional dan meningkatkan fleksibilitas jaringan.

7. Manajemen dan Monitoring Jaringan

   Quagga menyediakan alat dan antarmuka untuk memonitor dan mengelola tabel routing dan status protokol routing. Ini termasuk antarmuka CLI (*Command Line Interface*) yang mirip dengan Cisco IOS, yang memudahkan administrator jaringan untuk melakukan konfigurasi dan pemantauan.

8. Virtualisasi dan Jaringan Definisi Perangkat Lunak (SDN)

   Dalam lingkungan virtualisasi dan SDN, Quagga dapat digunakan untuk mengimplementasikan routing yang fleksibel dan dinamis. Dengan integrasi dengan platform virtualisasi dan alat SDN, Quagga membantu dalam membangun jaringan yang dapat dengan cepat beradaptasi dengan perubahan kebutuhan aplikasi dan beban kerja.

2.13 Zebra

Zebra adalah perangkat lunak routing open-source yang penting dalam sejarah pengembangan perangkat lunak routing dinamis. Meskipun pengembangannya telah dilanjutkan oleh Quagga dan FRRouting (FRR), konsep dan arsitektur dasar Zebra tetap menjadi fondasi dari banyak solusi routing modern. Dengan dukungan untuk berbagai protokol routing dan arsitektur yang fleksibel, Zebra (dan penerusnya) terus menjadi alat yang berguna untuk manajemen jaringan dinamis.