

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian [10], merencanakan penggunaan aplikasi SDN-IP dengan ONOS *controller* untuk menyambungkan dua *Autonomous System* (AS) yang berbeda melalui jaringan SDN. Kualitas layanan (QoS) diuji pada dua layanan, yaitu VoIP dan FTP. Hasil pengujian menunjukkan bahwa untuk layanan VoIP, *throughput* mencapai 6,367 MB/s, *delay* 0,15 ms, dan *jitter* 0,01 ms. Sedangkan untuk layanan FTP, *throughput* tercatat sebesar 32 KB/s, *delay* 6,66 ms, dan *jitter* 1,14 ms. Kesimpulan dari penelitian ini menunjukkan bahwa nilai QoS sesuai dengan standar yang ditetapkan oleh ITU-T G.1010.

Pada penelitian [11], melakukan analisis terhadap performa *high availability* (HA) pada jaringan SDN-IP yang menggunakan ONOS *controller* dengan topologi jaringan *full mesh* dan *2-D mesh*. Tujuan penelitian ini adalah untuk menentukan dampak pilihan topologi jaringan inti terhadap kinerja HA jaringan. Hasil penelitian menunjukkan bahwa pilihan topologi jaringan berpengaruh signifikan terhadap kinerja HA. Parameter yang diuji meliputi *failover delay*, perbandingan total *link*, *failover overhead size*, dan perbandingan *link* terhadap *flow*. Hasil pengujian menunjukkan bahwa topologi *2-D mesh* mempunyai nilai *Failover Delay* yang stabil sebesar 140 ms, sedangkan topologi *full mesh* mempunyai nilai lebih dari 200 ms. Selain itu, nilai *overhead size* pada topologi *2-D mesh* lebih rendah 21% dibandingkan dengan topologi *full mesh*. Dengan demikian, pada jaringan SDN-IP yang menggunakan ONOS *controller*, topologi *2-D mesh* memberikan performa HA yang lebih baik dibandingkan dengan topologi *full mesh*.

Penelitian [12], melakukan analisis perbandingan kinerja EBGp berdasarkan parameter QoS seperti *delay*, *jitter*, *throughput*, dan *packet loss* pada arsitektur jaringan konvensional dan jaringan SDN menggunakan ONOS *controller*. Hasil pengujian dengan beban trafik sebesar 7,5 MB, 10 MB, dan 12,5 MB menunjukkan bahwa jaringan SDN lebih cepat dan stabil dibandingkan dengan jaringan konvensional. Data dari pengujian kedua jaringan tersebut mendukung temuan ini. Penelitian ini memperlihatkan keunggulan jaringan SDN dalam

mengelola beban trafik yang lebih tinggi. Jaringan SDN menunjukkan performa yang sangat baik dalam hal kecepatan dan stabilitas.

Pada penelitian [13], memanfaatkan keunggulan SDN dalam mengatasi masalah *streaming* multimedia dan meningkatkan kualitas layanan yang disediakan. Hasil penelitian diuji dengan membandingkan hasil simulasi nilai *throughput* yang dikonsumsi dan rata-rata *delay* yang dihasilkan dengan standar ITU G.1010. Hasilnya menunjukkan bahwa penerapan *streaming* multimedia dengan SDN menghasilkan rata-rata *delay* kurang dari 150 ms, yang termasuk dalam kategori *excellent*. Selain itu, penerapan QoS pada jaringan juga menghasilkan peningkatan trafik. Penelitian ini menegaskan bahwa SDN dapat secara signifikan meningkatkan performa dan kualitas layanan *streaming* multimedia.

Tabel 2. 1 Literature Review

Peneliti	Perangkat Penelitian				
	Topologi	Kontroler	Layanan	Parameter Pengujian	Hasil
Muhammad Nurul Yaqin, Rohmat Tulloh, Indarini Dyah Irawati,	<i>Ring</i>	ONOS	SDN - IP	Pengukuran QoS dilakukan dengan mengukur <i>throughput, delay,</i> dan <i>jitter</i> pada layanan VoIP dan FTP.	Menurut standar ITU.T G.1010, kualitas kualitas <i>file</i> FTP dan VoIP masih dalam kategori baik.
Okatapani Panca Jaya, Ridha Muldina Negara, Danu Dwi Sanjoyo	<i>Full & Partial Mesh</i>	ONOS	SDN -IP	<i>High Availability (HA): Fail- over Delay, Fail-over Overhead</i>	<i>Topologi 2D Mesh</i> menunjukkan hasil lebih baik dibandingkan menggunakan topologi <i>Full-Mesh</i>
Lulu Hasna Mahdiyah, Jafaruddin Gusti Amri Ginting, Nanda Iryani	<i>Full Mesh</i>	ONOS	SDN -IP	Pengukuran dan perbandingan <i>Quality of Services (QoS): Throughput, Delay, Jitter, Packet loss</i>	Nilai QoS EBGp pada jaringan SDN lebih baik dari pada EBGp pada jaringan konvensional

Peneliti	Perangkat Penelitian				
	Topologi	Kontroler	Layanan	Parameter Pengujian	Hasil
Made Suartana, Aditya Prapanca	Tree	POX	<i>Open Flow</i>	Pengujian dilakukan dengan membandingkan hasil simulasi nilai <i>throughput</i> yang dikonsumsi dan rata-rata <i>delay</i> yang dihasilkan dibandingkan dengan standar ITU G.1010.	Penerapan <i>multimedia streaming</i> pada SDN mempunyai kualitas <i>delay</i> rata-rata kurang dari 150 ms dengan kategori <i>Excelent</i> peningkatan, <i>throughput</i> dibisakan dengan menerapkan QoS pada jaringan
Egia M Syahbana	2D <i>Mesh</i>	ONOS	SDN- IP	Pengukuran <i>Quality of Services (QoS): Throughput, Delay, Jitter, Packet loss, Convergence Time, Perbandingan Penentuan Jalur Terbaik BGP Secara Teori Dan Simulasi</i>	

2.2 DASAR TEORI

2.2.1 *Autonomous System (AS)*

Autonomous system (AS) adalah sekumpulan jaringan IP dan *router* yang dikelola oleh satu entitas administratif atau organisasi, yang menggunakan kebijakan *routing* bersama. Setiap AS mempunyai nomor unik yang disebut *autonomous system number (ASN)* yang digunakan untuk mengidentifikasi AS tersebut dalam proses *routing* BGP[14]. AS merepresentasikan *grup* jaringan yang saling tersambung yang terdiri dari satu atau lebih blok alamat IP. ASN digunakan dalam protokol BGP untuk memungkinkan pertukaran informasi *routing* antara AS yang berbeda. Dengan demikian, AS memainkan peran penting dalam memastikan

komunikasi yang efisien dan terstruktur antar jaringan di internet. AS dibagi menjadi dua jenis *routing* di internet, yaitu:

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

2.2.2 *Border Gateway Protocol* (BGP)

BGP adalah protokol *routing* EGP yang memungkinkan pertukaran informasi antar AS. Dalam *routing* BGP, keputusan *routing* didasarkan pada metrik *path vector*. Dapat dilihat pada tabel 2.1.

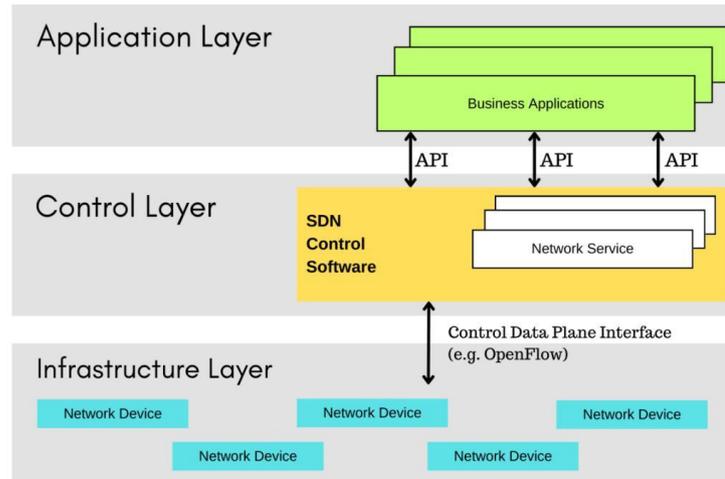
Tabel 2. 1 BGP attribute priority

1	<i>Highest Weight</i>
2	<i>Highest Local Preference</i>
3	<i>Prefer Locally Originated</i>
4	<i>Shortest AS Path</i>
5	<i>Lowest Origin Type</i>
6	<i>Lowest MED</i>
7	<i>Oldest Path</i>
8	<i>Lowest Router ID</i>

Dalam menentukan rute terbaik, BGP mempertimbangkan jalur terbaik yang diperoleh dari *router* BGP lainnya. Tabel *routing* BGP disusun berdasarkan jalur AS terpendek dan atribut lainnya. Agar BGP bisa berfungsi, *router* BGP *speaker* mesti membentuk hubungan (*adjacency*) dengan *router* tetangga. Ada dua jenis hubungan BGP *neighbors*, yaitu IBGP dan EBGP. IBGP mempunyai BGP *neighbor* yang berada dalam AS yang sama, sementara EBGP menyambungkan BGP *neighbor* yang berada dalam AS yang berbeda[15].

2.2.3 *Software Defined Network*

Menurut *white paper* yang diterbitkan oleh *open network foundation* berjudul "*Software-Defined Networking: The New Norm for Networks*," SDN didefinisikan sebagai pemisahan fisik antara *control plane* dan *data plane*. SDN adalah arsitektur jaringan yang dinamis, mudah dikelola, ekonomis, dan mudah beradaptasi, sehingga sangat cocok untuk jaringan dengan *bandwidth* tinggi.



Gambar 2. 1 Arsitektur SDN[16]

Pada gambar 2.1, arsitektur ini memungkinkan kontrol jaringan bisa diprogram melalui perangkat lunak, dan infrastruktur dasarnya mendukung aplikasi dan layanan jaringan. *OpenFlow* adalah protokol yang mendasari pengembangan SDN ini. Dengan pemisahan antara control plane dan data *plane*, administrator jaringan dapat membuat perubahan dan mengelola jaringan lebih efisien.

2.2.4 Jaringan IP Tradisional

Jaringan IP Tradisional merupakan infrastruktur komunikasi data yang memiliki logika kontrolnya sendiri, sehingga perubahan kebijakan atau konfigurasi harus diterapkan secara individual pada setiap perangkat jaringan. Pada jaringan IP tradisional fungsi dari *control plane* dan *data plane* berada berdampingan pada setiap perangkat jaringan seperti *switch* dan *router* [17]. Di antara jaringan tradisional adalah LAN, WAN, WLAN, dan *internet edge*. Jaringan-jaringan seperti ini bisa ditemukan di perangkat komputer, server, dan berbagai perangkat lain yang terhubung dengan infrastruktur kabel. Selain menggunakan kabel, biasanya traditional *network* juga menggunakan perangkat seperti *router* dan saklar untuk mengontrol jaringan. Kedua perangkat ini berguna untuk bekerja sama dan mendukung kelancaran jaringan.

2.2.5 ONOS

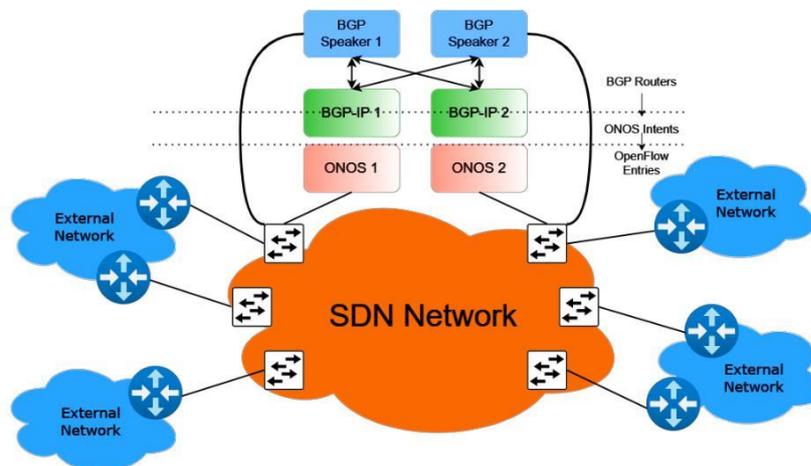
ONOS adalah sebuah *operation system* jaringan yang bersifat *opensource* yang dirancang khusus untuk mendukung teknologi SDN dan ditujukan bagi penyedia layanan dengan tingkat operator (*carrier-grade*)[18]. Pengembangan ONOS dilakukan oleh ON.Lab (ONF) bekerja sama dengan berbagai operator,

vendor, dan institusi akademik. ONOS pertama kali diperkenalkan ke publik pada tanggal 5 Desember 2014. Sejak peluncuran awalnya, ONOS terus diperbarui dengan versi baru yang dirilis setiap tiga bulan. Hingga Januari 2020, ONOS telah berkembang hingga mencapai versi 2.5.0.[19].

ONOS didesain untuk memenuhi kebutuhan para operator jaringan dengan menyediakan kemampuan untuk membentuk serta memanfaatkan layanan jaringan yang dinamis melalui antarmuka yang simpel dan terprogram. ONOS memungkinkan konfigurasi serta pengelolaan jaringan secara *real-time*, sehingga pengaturan *routing* dan *switching* manual dalam struktur jaringan tidak diperlukan. Dengan memindahkan fungsi kontrol ke ONOS *controller* berbasis *cloud*, banyak inovasi menjadi mungkin dilakukan. Hal ini memungkinkan pengguna untuk dengan mudah menyebarkan aplikasi jaringan baru tanpa perlu mengubah sistem yang sudah ada pada lapisan data *plane*. ONOS juga menyediakan solusi yang lebih fleksibel dan efisien untuk manajemen jaringan.

2.2.6 SDN-IP

SDN-IP adalah aplikasi dari ONOS *controller* yang memungkinkan jaringan SDN tersambung dengan jaringan eksternal di internet menggunakan *routing protocol* BGP[20].



Gambar 2. 2 Arsitektur SDN-IP [6]

Pada gambar 2.2, menunjukkan bahwa jaringan melibatkan berbagai perangkat, termasuk ONOS sebagai pengontrol, *router* BGP *speaker*, *Open vSwitch* sebagai data *plane*, dan *router* BGP eksternal. ONOS tersambung langsung dengan *Open vSwitch* dan *router* BGP *speaker* melalui protokol IBGP. *Router* BGP *speaker* lalu tersambung dengan salah satu *Open vSwitch* untuk menerima rute BGP dari *router*

eksternal. Rute-rute ini lalu diubah menjadi permintaan *intents* oleh aplikasi SDN-IP, yang mengirimkannya ke pengontrol ONOS. SDN-IP berfungsi sebagai Sistem Otonom (AS) yang mengarahkan lalu lintas dari *router* BGP eksternal yang ada. Rute yang diteruskan oleh *router* BGP eksternal diterima oleh BGP *speaker* di jaringan SDN-IP, diproses, dan lalu diteruskan kembali ke jaringan eksternal[21]. Aturan standar perutean BGP digunakan untuk menentukan rute terbaik. Aplikasi SDN-IP menggunakan aturan IBGP untuk menentukan rute terbaik, mengonversinya menjadi permintaan *intents*, dan lalu mengirimkannya ke ONOS. ONOS lalu menerjemahkan permintaan *intents* menjadi aturan pengalihan untuk data *plane*. Aturan-aturan ini mengatur lalu lintas transit di antara jaringan IP yang saling tersambung.

2.2.7 SDN-IP Reactive Routing

SDN-IP *reactive routing* adalah sebuah aplikasi kontroler ONOS yang memungkinkan komunikasi antara jaringan SDN dan jaringan IP menggunakan protokol BGP. Saat ONOS menerima paket IP yang masuk, aplikasi ini secara langsung menghitung dan menginstal jalur *routing* untuk lalu lintas paket tersebut. Proses ini dikenal sebagai *reactive routing*. Fungsi ini dipenerapkan dalam aplikasi ONOS yang bernama *ONOS-app-reactive-routing*[22]. Dalam jaringan SDN, koneksi antara jaringan dilakukan hanya menggunakan *switch*, sehingga tidak ada *gateway* fisik yang berfungsi di dalam jaringan SDN. Ketiadaan *gateway* fisik ini bisa menjadi masalah bagi *host* di dalam jaringan SDN. Ketika sebuah *host* ingin saling terhubung dengan jaringan lain, *host* tersebut tidak mempunyai cara untuk mengetahui alamat *next hop* yang mesti dilalui oleh paket yang dikirimnya. Untuk mengatasi masalah ini, ONOS menyediakan solusi dengan menciptakan virtual *gateway* bagi *host* di dalam jaringan SDN. Virtual *gateway* ini memungkinkan *host* untuk menentukan jalur yang benar untuk mencapai jaringan lain, meskipun tidak ada *gateway* fisik yang terlibat.

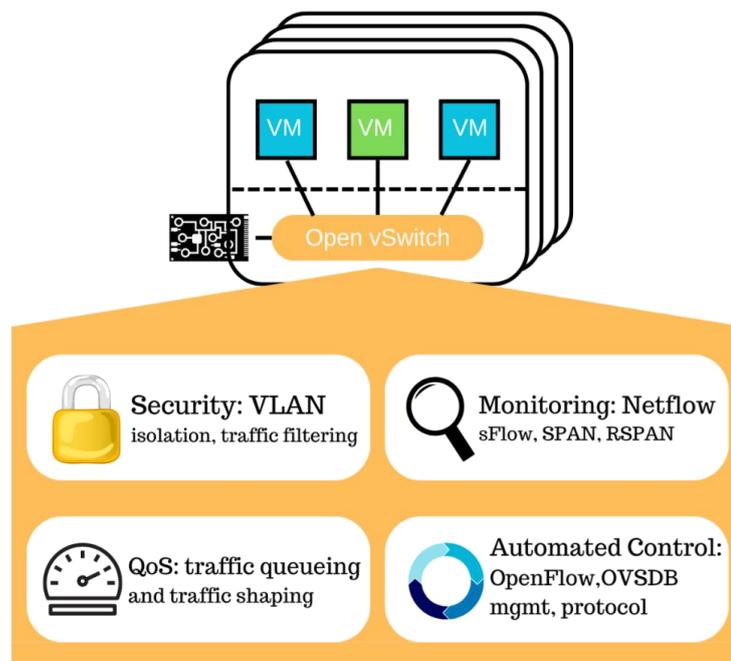
2.2.8 GNS3

GNS3 adalah alat yang sangat populer di kalangan *network engineer* di seluruh dunia, digunakan sebagai emulator untuk mengonfigurasi, menguji, dan memecahkan masalah pada jaringan, baik virtual maupun fisik. Dengan GNS3, pengguna bisa membuat berbagai topologi jaringan, mulai dari yang sederhana

dengan hanya beberapa perangkat yang dijalankan di laptop, hingga topologi yang sangat kompleks dengan banyak perangkat yang tersebar di beberapa server atau bahkan di *cloud*. GNS3 memberikan fleksibilitas bagi pengguna untuk mensimulasikan berbagai skenario jaringan, sehingga mempermudah proses belajar, pengujian, dan penyelesaian masalah jaringan secara efektif dan efisien. GNS3 adalah sebuah emulator jaringan sumber terbuka yang dikembangkan oleh komunitas, yang memungkinkan para insinyur jaringan untuk melakukan virtualisasi perangkat keras secara efektif. Awalnya, GNS3 hanya bisa mendukung perangkat dari cisco menggunakan *software dynamips*. Namun, saat ini GNS3 telah berkembang untuk mendukung berbagai perangkat dari berbagai vendor jaringan, termasuk cisco ASA, *router* virtual brocade, *switch* cumulus linux, *instance* docker, HPE VSRs, serta berbagai alat linux lainnya [23].

2.2.9 Open vSwitch

Open vswitch merupakan sebuah virtual *switch multi layer* yang dikembangkan di bawah lisensi *opensource apache 2.0*. Tujuan utama dari *open vswitch* adalah untuk memungkinkan otomatisasi jaringan skala besar dengan menggunakan ekstensi terprogram. Pada gambar 2.3, *open vswitch* juga mendukung berbagai antarmuka dan protokol manajemen standar seperti Netflow, sFlow, IPFIX, RSPAN, CLI, LACP, dan 802.1ag.[24].

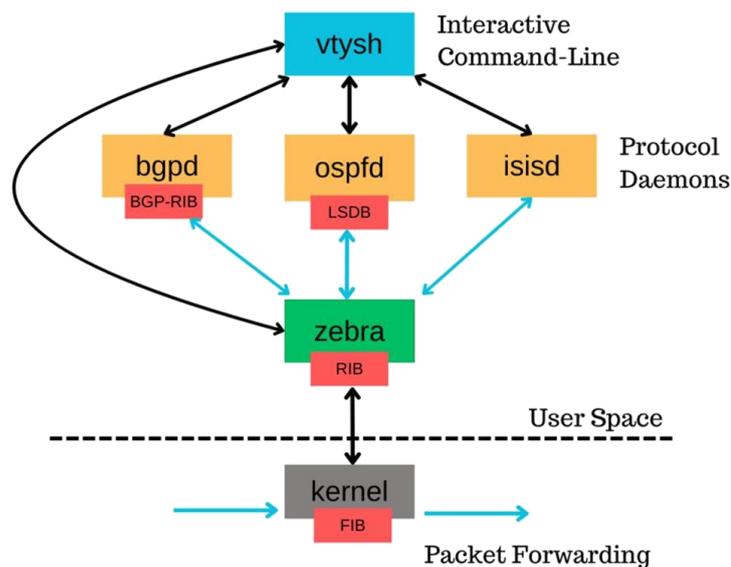


Gambar 2. 3 Open vSwitch [25]

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*.

2.2.10 Quagga

Quagga adalah *software* paket *routing* yang menyediakan layanan berbasis TCP/IP seperti Zebra dan mendukung RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, BGPv4, BGPv4+, protokol *routing* IS-IS dan juga protokol IPV4 dan IPV6 [16]. Selain mendukung protokol *routing*, quagga bisa mengatur alamat antarmuka dan rute statis. Quagga mempunyai metode sistem administrasi yang berbeda yang mempunyai dua mode pengguna. Dalam mode normal, pengguna hanya bisa melihat sistem dan dalam mode *enable*, pengguna bisa mengubah konfigurasi *system* [16]. Quagga mempunyai arsitektur multi proses, seperti zebra. Setiap daemon mempunyai file konfigurasi dan antarmuka terminal masing-masing.



Gambar 2. 4 Struktur Zebra [16]

Gambar 2.4 menunjukkan arsitektur quagga yang terdiri dari sekumpulan proses yang saling terhubung satu sama lain melalui IPC. Protokol *routing* jaringan seperti OSPF, RIP, IS IS masing-masing dipenerapkan dalam proses-proses seperti OSPFD, RIPD, IS-ISD [16].

2.2.11 VideoLAN Client (VLC)

VLC media *player* adalah program multimedia *player portabel*, dan memutar baik video maupun audio dalam berbagai format, seperti MPEG, DivX,

Ogg, dan lain-lain. VLC media *player* juga dapat digunakan untuk memutar DVD, VCD, maupun CD. VLC media *player* juga merupakan *opensource software* dan tersedia untuk berbagai sistem operasi. Kelebihan VLC media *player* adalah kelengkapan *codec* yang dimiliki[26]. Secara umum dalam sistem *streaming* terdapat 2 element yaitu *broadcaster* dan pengguna. *Broadcaster* adalah penyaji dari media video atau musik, sedangkan *user* adalah pengguna yang memanfaatkan *streaming* dari *broadcaster*. Untuk pemanfaatannya di dalam jaringan komputer, video *streaming* ini menggunakan metode penyampaian *content* multimedia yang berupa *streaming* yang dapat dilakukan dengan beberapa protokol yaitu RTP, RTSP, dan HTTP. Metode – metode tersebut mempunyai karakter masing – masing dan tentunya juga memiliki perbandingan kualitas pada masing – masing protokol tersebut[27].

2.2.12 Layanan Video Streaming

Video *streaming* teknik untuk mengirim data secara kontinu dan konsisten, sering kali dianggap sebagai *real-time*. Sebab utama yang membuat *streaming* bersifat *real-time* adalah ketiadaan media penyimpanan untuk menyimpan paket data, sehingga paket data disimpan dalam *buffer* dan lalu ditampilkan di layar. Setelah data dalam *buffer* ditampilkan, data tersebut dihapus dan digantikan dengan data baru[28]. Video *streaming* adalah metode yang menggunakan server *streaming* untuk mengirimkan video digital melalui jaringan data, sehingga video bisa diputar langsung tanpa mesti menunggu proses pengunduhan selesai atau menyimpan data terlebih dahulu di sisi PC klien. Sistem video *streaming* melibatkan proses encoding data video dan mengirimkannya melalui jaringan, sehingga klien bisa mengakses, melakukan *decoding*, dan menampilkan video secara *real-time*. Protokol yang digunakan dalam layanan video *streaming* serupa dengan yang digunakan dalam layanan *videophone*, namun video *streaming* tidak memungkinkan komunikasi dua arah.

Ada tiga jenis tipe video *streaming* menurut bentuk layanan yaitu[28]:

1. Video *On Demand* (VOD)

Video on Demand (VOD) memungkinkan pengguna untuk melakukan *pause*, *rewind*, *fast forward*, atau mengindeks isi multimedia dan bisa

diakses secara *online* melalui internet atau intranet. Video bisa langsung dilihat atau diunduh.

2. *Live Streaming*

Aplikasi *live streaming* bisa ditemukan dalam teknologi siaran radio dan televisi, memungkinkan pengguna menerima siaran radio dan televisi secara langsung (*live*).

3. *Real Time Streaming*

Aplikasi ini memungkinkan pengguna untuk saling terhubung dengan audio dan video secara *real-time*.

2.2.13 Throughput

Throughput adalah jumlah total paket yang berhasil tiba di tujuan selama interval waktu tertentu dibagi dengan durasi interval waktu tersebut. *Throughput* menunjukkan kemampuan sebenarnya suatu jaringan dalam mengirim data dan sering kali dikaitkan dengan *bandwidth* karena *throughput* juga bisa disebut sebagai *bandwidth* dalam keadaan nyata. Meskipun *bandwidth* adalah kapasitas maksimum suatu jaringan, *throughput* adalah ukuran seberapa banyak data yang benar-benar dapat ditransmisikan. Perbedaan ini penting karena *throughput* dipengaruhi oleh berbagai faktor seperti latensi, kehilangan paket, dan gangguan jaringan. Oleh karena itu, *throughput* memberikan gambaran lebih akurat tentang kinerja suatu jaringan[29].

Dengan persamaan perhitungan *throughput* [30]:

$$Throughput: \frac{\text{Jumlah Data Yang Diterima}}{\text{Waktu Pengiriman Data}} \quad (2.1)$$

Tabel 2. 2 Standarisasi TIPHON nilai *throughput*

Kategori <i>Throughput</i>	Besar <i>Throughput</i>	Indeks
Sangat Bagus	> 2,1 Mbps	4
Bagus	1200 Kbps-2,1 Mbps	3
Cukup Bagus	700-1200 Kbps	2
Kurang Bagus	338-700 Kbps	1
Buruk	0-338 Kbps	0

Persamaan perhitungan *throughput* dalam jaringan dihitung dengan membagi jumlah data yang ditransfer dengan waktu yang diperlukan untuk transfer tersebut. Secara matematis, dapat dilihat pada persamaan 2.1. Pengukuran

throughput biasanya dilakukan dalam bits per *second* (bps), kilobits per *second* (Kbps), megabits per *second* (Mbps), atau gigabits per *second* (Gbps). Tabel 2.2 menunjukkan kinerja jaringan yang dapat dievaluasi berdasarkan kategori *throughput* yang dihasilkan. Kualitas jaringan dapat diukur dengan melihat seberapa sering *throughput* masuk ke dalam kategori sangat bagus atau bagus. Area yang memerlukan perbaikan dapat diidentifikasi jika *throughput* sering berada dalam kategori kurang bagus atau buruk. Perbandingan antara penyedia layanan jaringan juga dapat dilakukan dengan menggunakan standarisasi TIPHON sebagai acuan. Kepuasan pengguna dapat diprediksi berdasarkan kategori *throughput* yang dialami.

2.2.14 Delay

Delay adalah waktu yang dibutuhkan data agar sampai dari asal ke tujuannya. *Delay* bisa disebabkan oleh beberapa hal seperti jarak, media, dan kepadatan jaringan. Selain itu, kecepatan perangkat keras seperti *router* dan *switch* juga berpengaruh pada *delay*. Terdapat beberapa jenis *delay*, termasuk *processing delay*, *queuing delay*, *transmission delay*, dan *propagation delay*. *Delay* dapat mempengaruhi kualitas layanan, terutama dalam komunikasi suara dan video. [31]. Menurut versi TIPHON kategori nilai *delay* bisa dilihat pada tabel 2.3

Tabel 2. 3 Standarisasi TIPHON nilai *latency*

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

Dengan persamaan perhitungan *delay* [30]:

$$Delay: \frac{\text{Total Latensi}}{\text{Total Paket Yang Diterima}} \quad (2.2)$$

Persamaan perhitungan *delay* dalam jaringan dihitung dengan membagi total latensi dengan total paket yang di terima. Secara matematis, dapat dilihat pada persamaan 2.2. Pengukuran *delay* biasanya dilakukan dalam satuan millisecond (ms). Tabel 2.3 menunjukkan kinerja jaringan yang dapat dievaluasi berdasarkan kategori *delay* yang dihasilkan. Kualitas jaringan dapat diukur dengan melihat seberapa sering *delay* masuk ke dalam kategori sangat bagus atau bagus. Area yang

memerlukan perbaikan dapat diidentifikasi jika *delay* sering berada dalam kategori sedang atau jelek. Perbandingan antara penyedia layanan jaringan juga dapat dilakukan dengan menggunakan standarisasi TIPHON sebagai acuan. Kepuasan pengguna juga dapat diprediksi berdasarkan kategori *delay* yang dialami.

2.2.15 Jitter

Parameter *jitter* berkaitan dengan parameter *delay*. *Jitter* adalah interval waktu antara kedatangan paket. Nilai *jitter* sangat disebabkan oleh perubahan beban lalu lintas dan jumlah antrean antara paket data di jaringan IP. Variasi dalam waktu kedatangan paket ini dapat mengganggu kualitas aplikasi *real-time* seperti VoIP dan video *streaming*. Ketika beban jaringan meningkat, antrian pada *router* dan *switch* menjadi lebih panjang, yang menyebabkan variasi waktu pengiriman paket. Selain itu, mekanisme *routing* dinamis juga dapat menyebabkan variasi dalam jalur yang diambil paket[32].

Tabel 2. 4 Standarisasi TIPHON nilai *jitter*

Kategori <i>Jitter</i>	Besar <i>Jitter</i>	Indeks
Sangat Bagus	0 ms	4
Bagus	0-75 ms	3
Sedang	75-125 ms	2
Jelek	125-225 ms	1

Dengan persamaan perhitungan *Jitter* [30]:

$$Jitter : \frac{\text{Total Variasi Delay}}{\text{Total Paket Yang Diterima}} \quad (2.3)$$

$$\text{Total variasi Delay: Delay - (Rata- Rata Delay)} \quad (2.4)$$

Persamaan perhitungan *jitter* dalam jaringan dihitung dengan membagi total variasi *delay* dengan total paket yang di terima. Secara matematis, dapat dilihat pada persamaan 2.3. Pengukuran *jitter* biasanya dilakukan dalam satuan millisecond (ms). Tabel 2.4 menunjukkan kinerja jaringan yang dapat dievaluasi berdasarkan kategori *jitter* yang dihasilkan. Kualitas jaringan dapat diukur dengan melihat seberapa sering *jitter* masuk ke dalam kategori sangat bagus atau bagus. Area yang memerlukan perbaikan dapat diidentifikasi jika *jitter* sering berada dalam kategori sedang atau jelek. Perbandingan antara penyedia layanan jaringan juga dapat dilakukan dengan menggunakan standarisasi TIPHON sebagai acuan. Kepuasan pengguna juga dapat diprediksi berdasarkan kategori *jitter* yang dialami

2.2.16 Packet Loss

Packet loss adalah kondisi di mana satu atau lebih paket data yang dikirim melalui jaringan tidak berhasil mencapai tujuan. Hal ini dapat disebabkan oleh kemacetan lalu lintas, kerusakan perangkat keras, atau gangguan pada jalur transmisi. Ketika paket hilang, data harus dikirim ulang, yang dapat menambah latensi dan mengurangi efisiensi jaringan. *Packet loss* sering kali mempengaruhi kualitas aplikasi *real-time* seperti VoIP dan video *streaming*, menyebabkan suara terputus-putus atau video yang tersendat. Mengukur *packet loss* melibatkan perhitungan persentase paket yang hilang dari jumlah paket yang dikirim, yang membantu dalam memantau kinerja jaringan. Untuk mengatasi *packet loss*, perlu dilakukan optimasi jaringan, peningkatan perangkat keras, dan manajemen lalu lintas yang efektif[29]. Dengan persamaan perhitungan *packet loss* [30]:

$$\text{Packet Loss} = \frac{(\text{Paket Data Dikirim} - \text{Paket Data Diterima})}{\text{Paket Yang Diterima}} \quad (2.5)$$

Tabel 2. 5 Standarisasi TIPHON nilai *packet loss*

Kategori <i>Packet Loss</i>	Besar <i>Packet Loss</i>	Indeks
Sangat Bagus	0 %	4
Bagus	3 %	3
Sedang	15 %	2
Jelek	> 25 %	1

Persamaan perhitungan *packet loss* dalam jaringan dihitung dengan mengurangi total paket yang dikirim dengan total paket yang diterima dan dibagi dengan total paket yang diterima. Secara matematis, dapat dilihat pada persamaan 2.5. Pengukuran *packet loss* biasanya dilakukan dalam satuan persentase (%). Tabel 2.5 menunjukkan kinerja jaringan yang dapat dievaluasi berdasarkan kategori *packet loss* yang dihasilkan. Kualitas jaringan dapat diukur dengan melihat seberapa sering *packet loss* masuk ke dalam kategori sangat bagus atau bagus. Area yang memerlukan perbaikan dapat diidentifikasi jika *packet loss* sering berada dalam kategori sedang atau jelek. Perbandingan antara penyedia layanan jaringan juga dapat dilakukan dengan menggunakan standarisasi TIPHON sebagai acuan. Pengalaman pengguna juga dapat diprediksi berdasarkan kategori *packet loss* yang dialami

2.2.17 Convergence Time

Convergence adalah proses di mana *router* mengambil informasi tentang keadaan jaringan yang valid untuk menemukan rute optimal sesuai dengan algoritma yang digunakan, dan memperbarui tabel *routing*. *Convergence* bisa terjadi jika ada penambahan *router* atau *link failure*, sehingga perubahan terjadi ketika setiap *router* mengoperasikan algoritma *routing* sendiri untuk menghitung metrik dan memperbarui tabel *routing* berdasarkan informasi terbaru. Waktu konvergensi mengukur durasi yang dibutuhkan dari saat perubahan topologi terjadi hingga jaringan kembali ke kondisi stabil dengan informasi *routing* yang konsisten di semua perangkat. Proses ini mempengaruhi kinerja jaringan secara langsung, karena periode ketidakstabilan dapat menyebabkan paket data dialihkan melalui jalur suboptimal atau bahkan hilang. Protokol *routing* yang berbeda memiliki kecepatan konvergensi yang berbeda, juga. Selain itu, faktor seperti ukuran jaringan, frekuensi pembaruan, dan kemampuan perangkat juga mempengaruhi waktu konvergensi[33].

Dengan persamaan perhitungan waktu konvergensi:

$$\begin{aligned} & \text{Waktu Konvergensi:} \\ & \text{Rata – rata waktu x jumlah paket yang hilang} \end{aligned} \tag{2.6}$$

Persamaan perhitungan waktu konvergensi dalam jaringan dihitung dengan mengkalikan rata-rata waktu dengan jumlah paket yang hilang dalam suatu data tertentu. Secara matematis, dapat dilihat pada persamaan 2.6. Pengukuran waktu konvergensi biasanya dilakukan dalam satuan persentase *second* (s). Kinerja jaringan dapat dievaluasi berdasarkan kecepatan waktu konvergensi yang dihasilkan. Semakin cepat waktu konvergensi pada suatu jaringan maka akan meningkatkan kualitas keandalan pada suatu jaringan.