

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Penelitian terdahulu adalah studi literatur atau pembahasan yang berkaitan dengan topik penelitian. Berdasarkan hasil dari studi literatur yang telah dilakukan dirangkum sumber – sumber penelitian sebelumnya yang digunakan untuk referensi pada penelitian ini. Berikut merupakan beberapa penelitian sebelumnya yang diangkat sebagai bahan penelitian oleh peneliti diantaranya adalah:

Penelitian dari Supriyatno, Jupriyadi, Syaiful Ahdan, Sampurna Dadi Riskiono pada tahun 2020 dengan judul Analisis Perbandingan Kinerja Protokol *Routing* Rip dan Ospf pada Topologi *Mesh* menggunakan *software* simulasi *Cisco Packet Tracer*. Hasil dari penelitian ini adalah saat jaringan tidak sibuk *routing* protokol OSPF lebih optimal dari *routing* protokol RIP, sedangkan pada saat jaringan sibuk *routing* protokol RIP lebih baik dibandingkan dengan *routing* protokol OSPF[8].

Penelitian kedua dilakukan oleh Chairul Mukmin dan Edi Surya Negara pada tahun 2019 dengan judul Analisis Kinerja Redistribusi *Routing* Protokol Dinamik (Studi Kasus: RIP, EIGRP, IS-IS) menggunakan *software* simulator *Graphical Network Simulator (GNS3)*. Penelitian tersebut menghasilkan nilai dari *throughput* RIP redistribusi EIGRP dan EIGRP redistribusi IS-IS lebih baik dari RIP redistribusi IS-IS dengan persentase nilai yaitu sama-sama 3%. Nilai *delay* RIP redistribusi IS-IS dan EIGRP Redistribusi IS-IS mendapatkan hasil pengukuran yang lebih baik daripada RIP redistribusi EIGRP dengan nilai rata-rata 59, sedangkan RIP redistribusi EIGRP mendapatkan nilai 290. Nilai *packet loss* yang dihasilkan dari penelitian tersebut memiliki kesamaan nilai yakni 0%[11].

Penelitian ketiga dilakukan oleh Sampurna Dadi Riskiono, Donaya Pasha, Muhamad Trianto pada tahun 2018 dengan judul Analisis Kinerja Metode *Routing*

OSPF dan RIP Pada Model Arsitektur Jaringan Di SMKN XYZ menggunakan *software Cisco Packet Tracer*. Hasil dari penelitian ini terdapat dua kondisi yaitu ketika kondisi jaringan kosong dan ketika jaringan sibuk. Ketika jaringan kosong nilai rata-rata di metode *routing* RIP untuk hasil *delay* sebesar 4,7 ms, untuk *packet loss* mendapatkan hasil 0% dan hasil *throughput* sebesar 163,156 Kbps, sedangkan di metode OSPF pada kondisi jaringan kosong mendapatkan nilai rata-rata untuk *delay* sebesar 5,4 ms, untuk *packet loss* mendapatkan hasil 0% dan hasil *throughput* sebesar 153,743 Kbps. Ketika kondisi jaringan sibuk pada metode *routing* RIP mendapatkan nilai rata-rata pada *delay* sebesar 16,9 ms, untuk *packet loss* mendapatkan hasil 0% dan hasil *throughput* sebesar 23,006 Kbps, sedangkan pada metode OSPF pada kondisi jaringan sibuk mendapatkan nilai rata-rata untuk *delay* sebesar 20,9 ms, untuk *packet loss* mendapatkan hasil 0% dan hasil *throughput* 20 Kbps[12].

Penelitian selanjutnya oleh Unung Verawardina pada tahun 2018 dengan judul Analisis Perbedaan *Performance* dan *Quality Of Service (QoS)* Antara Eigrp dengan Ospf (Studi Kasus Menggunakan 6 Router Melalui GNS 3 dan Wireshark) dengan menggunakan *software Graphical Network Simulator (GNS 3)* dan *Wireshark*. Hasil dari penelitian ini adalah nilai dari rata-rata *delay* pada metode *routing* protokol EIGRP sebesar 1,49 ms, sedangkan hasil dari rata-rata *delay* pada metode *routing* protokol OSPF sebesar 2,39 ms. Selanjutnya hasil dari nilai rata-rata *packet loss* pada metode EIGRP sebesar 0,66%, sedangkan hasil dari rata-rata *packet loss* OSPF sebesar 0,56%. Untuk hasil rata-rata dari *throughput* metode *routing* protokol EIGRP sebesar 58,38 bit/s, sedangkan pada metode *routing* protokol OSPF mendapatkan hasil rata-rata *throughput* sebesar 44,45 bit/s[13].

Penelitian oleh Febrianto Sabirin dan Ryan Permana pada tahun 2017 dengan judul Perbedaan *Routing* Menggunakan *Routing Information Protocol (RIP)* dengan *Open Shortest Path First (OSPF)* dengan menggunakan dua buah *software* yaitu *Graphical Network Simulator (GNS3)* dan *Enterprise Network Simulation Platform (ENSP)*. Hasil dari penelitian ini menunjukkan performa *routing* protokol OSPF lebih baik daripada RIP ketika diujikan menggunakan

software GNS3 dengan perbedaan sebesar 521 milidetik antara keduanya, sedangkan ketika diujikan dengan menggunakan software ENSP antara routing protokol RIP dan OSPF terdapat perbedaan sebesar 2,33 milidetik[14].

Tabel 2. 1 Tabel Penelitian Terdahulu

No	Penulis	Judul Penelitian	Metode	Hasil
1	Supriyatno, Jupriyadi, Syaiful Ahdan, Sampurna Dadi Riskiono[8]	Analisis Perbandingan Kinerja Protokol Routing RIP dan OSPF pada Topologi MESH	Metode Eksperimen	Pada saat jaringan tidak sibuk routing protokol OSPF lebih optimal dari routing protokol RIP, sedangkan pada saat jaringan sibuk routing protokol RIP lebih baik dibandingkan OSPF.
2	Chairul Mukmin dan Edi Surya Negara[11]	Analisis Kinerja Redistribusi Routing Protokol Dinamik (Studi Kasus : RIP, EIGRP, IS-SIS)	Metode Simulasi	Hasil dari pengujian delay RIP redistribusi IS-IS dan EIGRP redistribusi IS-IS lebih unggul dari RIP redistribusi EIGRP dengan rata-rata nilai 59, sedangkan nilai dari RIP redistribusi EIGRP adalah 290. Kemudian di pengujian packet loss memiliki hasil persentase 0%. Terakhir pada pengujian throughput RIP redistribusi EIGRP dan EIGRP redistribusi IS-IS memiliki hasil yang lebih baik dari RIP redistribusi IS-IS dengan persentase nilai masing-masing 3%.
3	Sampurna Dadi Riskiono, Donaya Pasha, Muhamad Trianto[12]	Analisis Kinerja Metode Routing OSPF dan RIP pada Model Arsitektur Jaringan di SMKN XYZ	Metode Simulasi	Hasil dari pengujian ketika jaringan kosong pada routing protokol RIP yaitu untuk delay sebesar 4,7 ms, packet loss sebesar 0%, dan throughput sebesar 163,156 Kbps, sedangkan pada routing protokol OSPF yaitu untuk delay sebesar 5,4 ms, packet loss sebesar 0%, dan

No	Penulis	Judul Penelitian	Metode	Hasil
				<i>throughput</i> sebesar 153,743 Kbps. Ketika jaringan sibuk protokol <i>routing</i> RIP mendapatkan hasil <i>delay</i> sebesar 16,9 ms, <i>packet loss</i> sebesar 0% dan <i>throughput</i> 23,006 Kbps, sedangkan pada <i>routing</i> protokol OSPF untuk hasil <i>delay</i> 20,9 ms, <i>packet loss</i> 0% dan <i>throughput</i> 20 Kbps.
4	Unung Verawardina[13]	Analisis Perbedaan <i>Performance</i> dan <i>Quality Of Service (QoS)</i> antara EIGRP dengan OSPF (Studi Kasus Menggunakan 6 Router Melalui GNS 3 dan Wireshark)	Metode Eksperimen	Hasil dari penelitian ini untuk <i>routing</i> protokol EIGRP mendapatkan hasil <i>delay</i> 1,49 ms, <i>packet loss</i> 0,66% dan <i>throughput</i> 58,38 bit/s, sedangkan pada <i>routing</i> protokol OSPF mendapatkan hasil <i>delay</i> 2,39 ms, <i>packet loss</i> 0,56%, dan <i>throughput</i> 44,45 bit/s.
5	Febrianto Sabirin dan Ryan Permana[14]	Perbedaan <i>Routing</i> Menggunakan <i>Routing Information Protocol (RIP)</i> dengan <i>Open Shortest Path First (OSPF)</i>	Metode Analisis dan Simulasi	Hasil pengujian GNS3 yaitu OSPF memiliki waktu transfer data yang lebih baik daripada RIP dengan perbedaan 521 milidetik, sedangkan menggunakan <i>software</i> ENSP keduanya memiliki perbedaan 2,33 milidetik.

## 2.2 Dasar Teori

### 2.2.1 *Routing* Protokol

*Routing* protokol ialah standarisasi aturan yang digunakan untuk memastikan perangkat router saling bertukar informasi dan berkomunikasi dengan perangkat lainnya dan juga menentukan rute tercepat yang akan dilewati[2]. *Routing* protokol terdiri dari 2 jenis yaitu *routing* protokol dinamis dan *routing* protokol statis. *Routing* protokol dinamis mempunyai keunggulan yaitu tidak perlu mengetahui semua alamat jaringan yang tersedia, apabila terdapat penambahan jaringan, tidak perlu melakukan konfigurasi pada semua router, dan hanya mengenalkan alamat jaringan yang terhubung langsung dengan routernya. Adapun keunggulan dari *routing* protokol statis adalah beban kerja *router* ringan dibandingkan dengan *routing* dinamis, pengiriman paket lebih cepat dan lebih mudah mendeteksi kesalahan pada topologi jaringan[15].

### 2.2.2 *Open Shortest Path First (OSPF)*

OSPF adalah *routing* protokol yang bisa menggunakan pengalamatan IPv4 dan IPv6 dengan menggunakan metode *link-state*. Untuk menentukan jalur pada setiap *routing*nya menggunakan algoritma *dijkstra*. OSPF juga merupakan *routing* protokol yang sering digunakan pada sebuah jaringan *Autonomus System*[6]. OSPF dapat melakukan konvergensi secara cepat dan menentukan *path* terbaik berdasarkan *cost* terendah[16]. Pada sebuah jaringan yang besar membutuhkan memori yang besar serta perhitungan waktu yang lama. Maka dari itu OSPF menggunakan metode area dan *backbone*. Pada setiap jaringan pada sebuah area terhubung ke sebuah *backbone*. Setiap area merupakan jaringan tersendiri dan beberapa router didalamnya hanya perlu memiliki topologi jaringan di area tersebut. Router yang terletak di perbatasan antar area hanya mengirimkan sebuah ringkasan dari *link* yang terdapat pada area dan tidak mengirimkan topologi area

satu dengan yang lainnya. Dengan begitu perhitungan *router* menjadi lebih sederhana[17].

Proses *update* tabel *routing* pada OSPF sangat efisien, dikarenakan pembaruan dapat terjadi apabila ada perubahan dalam suatu jaringan (bertambah atau berkurang), sehingga *routing* protokol OSPF tidak melakukan pengiriman informasi yang terdapat pada *router* ke *router* tetangga pada area tersebut. Area tersebut merupakan sekumpulan dari *router* yang mempunyai persamaan pada area *ID*[18]. OSPF melakukan pembagian pada jaringan dalam beberapa *logical area* untuk menjaga kinerja pada jaringan tersebut. Pada saat perhitungan *cost* algoritma OSPF, apabila terdapat suatu jaringan besar dan kompleks tentu akan memakan *resources* dari perangkat *router* yang digunakan, oleh karena itu dalam suatu jaringan dibagi menjadi beberapa area. OSPF mempunyai area khusus yang disebut dengan area 0 atau area *backbone*. Selain area 0 harus terhubung dengan area 0 dan tidak dapat terhubung dengan area lain secara langsung karena semua *traffic* yang berada pada jaringan tersebut melalui area 0. Dari masing-masing area kemudian dihubungkan dengan perangkat *router* yang disebut ABR (*area border routing*). Fungsi dari ABR adalah agar area yang lainnya dapat terhubung dengan area 0[19].

### **2.2.3 Routing Information Protocol (RIP)**

RIP merupakan *routing* protokol yang menerapkan metode *distance-vector* yang terdapat pada sebuah *Autonomus System*. *Routing* protokol RIP menggunakan *hop* dimana pada *routing* protokol ini jumlah maksimal untuk *hop* adalah 15[7]. Rip menggunakan algoritma *Bellman-Ford*, atau *distance vector* untuk menentukan rute terbaik yang digunakan menuju destinasi paket. Algoritma *Bellman-Ford* digunakan untuk menghitung jarak terpendek pada sebuah graf. Tindakan yang dilakukan pada perhitungan algoritma *bellman-ford* yaitu menjadikan nilai jarak dengan 0. Kemudian perhitungan diperbarui dengan

mengecek nilai yang paling kecil pada penjumlahan nilai jarak pada 2 *node*, setelah itu ulangi langkah tersebut hingga seluruh *node* terlewati[7]. RIP juga mengaplikasikan metode *triggered update* sehingga dapat mengetahui kapan *router* harus kembali memberikan informasi seputar *routing*. Apabila terdapat peralihan pada jaringan, sementara *timer* belum berakhir maka *router* akan tetap mengirimkan informasi *routing* karena disebabkan oleh perubahan tersebut. Oleh karena itu, *router* yang di jaringan dapat mengetahui secara cepat perubahan yang terjadi dan meminimalisir terjadinya *routing loop*[17].

#### **2.2.4 Graphical Network Simulator (GNS3)**

GNS3 merupakan sebuah *software* simulator jaringan dengan model grafis dan digunakan untuk melakukan simulasi pada sebuah jaringan dan dapat digabungkan ke jaringan yang nyata. GNS3 bersifat *open source* dan tersedia pada berbagai macam sistem operasi seperti *windows*, *mac os*, dan *linux*[14].

#### **2.2.5 Wireshark**

*Wireshark* merupakan *software* yang dapat melakukan dan meretas paket pada sebuah jaringan. *Wireshark* mendapatkan informasi berupa *internet protocol* yang ditangkap melalui komunikasi data dari protokol HTTP, TCP, dan paket pengiriman ICMP[20].

#### **2.2.6 Free Range Routing (FRR)**

*FRRouting* (FRR) merupakan sebuah perangkat *routing* yang bersifat *open source* serta memiliki fitur lengkap dan performa yang tinggi. FRR dapat mengaplikasikan *routing* protokol seperti BGP, EIGRP, IS-IS, OSPF, dan RIP[10]. Integrasi FRR dengan jaringan Linux/Unix asli menjadikannya sebagai perangkat *router* yang dapat diterapkan secara *universal* dan dapat digunakan di berbagai tujuan, seperti menghubungkan antar *host*, mesin virtual, dan perangkat yang terhubung ke jaringan rute dalam LAN serta akses internet[21].

### 2.2.7 *Internet Protocol* versi 4

IPv4 merupakan jenis pengalamatan jaringan yang terdapat pada protokol TCP/IP. IPv4 dapat menghubungkan koneksi internet sekitar 4 miliar host komputer di dunia dan memiliki panjang 32 bit[9]. IPv4 juga merupakan protokol tanpa koneksi, yang berarti paket tetap dikirim tanpa memeriksa apakah perangkat tujuan sudah siap. Keuntungan IPv4 adalah jika koneksi diblokir atau ada masalah dengan router, protokol dapat mengirimkan paket data melalui rute alternatif[22].

### 2.2.8 *Quality of Service (QoS)*

*Quality of service* adalah perhitungan performansi untuk meningkatkan kinerja dan efektifitas pada sebuah jaringan. Adapun kriteria yang diukur pada QoS adalah : *Delay*, *Packet loss*, *Throughput*, dan *jitter*[8]. Tujuan utama QoS adalah untuk menjamin aliran data aplikasi sampai batas tertentu, seperti ketersediaan bandwidth yang cukup, kemampuan untuk mengontrol *delay*, *jitter*, dan mengurangi kehilangan data. Pengukuran QoS dilakukan secara kuantitatif, sehingga mendapatkan karakteristik jaringan yang sesuai[23]. Terdapat empat kategori Kualitas Layanan dalam standar *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)* yang dapat digunakan untuk mengategorikan sebuah layanan. Klasifikasi ini diterapkan dalam pengaturan peering dan perjanjian pasokan di mana tarif yang berbeda bisa diterapkan dalam berbagai tingkat kualitas atau dimana jaminan kinerja dapat diberikan[24].

### 2.2.9 *Delay*

*Delay* adalah total waktu yang diperlukan paket data untuk melintasi jarak dari jaringan asalnya menuju jaringan yang akan dituju. Faktor penyebab terjadinya *delay* adalah media fisik, jarak, dan waktu pemrosesan yang lama[1]. Persamaan untuk menghitung rata-rata *delay* adalah sesuai rumus 2.1 :



$$Delay = \frac{Total\ Delay}{Total\ Paket\ yang\ diterima} \quad (2.1)$$

Tabel 2. 2 Tabel kriteria *delay* berdasarkan standar tiphon[25]

Kategori	<i>Delay</i>	Indeks
Sangat Bagus	<150 ms	4
Bagus	151 ms – 250 ms	3
Sedang	251 ms – 350 ms	2
Buruk	351 ms – 450 ms	1

### 2.2.10 *Packet loss*

*Packet loss* merupakan jumlah paket yang hilang pada proses pengiriman data dari trafik awal menuju *node* tujuan. Faktor penyebab terjadinya *packet loss* yaitu ketika paket data dalam suatu jaringan saling bertabrakan[15]. Untuk menghitung *packet loss* menggunakan rumus 2.2 :

$$PacketLoss = \frac{Paket\ data\ yang\ dikirim - Paket\ data\ yang\ diterima}{Total\ Paket\ yang\ diterima} \times 100\% \quad (2.2)$$

Tabel 2. 3 Tabel kriteria *packet loss* berdasarkan standar tiphon

Kategori	<i>Packet loss</i>	Indeks
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Buruk	25%	1

### 2.2.11 *Throughput*

*Throughput* adalah banyaknya kecepatan data yang diukur dalam satuan bps dan bisa diartikan juga sebagai jumlah total paket yang sukses diperiksa di selang waktu tertentu dibagi dengan panjang interval waktu[26]. Perhitungan *throughput* dapat dilakukan dengan rumus 2.3 :

$$\textit{Throughput} = \frac{\textit{Paket data yang dikirim(byte)}}{\textit{Waktu pengiriman}} \quad (2.3)$$

Tabel 2. 4 Tabel kriteria *throughput* berdasarkan standar tiphon

Kategori	<i>Throughput</i>	Indeks
<i>Bad</i>	0 – 338 Kbps	0
<i>Poor</i>	338 – 700 Kbps	1
<i>Fair</i>	700 – 1200 Kbps	2
<i>Good</i>	1200 – 2,1 Mbps	3
<i>Excellent</i>	>2,1 Mbps	4

### 2.2.12 Jitter

*Jitter* sebenarnya merupakan bentuk *delay* juga, namun *delay* ini memiliki variasi yang berbeda-beda dan erat kaitannya dengan *latency*. *Jitter* mencerminkan besaran *delay* yang terjadi dalam transmisi data di dalam jaringan. Jika terdapat *delay* pada antrian dalam perangkat jaringan seperti router dan switch dapat mengakibatkan terjadinya jitter[10]. Perhitungan *jitter* dapat dilakukan dengan rumus 2.4 :

$$\textit{Jitter} = \frac{\textit{Total Variasi Delay}}{\textit{Total Paket yang Diterima}} \quad (2.4)$$

Tabel 2. 5 Tabel kriteria *jitter* berdasarkan standar tiphon

Kategori	<i>Jitter</i>
Sangat Bagus	0 ms
Bagus	0 – 75 ms
Sedang	75 – 125 ms
Buruk	125 – 225 ms