

BAB II

DASAR TEORI

2.1. Kajian Pustaka

Penelitian[8] yang dilakukan pada tahun 2020, membuat rancangan untuk menghubungkan 2 AS yang berbeda melalui jaringan SDN menggunakan aplikasi SDN - IP dari kontroler ONOS. Pengukuran QoS dilakukan menggunakan 2 layanan yang berbeda yaitu, layanan VoIP dan FTP. Hasil yang didapatkan untuk layanan VoIP menunjukkan nilai *throughput* sebesar 6,367 MB/sec, *delay* sebesar 0,15 ms dan *jitter* sebesar 0,01 ms. QoS layanan FTP mendapatkan nilai *throughput* sebesar 32 KB/sec, *delay* sebesar 6,66 ms, *jitter* sebesar 1,14 ms. Kesimpulan penelitian ini menunjukkan nilai QoS yang didapatkan masih berada pada nilai yang menjadi standar ITU-T G.1010.

Pada penelitian[9], menganalisis performansi *high availability* (HA) pada jaringan SDN – IP dengan kontroler ONOS menggunakan topologi 2 – *D Mesh* dan *Full Mesh*. Tujuan dari penelitian ini adalah untuk mengetahui pengaruh dari pemilihan topologi pada jaringan inti terhadap performansi HA sebuah jaringan. Parameter yang diuji pada penelitian ini antara lain pengukuran *Failover Delay*, perbandingan *Total Link*, *Failover Overhead Size* dan perbandingan *Link* terhadap *flow*. Hasil dari pengujian menunjukkan bahwa pemilihan topologi jaringan berpengaruh terhadap performansi HA jaringan. Topologi 2 – *D Mesh* memiliki nilai nilai *Failover Delay* yang stabil sebesar 140 ms dibandingkan pada topolgi *Full Mesh* yang mendapatkan angka lebih dari 200 ms. Nilai *Overhead Size* pada topologi 2 – *D Mesh* juga lebih rendah 21% dari pada topologi *Full Mesh*. Sehingga dapat ditarik kesimpulan bahwa penggunaan topologi 2 – *D Mesh* memberikan performansi HA yang lebih baik dari pada menggunakan topologi *Full Mesh* pada jaringan SDN – IP dengan kontroler ONOS.

Penelitian[10] dilakukan penerapan **routing** *Border Gateway Protocol* (BGP) menggunakan mekanisme routing external BGP berbasis *RouteFlow* pada jaringan SDN. Kontroler yang digunakan pada penelitian

ini adalah POX. Hasil penelitian mendapatkan nilai *throughput* sebesar 1049 Mbps, *delay* 69,03 ms, *jitter* 0,315 ms, *packet loss* 0% dan nilai *convergence time* sebesar 163,441 s. Kesimpulan yang didapat menunjukkan bahwa nilai QoS yang didapatkan masih berada pada nilai yang menjadi standar ITU-T G.1010.

Tabel 2.1 Rangkuman Keterkaitan dengan Penelitian Sebelumnya.

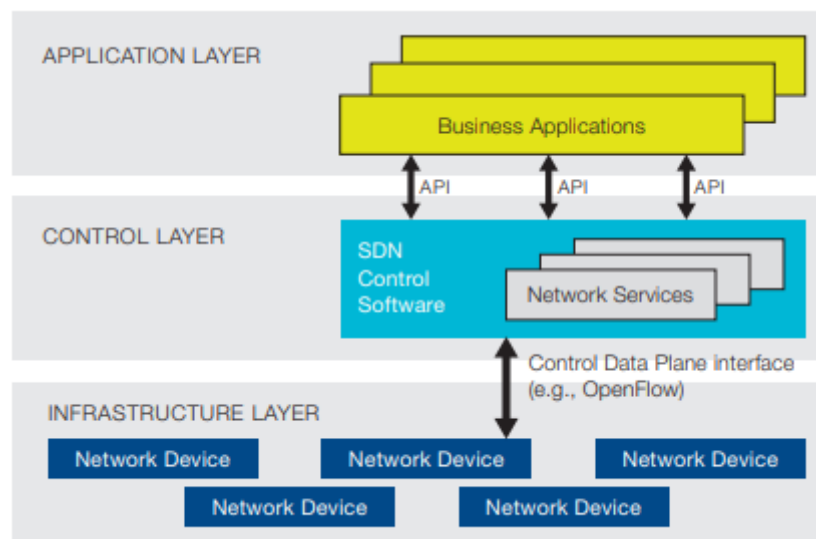
Peneliti	Komponen Penelitian				
	Topologi	Kontroler	Layanan	Parameter Pengujian	Hasil
Muhammad Nurul Yaqin, Rohmat Tulloh, S.T., M.T., Dr. Indarini Dyah Irawati, S.T.,M.T.	Ring	ONOS	SDN - IP	Pengukuran <i>Quality of Services</i> (QoS): <i>Throughput, Delay, Jitter</i> dengan layanan VoIP dan FTP.	Nilai QoS file FTP dan VoIP masih masuk dalam kategori baik menurut standarisasi ITU.T G.1010
Okatapani Panca Jaya, Ridha Muldina Negara, Danu Dwi Sanjoyo	<i>Full & Partial Mesh</i>	ONOS	SDN - IP	<i>High Availability</i> (HA): <i>Fail-over Delay, Fail-over Overhead</i>	Topologi 2D Mesh menunjukkan hasil lebih baik dibandingkan menggunakan topologi Full-Mesh
Henky Agie Friwansya,	<i>Partial Mesh</i>	POX	RouteFlow	Pengukuran <i>Quality of</i>	Nilai QoS file FTP dan

Indrarini Dyah Irawati, ST., MT., Yuli Sun Hariyani, ST.,MT.				<i>Services</i> (QoS): <i>Throughput,</i> <i>Delay,</i> <i>Jitter,</i> <i>Packet loss,</i> <i>Convergence time.</i>	VoIP masih masuk dalam kategori baik menurut standarisasi ITU.T G.1010
Donny Arief Oktavian	Tree	ONOS	SDN – IP Reactive Routing	Pengukuran <i>Quality of Services</i> (QoS): <i>Throughput,</i> <i>Delay,</i> <i>Jitter.</i>	Nilai QoS file FTP dan VoIP masih masuk dalam kategori baik menurut standarisasi ITU.T G.1010. Hasil skenario 2 menunjukkan nilai lebih baik dibanding skenario 1

2.2. Dasar Teori

2.2.1. *Software Defined Networking*

Dikutip dari *white paper* yang dicetak oleh *Open Network Foundation* yang berjudul *Software-Defined Networking: The New Norm for Networks*[11]. Definisi dari *software defined networking* adalah pemisahan fisik antara *control plane* dan *data plane*, dan dimana *control plane* itu mengontrol beberapa perangkat. SDN ini adalah arsitektur jaringan yang bersifat dinamis, dapat dikelola, hemat, dan mudah beradaptasi menjadikannya ideal untuk jaringan dengan *bandwidth* tinggi. Arsitektur ini memungkinkan kontrol jaringan dapat diprogram melalui *software* dan arsitektur yang mendasarinya menjadi aplikasi dan layanan *network*. Openflow merupakan protokol yang mendasari pembangunan SDN ini.



Gambar 2.1 Arsitektur SDN

Arsitektur SDN memiliki beberapa sifat di antaranya :

1. Pengelolaan dari berbagai vendor yang tersentralisasi. Dengan SDN dapat mengendalikan berbagai perangkat *network* seperti *switch*, *routers* dan *virtual switch*. SDN dapat digunakan untuk melakukan manajemen konfigurasi, pembangunan, pembaharuan perangkat secara cepat di seluruh jaringan.

2. Mengurangi kompleksitas melalui *Automation*. SDN berbasis *openflow* menawarkan *network automation* dan manajemen *network* yang fleksibel. *Automation* ini akan mengurangi lamanya waktu konfigurasi, kesalahan konfigurasi yang dilakukan oleh operator.
3. Mempercepat inovasi bisnis. Adopsi SDN memungkinkan operator IT untuk memprogram ulang jaringan secara *real time* untuk memenuhi kebutuhan bisnis dan pengguna. Dengan virtualisasi dari layanan jaringan operator IT dapat membuat layanan dan kapabilitas jaringan baru dalam hitungan jam.
4. Meningkatkan kehandalan dan keamanan jaringan. SDN membuat IT dapat mendefinisikan konfigurasi dan kebijakan tingkat tinggi yang kemudian diterjemahkan ke infrastruktur jaringan melalui *openflow*. Arsitektur SDN berbasis *openflow* membuat konfigurasi perangkat jaringan dapat dilakukan secara otomatis. Sehingga mengurangi kegagalan konfigurasi dari perubahan layanan atau kebijakan.
5. Pengalaman pengguna yang lebih baik. Dengan kendali jaringan yang tersentralisasi SDN dapat lebih beradaptasi dengan kebutuhan pengguna yang dinamis. Misalkan operator memperkenalkan layanan *video* yang menawarkan kualitas resolusi tinggi kepada pelanggan premium. Pelanggan harus mengatur sendiri pemilihan resolusi yang tersedia, hal itu mengakibatkan pengalaman pengguna menurun. Pada jaringan SDN yang berbasis *Openflow*, aplikasi *video* akan mendeteksi badwidth yang tersedia di jaringan secara *real time* dan secara otomatis menyesuaikan resolusi *video* yang sesuai.

2.2.2. ONOS

ONOS merupakan sistem operasi jaringan *open source* yang fokus dengan teknologi SDN dan berorientasi kepada operator (*carrier – grade*

service provider)[12]. ONOS dibuat oleh ON.Lab (ONF) dengan kerja sama lintas sektor, mulai dari operator, vendor, hingga universitas. ONOS versi pertama dirilis pada tanggal 5 Desember 2014. Rilis versi baru setiap 3 bulan, saat ini bulan pada tahun 2020 ONOS sudah mencapai versi 2.5.0[13].

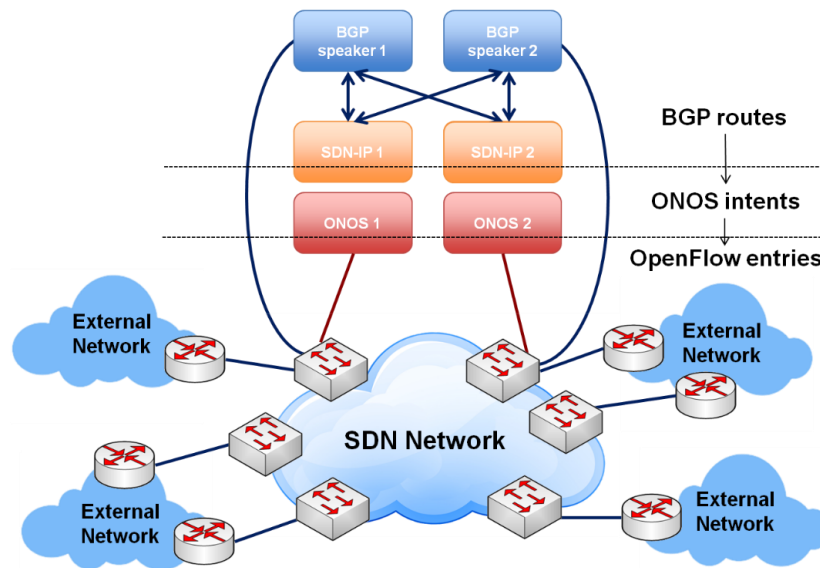


Gambar 2.2 ONOS Controller

ONOS dirancang untuk memenuhi kebutuhan operator, dengan menawarkan fleksibilitas untuk membuat dan menggunakan layanan jaringan dinamis baru dengan menggunakan *interface* terprogram yang disederhanakan[1]. ONOS mendukung konfigurasi dan pengaturan jaringan secara real time, sehingga tidak perlu menjalankan pengaturan *routing* dan *switching* di dalam struktur jaringan. dengan memindahkan kecerdasan ke ONOS *controller cloud* dimungkinkan adanya berbagai inovasi dan *user* dapat dengan mudah membuat aplikasi jaringan tanpa perlu mengubah sistem yang ada di *data plane*.

2.2.3. SDN-IP

SDN-IP adalah aplikasi dari *controller* ONOS yang memungkinkan jaringan SDN terhubung dengan jaringan eksternal di internet menggunakan *Border Gateway Protocol* (BGP)[14].



Gambar 2.3 Arsitektur SDN-IP

Seperti pada topologi di atas, jaringan terdiri dari beberapa komponen seperti ONOS, router BGP *Speaker*, Open vSwitch sebagai *data plane*, dan ruter BGP eksternal. ONOS *controller* terhubung langsung dengan open vswitch dan ruter BGP *speaker* melalui iBGP. Ruter BGP *Speaker* terhubung dengan salah satu open vswitch untuk menerima BGP *routes* dari ruter eksternal yang kemudian di rubah menjadi *intents request* oleh SDN-IP ke kontroler ONOS[16].

SDN – IP akan bertindak seperti sebuah *Autonomous System (AS)* yang meneruskan trafik dari ruter eksternal BGP yang ada. Rute yang diteruskan oleh router eksternal BGP diterima oleh BGP *speaker* dalam jaringan SDN – IP selanjutnya diproses dan akhirnya diteruskan kembali ke jaringan eksternal[17]. Rute diproses sesuai dengan kebijakan yang ditetapkan oleh perutean BGP normal. Rute terbaik ditentukan oleh aplikasi SDN – IP berdasarkan aturan iBGP, kemudian diterjemahkan ke dalam *intents request* dan dikirimkan ke dalam ONOS. ONOS menerjemahkan *intents request* ke dalam aturan *forwarding* ke *data plane*. Aturan – aturan tersebut digunakan untuk meneruskan trafik transit antar jaringan IP yang saling terhubung.

2.2.4. SDN – IP Reactive Routing

SDN – IP *Reactive Routing* merupakan aplikasi kontroler ONOS yang memungkinkan jaringan SDN dapat berkomunikasi dengan jaringan IP melalui protokol *Border Gateway Protocol* (BGP). Pada saat ONOS menerima paket IP masuk, ONOS akan secara langsung menghitung dan meng-*install* jalur peruteaan untuk trafik paket. Fungsi ini disebut dengan *Reactive Routing*. ONOS mengimplementasikan fungsi tersebut pada aplikasi *onos-app-reactive-routing*[18].

Pada jaringan SDN untuk menghubungkan jaringan menggunakan hanya *switch*, sehingga tidak terdapat *Gateway* Fisik pada jaringan SDN. Tanpa adanya *gateway* akan menjadi masalah bagi *hosts* yang ada di dalam jaringan SDN. Pada saat *host* ingin berkomunikasi dengan jaringan lain, *host* tersebut tidak dapat mengetahui alamat *next hop* yang akan dilewati paket. Untuk mengatasi masalah ini, ONOS akan membuat *Virtual Gateway* bagi *hosts* yang ada di dalam jaringan SDN.

2.2.5. GNS3

GNS3 populer digunakan oleh *network engineers* diseluruh dunia sebagai *emulator*, mengkonfigurasi, menguji, memecahkan masalah jaringan *virtual* dan nyata. GNS3 memungkinkan pengguna untuk menjalankan topologi mulai dari topologi sederhana yang terdiri dari beberapa perangkat di laptop, sampai topologi rumit yang terdiri dari banyak perangkat di beberapa *server* atau bahkan di *cloud*.

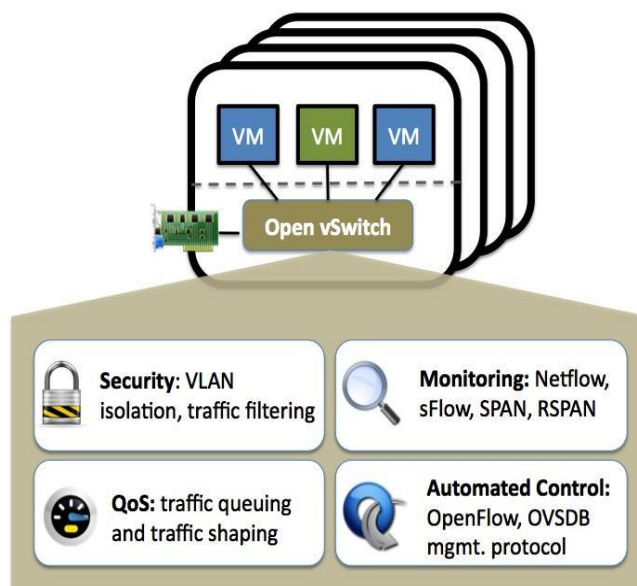


Gambar 2.4 GNS3 Emulator

GNS3 adalah *emulator* jaringan *open source* yang dikembangkan oleh komunitas. *Network engineer* memungkinkan untuk memvirtualisasi perangkat keras nyata. Awalnya GNS3 hanya dapat menggunakan perangkat Cisco menggunakan *software* yang disebut *Dynamips*. GNS3 kini telah mendukung banyak perangkat dari beberapa vendor jaringan seperti Cisco, ASA Cisco, Brocade vRouters, switch Cumulus Linux, *instance* Docker, HPE VSRs, dan beberapa tools linux lainnya[19]. Pada Tugas Akhir ini GNS3 di gunakan sebagai *emulator* untuk menjalankan topologi penelitian.

2.2.6. Open vSwitch

Open vSwitch adalah *multilayer virtual switch* yang dibuat di bawah lisensi *open source* Apache 2.0. Open vSwitch dirancang untuk membuat otomatisasi jaringan besar – besaran melalui ekstensi terprogram, sambil memiliki dukungan antarmuka dan protokol manajemen standar seperti Netflow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag[20]. selain itu, open vswitch juga dirancang untuk kebutuhan distribusi di beberapa *server* fisik mirip seperti *vswitch* pada VMWare dan Cisco's Nexus 1000V. Pada penelitian ini Open vSwitch digunakan untuk melakukan *forwarding packet* pada *data plane*.



Gambar 2.5 Open vSwitch

2.2.7. FRRouting

OpenRefractory melaporkan bahwa dia telah melakukan analisis proyek *open source* yang sedang populer yaitu FRRouting. FRRouting merupakan anggota dari Linux *Foundation*. FRRouting sendiri adalah rangkaian IP *routing* protokol untuk *platform* Linux dan Unix yang memiliki *protocol daemons* seperti BGP, IS – IS, LDP, OSPF, PIM dan RIP[22]. Pada penelitian ini FRRouting digunakan untuk men-*generate* paket BGP dari jaringan IP tradisional.



Gambar 2.6 FRRouting logo

2.2.8. Border Gateway Protocol (BGP)

Autonomous System (AS) adalah kumpulan ruter yang *prefixes* dan *routing policies* berada di bawah kendali administratif yang diberikan[23]. AS mewakili grup yang saling terhubung terdiri dari satu atau lebih blok IP *address*. AS dibagi menjadi dua perutean dalam internet, antara lain :

1. *Interior Gateway Protocol* (IGP) : memungkinkan pertukaran informasi di dalam sebuah AS. Contoh protokol routing ini seperti RIP, OSPF, EIGRP.
2. *Eksterior Gateway Protocol* (EGP) : memungkinkan pertukaran informasi antar AS. Contoh protokol ruting ini adalah BGP.

BGP adalah protokol ruting *Eksterior Gateway Protocol* (EGP) yang memungkinkan pertukaran informasi antar *Autonomous System* (AS). Ruting BGP adalah ruting protokol “*Path Vector*”. Dalam menentukan rute

– rute terbaiknya melihat pada *path* terbaik yang didapat dari router BGP lainnya. Tabel *routing* pada BGP terletak terpisah berdasarkan path AS terpendek dan atribut yang lain yaitu jarak dan biaya.

Agar BGP dapat berfungsi, ruter BGP *Speaker* harus membuat hubungan (*adjency*) dengan tetangganya. Terdapat dua jenis hubungan BGP *neighbor* yaitu iBGP dan eBGP. Pada iBGP memiliki BGP *neighbors* yang terletak dalam *Autonomous System (AS)* yang sama, sedangkan pada eBGP memiliki BGP *neighbors* yang menghubungkan *Autonomous System* yang terpisah[24].

2.2.9. Throughput

Parameter pengukuran dalam jaringan salah satunya adalah *throughput*. Pengukuran parameter *throughput* dilakukan untuk melihat besaran nilai *bandwidth* sebenarnya pada saat terjadi pengiriman data oleh komputer. Nilai *bandwidth* tidak sama dengan nilai *throughput* yang diterima oleh masing – masing komputer. Besaran dari nilai *throughput* akan bergantung pada jumlah komputer yang terhubung pada jaringan atau besaran trafik data yang mengalir di jaringan[25].

2.2.10. Delay

Delay adalah waktu yang dibutuhkan data agar sampai dari asal ke tujuannya. *Delay* dapat dipengaruhi oleh beberapa hal seperti jarak, media, dll[26]. Menurut versi ITU.T G.1010 kategori nilai *delay* dapat dilihat pada tabel 2.2.1.

Persamaan perhitungan *delay* :

$$Delay\ rata - rata = \frac{Total\ Latency}{Paket\ yang\ diterima}$$

Tabel 2.1 Standarisasi nilai latency dari ITU.T G.1010 [26]

Kategori <i>Delay</i>	Besar <i>Delay</i>	Indeks
Sangat Bagus	< 150 ms	1
Bagus	150 s/d 300ms	2
Sedang	300 s/d 450 ms	3
Jelek	> 450 ms	4

2.2.11. Jitter

Parameter *Jitter* berhubungan dengan parameter delay. *Jitter* adalah interval waktu antar kedatangan paket. Pengukuran nilai *jitter* dilakukan dengan menggunakan pendekatan rata – rata. Nilai *jitter* akan diukur menggunakan *software Wireshark*[27].

Tabel 2.2 Standarisasi nilai jitter dari ITU.T G.1010 [26]

Jitter	Category	Jitter
Standar	<i>Good</i>	0 – 20 ms
	<i>Medium</i>	20 – 50 ms
	<i>Poor</i>	> 50 ms