

BAB IV

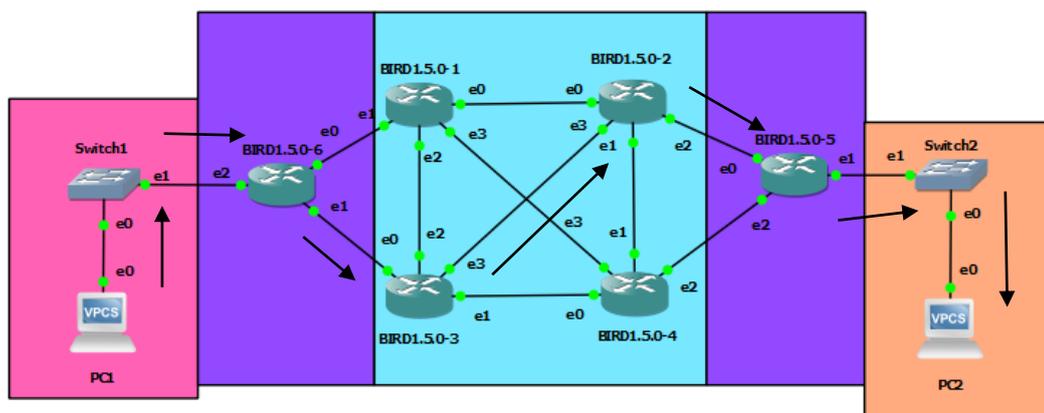
HASIL DAN PEMBAHASAN

4.1. Hasil dan Pembahasan Penelitian

Tahap ini merupakan hasil dan pembahasan dari pengujian penelitian yang telah dilakukan. Pengujian dilakukan dengan *capture* data yang dikirim dari PC1 ke PC2 dengan menggunakan *wireshark*. Data yang diambil terdapat dua protokol yaitu UDP dan TCP. Berdasarkan data yang telah berhasil diambil, kemudian data tersebut akan dihitung untuk mencari *jitter*, *delay*, serta *throughput*-nya. Ketika tahap pengambilan data selesai dilakukan kemudian dilakukan analisis guna mendapat hasil dan perbandingan dari performa kedua *routing protocol* yang diuji.

4.1.1. Alur Pengujian Jaringan OSPF

Pengujian dilakukan dengan melakukan pengiriman data dari PC1 yang merupakan *client* menuju PC2 yang merupakan *server* dengan menggunakan *iperf3*.



Gambar 4. 1 Alur pengujian jaringan OSPF

Pengujian jaringan pada *routing* OSPF dimulai dari PC1 yang merupakan *client* dan akan menuju PC2 sebagai *server*. Pengiriman dilanjutkan menuju ke *router* BIRD1.5.0-6 kemudian menuju ke *router* BIRD 1.5.0-3. Dari *router* BIRD paket dilanjutkan menuju *router* BIRD1.5.0-2, lalu diteruskan menuju *router* BIRD1.5.0-5. Dari *router* BIRD1.5.0-5 diteruskan menuju ke PC2 yang merupakan tujuan dari paket yang dikirimkan.

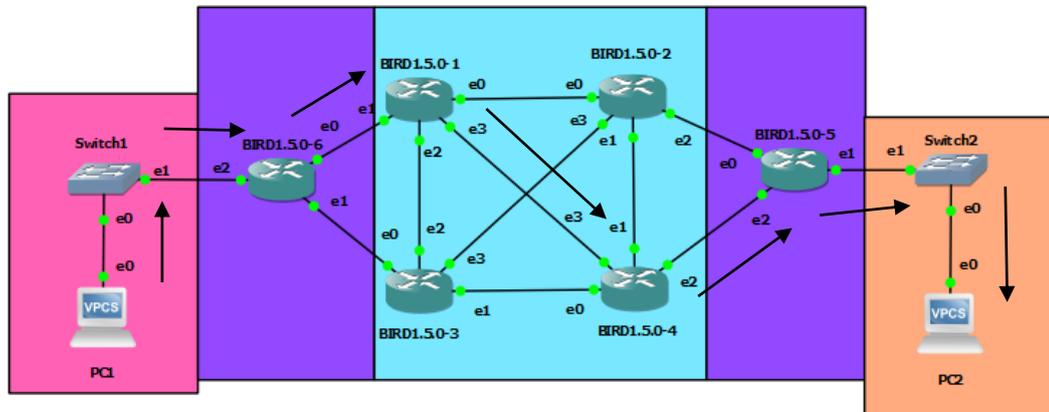
Tabel 4. 1 Alur Pengujian Jaringan OSPF

Perangkat	IP address	Tujuan
PC1	172.16.2.2/24	172.16.2.1/24
BIRD 1.5.0-6	172.16.2.1/24	192.168.3.1/24
BIRD 1.5.0-3	192.168.3.1/24	10.10.5.1/24
BIRD 1.5.0-2	10.10.5.1/24	172.16.1.1/24
BIRD 1.5.0-5	172.16.1.1/24	172.16.1.2/24

Pada tabel 4.1 merupakan jalur yang dilalui oleh paket yang dikirimkan dengan menggunakan protoko *routing* OSPF dengan menggunakan iperf dari PC1 menuju PC2. Pengujian dimulai dari PC1 yang merupakan *client* dengan ip 172.16.2.2/24 dan berakhir pada PC2 dengan ip 172.16.1.2/24.

4.1.2. Alur Pengujian Jaringan RIP

Alur pengujian pada jaringan RIP sama dengan pengujian pada pengujian jaringan OSPF yaitu dimulai dari PC1 yang merupakan *client* menuju PC2 yang merupakan *server*. Dari pengujian tersebut diketahui bahwa alur pengujian seperti terdapat pada gambar 4.2.



Gambar 4. 2 Alur pengujian jaringan RIP

Uji coba jaringan pada pengaturan routing RIP diawali dengan inisiasi dari PC1, yang berperan sebagai *client* dan bertujuan terhubung ke PC2 sebagai *server*. Proses pengiriman data berlanjut ke router BIRD1.5.0-6, kemudian dilanjutkan ke router BIRD 1.5.0-1. Dari router BIRD tersebut, paket data diarahkan ke router BIRD1.5.0-4, dan seterusnya ke router BIRD1.5.0-5. Dari router BIRD1.5.0-5, data diteruskan hingga mencapai PC2, yang merupakan tujuan akhir dari paket yang sedang dikirimkan.

Tabel 4. 2 Alur Pengujian Jaringan OSPF

Router	IP address	Tujuan
PC1	172.16.2.2/24	172.16.2.1/24
BIRD 1.5.0-6	172.16.2.1/24	192.168.1.1/24
BIRD 1.5.0-1	192.168.1.1/24	10.10.3.2/24
BIRD 1.5.0-4	10.10.3.2/24	192.168.4.2/24
BIRD 1.5.0-5	192.168.4.2/24	172.16.1.2/24

Tabel 4.2 mencatat rute yang ditempuh oleh paket dalam pengiriman melalui protokol routing OSPF dengan menggunakan iperf dari PC1 menuju PC2. Proses uji dimulai dari PC1 yang berperan sebagai klien dengan alamat IP 172.16.2.2/24 dan berakhir pada PC2 dengan alamat IP 172.16.1.2/24.

4.1.3. Verifikasi Konfigurasi OSPF

Untuk melakukan verifikasi apakah konfigurasi yang telah dilakukan berhasil maka dilakukan percobaan pengiriman data dengan menggunakan iperf3 dari PC1 ke PC2 untuk memastikan tidak ada kesalahan atau *error* pada konfigurasi. Gambar 4.1 merupakan hasil percobaan pengiriman data dengan menggunakan iperf3 dari PC1 ke PC2 :

```

ipine:~# iperf3 -c 172.16.1.2
Connecting to host 172.16.1.2, port 5201
5] local 172.16.2.2 port 47736 connected to 172.16.1.2 port 5201
ID] Interval          Transfer          Bitrate          Retr  Cwnd
5]  0.00-1.00    sec  2.42 MBytes    20.3 Mbits/sec    0   140 KBytes
5]  1.00-2.00    sec  2.24 MBytes    18.8 Mbits/sec    0   247 KBytes
5]  2.00-3.00    sec  2.34 MBytes    19.6 Mbits/sec    0   363 KBytes
5]  3.00-4.00    sec  2.58 MBytes    21.6 Mbits/sec    0   380 KBytes
5]  4.00-5.00    sec  2.57 MBytes    21.6 Mbits/sec    0   380 KBytes
5]  5.00-6.00    sec  2.63 MBytes    22.1 Mbits/sec    0   380 KBytes
5]  6.00-7.00    sec  2.12 MBytes    17.8 Mbits/sec   149  266 KBytes
5]  7.00-8.00    sec  2.73 MBytes    22.9 Mbits/sec    4   208 KBytes
5]  8.00-9.00    sec  2.24 MBytes    18.8 Mbits/sec    0   222 KBytes
5]  9.00-10.00   sec  2.73 MBytes    22.9 Mbits/sec    0   229 KBytes
-----
ID] Interval          Transfer          Bitrate          Retr
5]  0.00-10.00   sec  24.6 MBytes    20.6 Mbits/sec   153
5]  0.00-10.04   sec  24.3 MBytes    20.3 Mbits/sec
sender
receiver

perf Done.
ipine:~#

```

Gambar 4. 3 Verifikasi Jaringan OSPF dengan menggunakan iperf

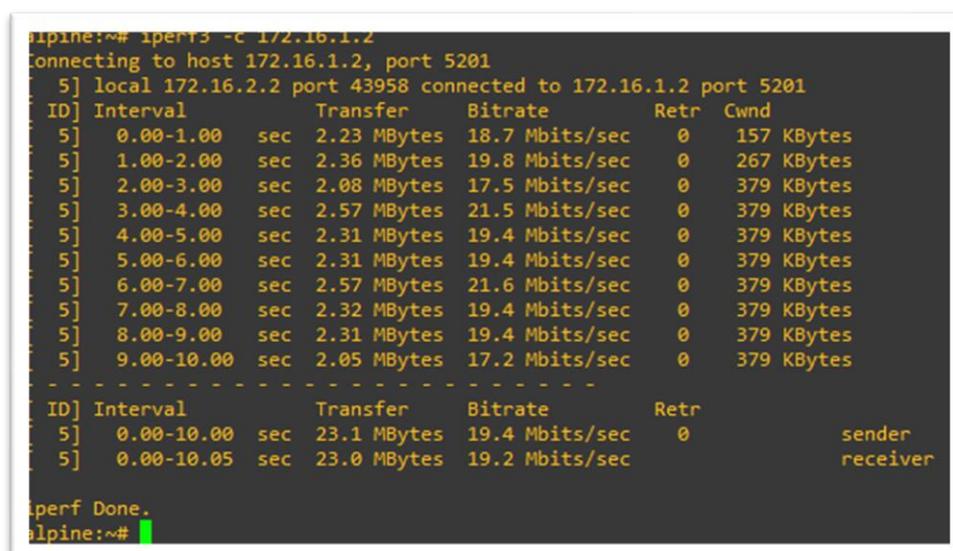
Dengan begitu maka koneksi berhasil dan dapat dilakukan pengambilan data dengan menggunakan *wireshark* untuk dilakukan pengujian dan analisis. Proses ini dilakukan berulang kali dengan menggunakan iperf dan memiliki beberapa beban yang berbeda, serta dengan protokol yang berbeda. Protokol yang digunakan adalah TCP dan UDP dengan beberapa beban yang dilakukan pengambilan yaitu 15M, 30M, dan 50M dengan masing-masing beban dilakukan pengambilan data sebanyak 10 kali.

“Iperf3 -c” merupakan perintah yang digunakan untuk melakukan pengujian dengan menggunakan iperf. Dan “-c” sendiri adalah perintah yang digunakan untuk menandakan bahwa PC yang menggunakan perintah tersebut merupakan PC *client* dan merupakan asal dari pengiriman paket yang dilakukan.

Pada PC *server* digunakan “-s” yang menunjukkan bahwa pc tersebut adalah PC *server*.

4.1.4. Verifikasi Konfigurasi RIP

Berikut merupakan dilakukan percobaan pengiriman data dengan menggunakan iperf3 dari PC1 ke PC2 untuk memastikan tidak ada kesalahan atau *error* pada konfigurasi ping dari PC1 ke PC2 pada hasil konfigurasi routing protokol RIP:



```

alpine:~# iperf3 -c 172.16.1.2
Connecting to host 172.16.1.2, port 5201
5] local 172.16.2.2 port 43958 connected to 172.16.1.2 port 5201
ID] Interval      Transfer      Bitrate      Retr  Cwnd
5]  0.00-1.00    sec  2.23 MBytes  18.7 Mbits/sec  0    157 KBytes
5]  1.00-2.00    sec  2.36 MBytes  19.8 Mbits/sec  0    267 KBytes
5]  2.00-3.00    sec  2.08 MBytes  17.5 Mbits/sec  0    379 KBytes
5]  3.00-4.00    sec  2.57 MBytes  21.5 Mbits/sec  0    379 KBytes
5]  4.00-5.00    sec  2.31 MBytes  19.4 Mbits/sec  0    379 KBytes
5]  5.00-6.00    sec  2.31 MBytes  19.4 Mbits/sec  0    379 KBytes
5]  6.00-7.00    sec  2.57 MBytes  21.6 Mbits/sec  0    379 KBytes
5]  7.00-8.00    sec  2.32 MBytes  19.4 Mbits/sec  0    379 KBytes
5]  8.00-9.00    sec  2.31 MBytes  19.4 Mbits/sec  0    379 KBytes
5]  9.00-10.00   sec  2.05 MBytes  17.2 Mbits/sec  0    379 KBytes
-----
ID] Interval      Transfer      Bitrate      Retr
5]  0.00-10.00   sec  23.1 MBytes  19.4 Mbits/sec  0
5]  0.00-10.05   sec  23.0 MBytes  19.2 Mbits/sec
iperf Done.
alpine:~#

```

Gambar 4. 4 Verifikasi Jaringan RIP dengan menggunakan iperf

Pada protokol jaringan RIP juga dilakukan hal yang sama yaitu dilakukan dengan iperf serta memiliki beban pengambilan data yaitu 15M, 30M, dan 50M dengan masing-masing beban dilakukan pengambilan data sebanyak 10 kali. Protokol yang digunakan yaitu TCP dan UDP.

Pada protocol *routing* RIP juga digunakan perintah "iperf3 -c" digunakan untuk melaksanakan pengujian menggunakan aplikasi iperf. Opsi "-c" dalam perintah ini memberi petunjuk bahwa perangkat PC yang memanfaatkan perintah tersebut berfungsi sebagai klien, bertindak sebagai sumber pengiriman paket dalam uji coba. Di sisi lain, pada PC yang berperan sebagai server, perintah "-s"

digunakan untuk mengindikasikan bahwa PC tersebut berfungsi sebagai server dalam skenario pengujian.

4.1.5. Pengujian Jaringan

Tahap pengujian dilakukan dengan *capture* pada pengiriman data antara PC1 ke PC2 dengan menggunakan *wireshark*. Data yang diambil merupakan pengiriman protokol UDP dan TCP. Proses pengujian diambil dengan masing uji berjumlah sepuluh kali yang dilakukan pada beban 15M, 30M, dan 50M.

Hasil pengujian tersebut akan menjadi patokan untuk melakukan perhitungan data. Hasil dari perhitungan data akan menjadi parameter yang digunakan untuk menentukan QOS dari kedua jaringan *routing protocol* RIP dan OSPF menggunakan *BIRD routing*.

No.	Time	Source	Destination	Protocol	Length	Info
24	31.325048	172.16.2.2	172.16.1.2	UDP	46	60386 → 5201 Len=4
29	31.331567	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
30	31.341456	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
31	31.352406	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
32	31.363401	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
33	31.374531	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
34	31.385598	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
35	31.397571	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
36	31.408522	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
37	31.419567	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
38	31.430572	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
39	31.441535	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
40	31.452620	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
41	31.463618	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
42	31.474537	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
43	31.485721	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
44	31.496626	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
45	31.507617	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
46	31.518595	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
47	31.529623	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
48	31.540591	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
49	31.551612	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
50	31.562646	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
51	31.573669	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
52	31.584306	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
53	31.595700	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
54	31.606788	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
55	31.617652	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
56	31.629664	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
57	31.639670	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
58	31.650663	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
59	31.661735	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448
60	31.672696	172.16.2.2	172.16.1.2	UDP	1490	60386 → 5201 Len=1448

Gambar 4. 5 Wireshark udp

Pada proses pengujian *wireshark* dilakukan untuk memonitoring pengiriman data yang terjadi. Setelah monitoring selesai dilakukan *filtering* dengan menggunakan perintah “`udp && ip.dst == 172.16.1.2 && ip.src=172.16.2.2`”. Dimana perintah tersebut digunakan untuk menyaring pengiriman UDP yang dengan destinasi yang berasal dari ip 172.16.1.2 atau dari PC *server* dan *source* atau asal yang berasal dari ip 172.16.2.2 atau pc *client*.

Time	Source	Destination	Protocol	Length	Info
4	11.312911	172.16.2.2	172.16.1.2	TCP	74 46388
6	11.316312	172.16.2.2	172.16.1.2	TCP	66 46388
7	11.316738	172.16.2.2	172.16.1.2	TCP	103 46388
10	11.319753	172.16.2.2	172.16.1.2	TCP	66 46388
11	11.319922	172.16.2.2	172.16.1.2	TCP	70 46388
12	11.320225	172.16.2.2	172.16.1.2	TCP	181 46388
15	11.326388	172.16.2.2	172.16.1.2	TCP	74 46398
17	11.327826	172.16.2.2	172.16.1.2	TCP	66 46398
18	11.328088	172.16.2.2	172.16.1.2	TCP	103 46398
22	11.335103	172.16.2.2	172.16.1.2	TCP	66 46388
23	11.336282	172.16.2.2	172.16.1.2	TCP	1514 46398
24	11.336293	172.16.2.2	172.16.1.2	TCP	1514 46398
25	11.336305	172.16.2.2	172.16.1.2	TCP	1514 46398
26	11.336309	172.16.2.2	172.16.1.2	TCP	1514 46398
27	11.336315	172.16.2.2	172.16.1.2	TCP	1514 46398
33	11.342414	172.16.2.2	172.16.1.2	TCP	1514 46398
34	11.342425	172.16.2.2	172.16.1.2	TCP	1514 46398
35	11.342434	172.16.2.2	172.16.1.2	TCP	1514 46398
36	11.342438	172.16.2.2	172.16.1.2	TCP	1514 46398
37	11.342442	172.16.2.2	172.16.1.2	TCP	1514 46398
42	11.348598	172.16.2.2	172.16.1.2	TCP	1514 46398
43	11.348608	172.16.2.2	172.16.1.2	TCP	1514 46398
45	11.348996	172.16.2.2	172.16.1.2	TCP	1514 46398
46	11.349006	172.16.2.2	172.16.1.2	TCP	1514 46398
47	11.349010	172.16.2.2	172.16.1.2	TCP	1514 46398
48	11.349019	172.16.2.2	172.16.1.2	TCP	1514 46398
49	11.349172	172.16.2.2	172.16.1.2	TCP	1514 46398
50	11.349186	172.16.2.2	172.16.1.2	TCP	1514 46398
51	11.349190	172.16.2.2	172.16.1.2	TCP	1514 46398
52	11.349194	172.16.2.2	172.16.1.2	TCP	1514 46398
53	11.349294	172.16.2.2	172.16.1.2	TCP	1514 46398
54	11.349302	172.16.2.2	172.16.1.2	TCP	1514 46398
55	11.349306	172.16.2.2	172.16.1.2	TCP	1514 46398

Gambar 4. 6 Wireshark TCP

Wireshark pada TCP digunakan perintah “tcp && ip.dst == 172.16.1.2 && ip.src=172.16.2.2”. Dimana perintah “tcp” berfungsi untuk menyaring pengiriman dengan protokol tcp pada saat proses pengiriman data.

Hasil dari pengujian ini dimanfaatkan untuk ekstraksi informasi dengan menerapkan perhitungan berdasarkan nilai-nilai yang relevan sebagai variabel. Pengumpulan data juga dilakukan sesuai dengan prinsip protokol jaringan yang telah diujikan. Hasil data yang diperoleh akan menjadi faktor yang berperan dalam penentuan Kualitas Layanan (QOS) pada jaringan RIP dan OSPF.

4.1.6. Pengambilan Data

Pengambilan data pada masing-masing router dilakukan dengan menghitung parameter yang telah ditentukan, parameter-parameter tersebut adalah *jitter*, *delay*, *throughput*, dan *packet loss*.

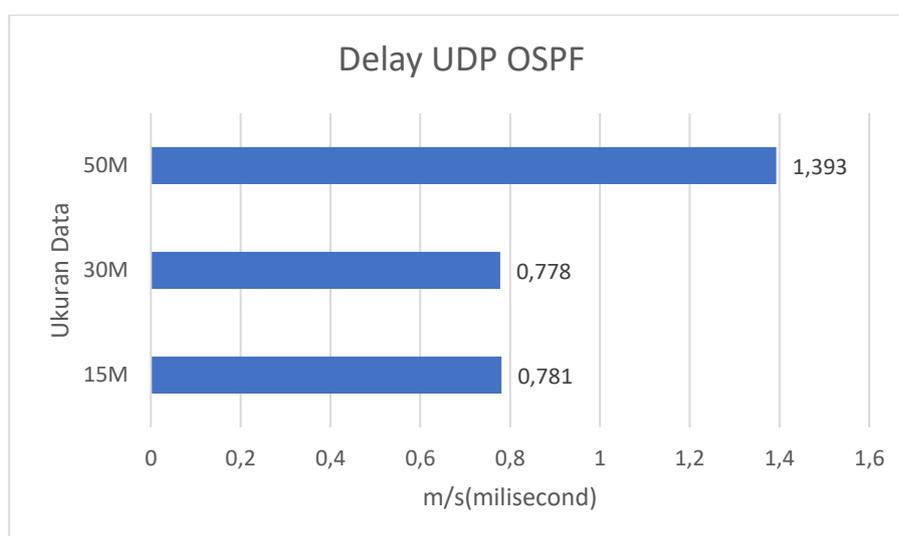
Parameter yang pertama yaitu *delay*. *Delay* sendiri bertujuan untuk menguji selisih waktu yang dibutuhkan pada saat pengiriman data. Perhitungan delay akan dihitung sesuai rumus yang terdapat pada rumus 2.1. Terdapat dua

protokol yang akan dilakukan perhitungan delay yaitu protokol UDP dan protokol TCP. Perhitungan dilakukan dengan menggunakan data yang telah diambil sebelumnya. Berikut merupakan tabel *delay* protokol UDP dan TCP pada routing protokol OSPF:

Tabel 4. 3 Delay UDP OSPF

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15M B	0,78 ms	0,79 ms	0,77 ms	0,76 ms	0,78 ms	0,76 ms	0,77 ms	0,76 ms	0,88 ms	0,76 ms	0,781 ms
30M B	0,77 ms	0,77 ms	0,78 ms	0,76 ms	0,77 ms	0,83 ms	0,81 ms	0,79 ms	0,76 ms	0,74 ms	0,778 ms
50M B	1,08 ms	1,36 ms	1,4ms	1,46 ms	1,45 ms	1,49 ms	1,42 ms	1,47 ms	1,47 ms	1,33 ms	1,393 ms

Tabel 4.1 diatas merupakan hasil *delay* dari setiap proses pengujian yang dilakukan dengan menggunakan protokol UDP pada routing protokol OSPF, setiap nilai *delay* adalah rata-rata pada setiap pengujian.



Gambar 4. 7 Delay UDP OSPF

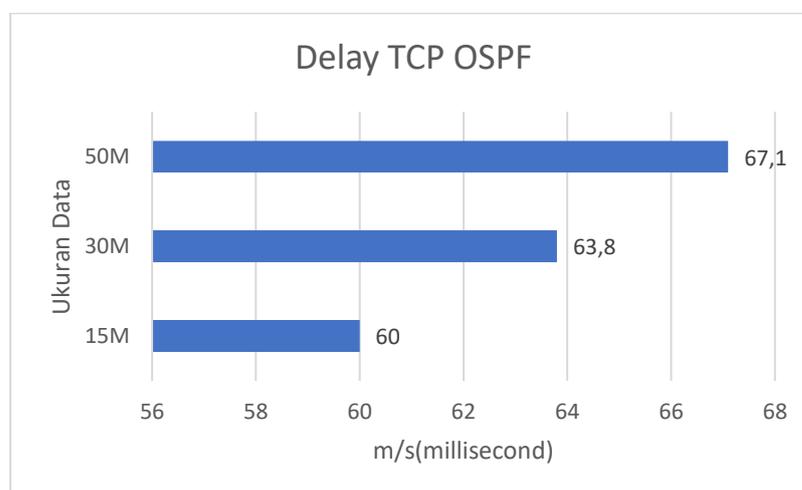
Gambar 4.3 diatas merupakan grafik dari *delay* pada protokol UDP pada routing protokol OSPF dari hasil pengujian yang telah dilakukan. Berdasarkan

grafik diatas pada ukuran data 50M terdapat kenaikan delay yang cukup signifikan dengan nilai delay sebesar 1,393ms. Sedangkan pada pengujian ukuran data lainnya *delay* memiliki nilai yang stabil. Dengan nilai 0,781ms pada beban 15M dan pada beban 30M memiliki nilai delay sebesar 0,778ms

Tabel 4. 4 Delay TCP OSPF

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15MB	64ms	60ms	64ms	62ms	59ms	57ms	60ms	58ms	58ms	58ms	60ms
30MB	65ms	67ms	52ms	64ms	65ms	66ms	64ms	65ms	65ms	65ms	63,8ms
50MB	67ms	67ms	68ms	66ms	68ms	67ms	67ms	68ms	66ms	67ms	67,1ms

Tabel 4.2 diatas adalah hasil pengujian dengan *delay* pada protokol pengiriman TCP. Nilai tersebut merupakan rata-rata dari setiap hasil uji pada setiap beban yang diambil dari banyak 10 kali percobaan pada masing-masing beban.



Gambar 4. 8 Delay TCP

Berdasarkan grafik 4.4, untuk ukuran data yang dikirim lebih besar maka semakin besar pula untuk *delay*-nya. Pada beban 15M memiliki nilai delay terkecil dari ketiga beban pengujian yaitu sebesar 60ms. Kemudian pada beban yang lebih tinggi yaitu 30M delay mengalami kenaikan menjadi sebesar 63,8ms. Dan pada

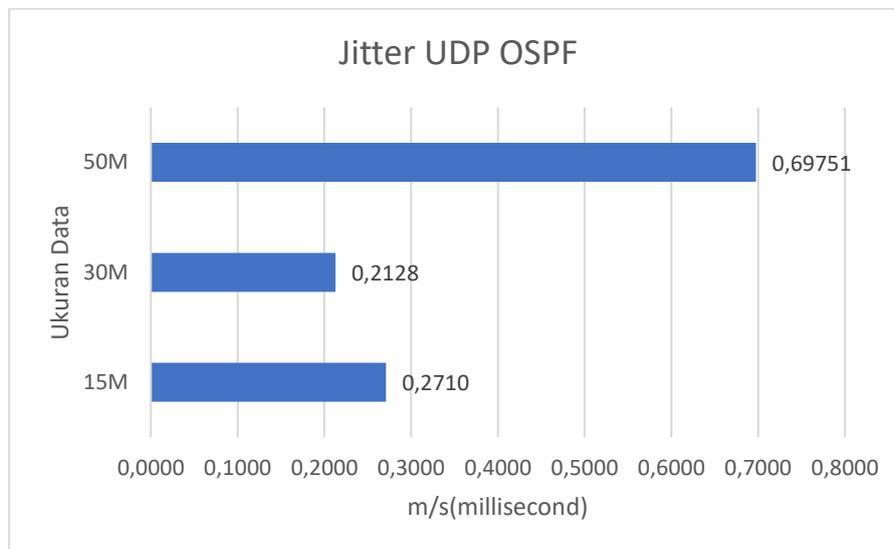
beban 50M delay kembali mengalami kenaikan sehingga nilai delay menjadi 67,1 ms. Dengan demikian dapat disimpulkan bahwa pada protokol TCP, seiring dengan bertambahnya beban yang dikirimkan maka delay juga akan mengalami kenaikan.

Selanjutnya parameter yang dihitung adalah jitter. Jitter merupakan variasi dari *delay* yang terjadi pada jaringan. Jitter adalah salah satu parameter yang digunakan untuk pengambilan data. Pengujian *jitter* hanya dilakukan pada protokol UDP. Berikut merupakan tabel hasil pengambilan data *jitter* pada protokol UDP:

Tabel 4. 5 Jitter UDP OSPF

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15M B	0,196 ms	0,249 ms	0,15 ms	0,191 ms	0,202 ms	0,149 ms	0,181 ms	0,341 ms	0,834 ms	0,217 ms	0,271 ms
30M B	0,207 ms	0,174 ms	0,187 ms	0,294 ms	0,194 ms	0,215 ms	0,287 ms	0,178 ms	0,202 ms	0,19 ms	0,2128 ms
50M B	0,59 ms	0,559 ms	0,779 ms	0,773 ms	0,755 ms	1,100 ms	0,515 ms	0,703 ms	0,639 ms	0,562 ms	0,6975 ms

Pada tabel 4.3 merupakan tabel *jitter* yang didapatkan dari hasil pengujian protokol UDP. Karena *jitter* merupakan variasi delay maka hasil dari *jitter* tidak terlalu berbeda dengan *delay*. Hasil yang didapatkan adalah pada beban 50M memiliki nilai *jitter* yang lebih tinggi dibandingkan dengan kedua beban lain.



Gambar 4. 9 Jitter UDP

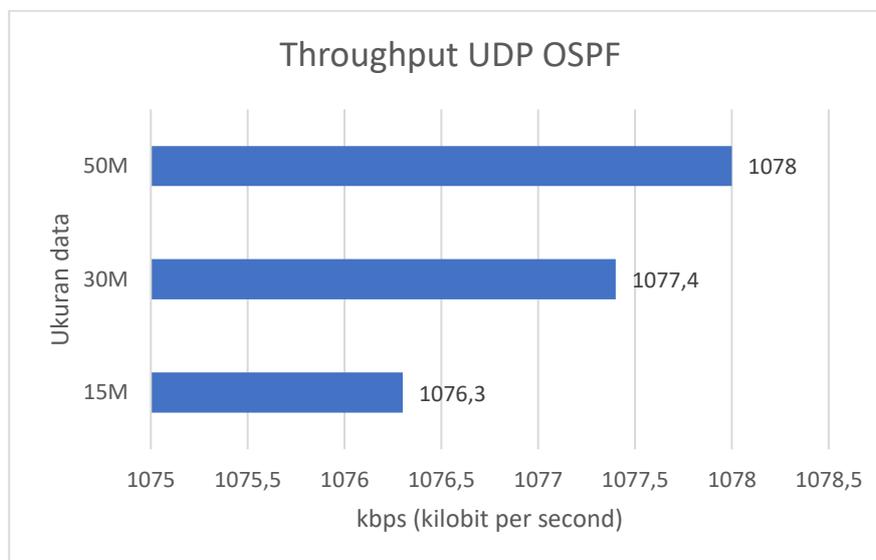
Berdasar grafik pada gambar 4.5 *jitter* tertinggi terdapat pada beban 50M. Pada beban 50M memiliki kenaikan yang cukup signifikan dikarenakan nilai *delay* pada beban ini juga memiliki kenaikan. Pada beban 50M memiliki nilai *jitter* sebesar 0,69751ms. Pada beban 30M memiliki nilai *jitter* sebesar 0,2128ms. Dan pada 15M memiliki nilai *jitter* sebesar 0,2710ms.

Parameter selanjutnya yang digunakan untuk pengambilan data adalah *throughput*, yang merupakan kemampuan jaringan dalam melakukan pengiriman data. Pengujian *throughput* bertujuan untuk mengukur kehandalan jaringan dalam mengirim data paket. Berikut merupakan nilai *throughput* dari protokol UDP:

Tabel 4. 6 *Throughput* UDP OSPF

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15MB	1075k	1071k	1081k	1076k	1068k	1085k	1074k	1080k	1067k	1086k	1076,3k
30MB	1074k	1080k	1078k	1077k	1077k	1078k	1078k	1078k	1076k	1078k	1077,4k
50MB	1079k	1077k	1078k	1079k	1078k	1077k	1076k	1080k	1079k	1077k	1078k

Tabel 4.4 merupakan hasil *throughput* dari pengujian yang telah dilakukan.



Gambar 4. 10 Throughput UDP OSPF

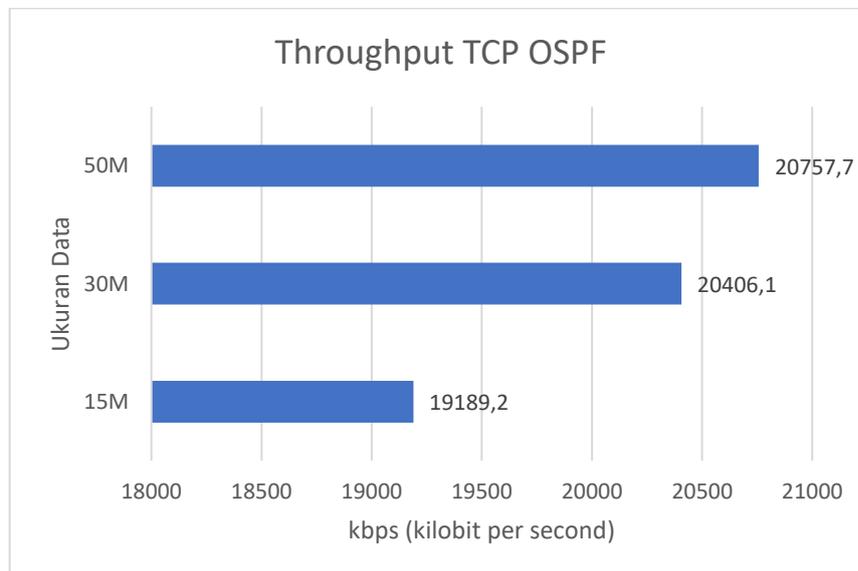
Berdasarkan grafik pada gambar 4.6 setiap peningkatan beban, terdapat pula peningkatan pada hasil *throughput* yang didapatkan. Pada *throughput* protokol UDP semakin tinggi beban maka *throughput* yang dihasilkan juga akan mengalami kenaikan. *Throughput* terkecil terdapat pada beban 15M dengan nilai 1076,3 kbps. Mengalami kenaikan pada beban yang lebih tinggi, pada beban 30M nilai *throughput* adalah 1077,4 kbps. Dan nilai *throughput* terbesar terdapat pada beban pengujian tertinggi yaitu 50M dengan nilai *throughput* sebesar 1078 kbps.

Selanjutnya merupakan pengujian *throughput* pada protokol TCP. Berikut merupakan hasil pengambilan data *throughput* pada protokol TCP:

Tabel 4. 7 *Throughput* TCP OSPF

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15M B	1859 5k	1987 3k	1749 0k	1792 9k	2004 5k	1959 5k	1964 2k	1987 6k	1946 9k	1937 8k	19189, 2k
30M B	2072 8k	1968 0k	1923 1k	2049 4k	2048 5k	2076 3k	2057 1k	2058 1k	2085 0k	2067 8k	20406, 1k
50M B	2088 6k	2077 6k	2077 2k	2063 7k	2084 1k	2068 2k	2080 6k	2072 1k	2062 5k	2083 1k	20757, 7k

Pada tabel 4.5 diatas merupakan hasil throughput yang didapat setelah proses pengujian pada protokol TCP.



Gambar 4. 11 Throughput TCP

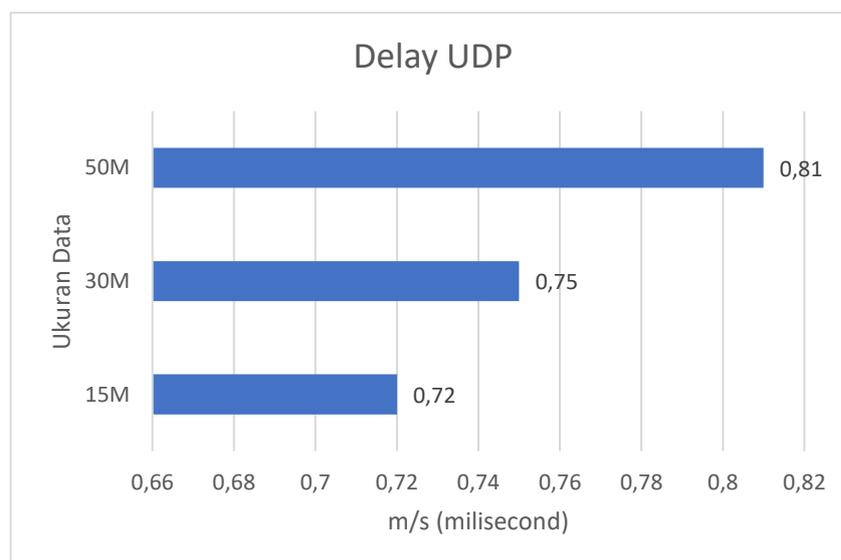
Berdasarkan pada gambar 4.7 diatas nilai *throughput* mengalami kenaikan seiring naiknya beban yang diuji. Pada 15M memiliki nilai *throughput* sebesar 19189,2 kbps. Pada 30M memiliki nilai *throughput* sebesar 20406,1 kbps. Dan nilai *throughput* terbesar terdapat pada beban 50M yaitu sebesar 10757,7 kbps.

Setelah melakukan pengambilan data pada routing protokol OSPF. Selanjutnya merupakan pengambilan data pada routing protokol RIP. Parameter dan proses pengambilan data yang dilakukan pada routing protokol RIP sama dengan proses yang dilakukan pada routing protokol OSPF. Yaitu protokol yang diuji adalah TCP dan UDP, serta memiliki beban sebesar 15M, 30M, dan 50M sebanyak masing-masing 10 kali uji. Dan parameter yang dihitung adalah *delay*, *jitter*, dan *throughput*.

Untuk parameter yang pertama adalah pengambilan data pada *delay* terlebih dahulu. Berikut merupakan hasil dari pengujian *delay* protokol UDP pada routing protokol RIP:

Tabel 4. 8 *Delay* UDP RIP

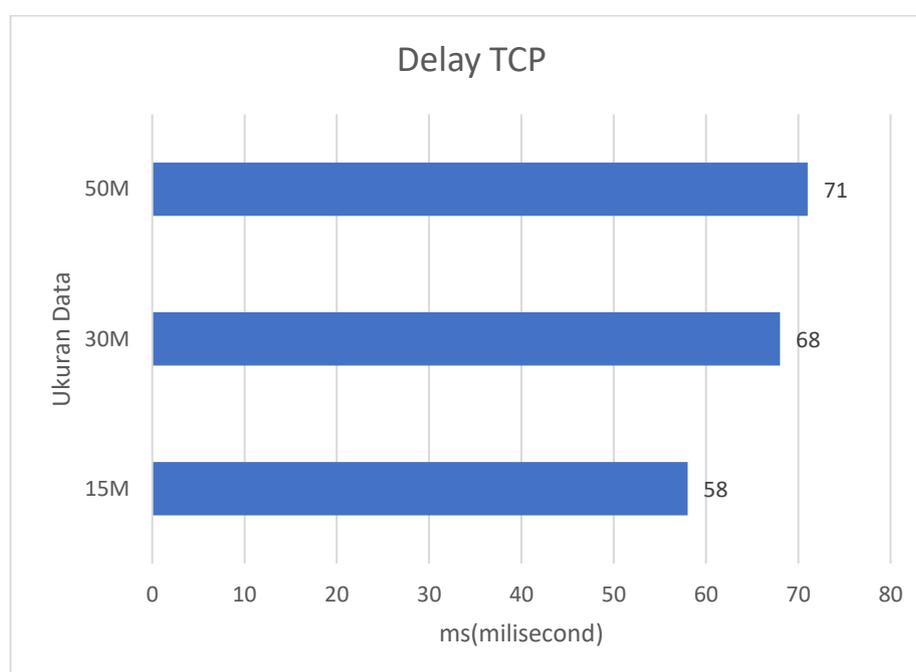
Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15M B	0,7ms	0,71ms	0,71ms	0,71ms	0,71ms	0,72ms	0,79ms	0,71ms	0,71ms	0,71ms	0,72ms
30M B	0,7ms	0,72ms	0,74ms	0,71ms	0,72ms	0,71ms	0,71ms	0,78ms	0,86ms	0,79ms	0,75ms
50M B	0,79ms	0,79ms	0,79ms	0,96ms	0,76ms	0,75ms	0,87ms	0,75ms	0,89ms	0,75ms	0,81ms

Gambar 4. 12 *Delay* UDP RIP

Dengan melihat dari hasil grafik pada gambar 4.8 dan tabel 4.6 di atas semakin besar ukuran data yang dikirim semakin tinggi pula *delay* yang didapatkan. Nilai *delay* terkecil terdapat pada ukuran beban terkecil yaitu 15M dengan nilai *delay* sebesar 0,72ms. Selanjutnya pada beban 30M *delay* naik menjadi 0,75ms. Dan *delay* tertinggi terdapat pada ukuran beban tertinggi yaitu 50M dengan nilai 0,81ms.

Tabel 4. 9 Delay TCP RIP

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15MB	40ms	66ms	60ms	60ms	62ms	59ms	61ms	60ms	46ms	63ms	58ms
30MB	70ms	71ms	68ms	72ms	47ms	72ms	71ms	70ms	70ms	68ms	68ms
50MB	71ms	69ms	70ms	75ms	71ms	71ms	71ms	75ms	65ms	71ms	71ms



Gambar 4. 13 Delay TCP RIP

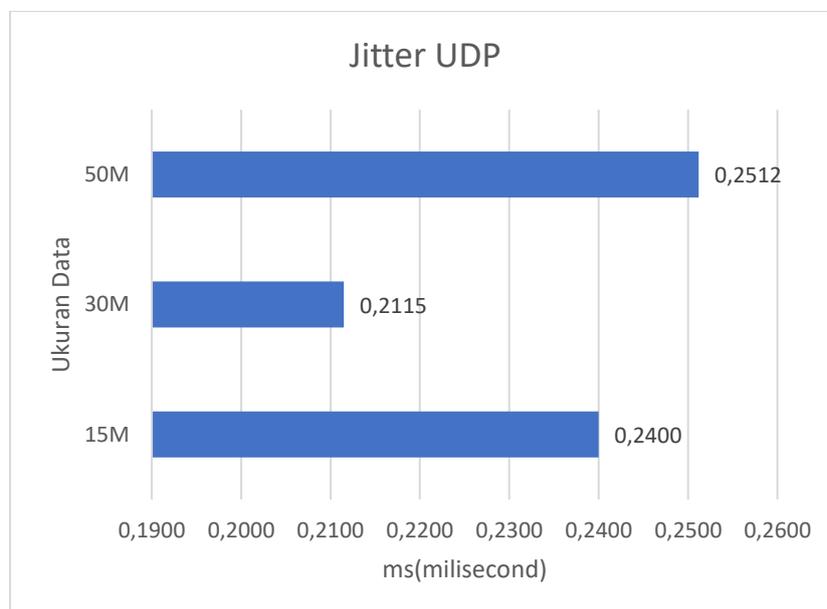
Hasil yang sama juga terdapat pada protokol TCP dimana semakin besar ukuran data yang dikirim maka ukuran delay akan semakin besar pula. Pada beban 15M hasil dari delay yang didapat adalah 58ms. Pada beban 30M delay mengalami kenaikan menjadi 68ms. Dan semakin tinggi pada beban 50M yaitu dengan nilai 71ms. Dari hasil tersebut dapat disimpulkan bahwa semakin tinggi beban yang dikirim nilai delay akan mengalami kenaikan juga.

Parameter selanjutnya adalah jitter UDP pada routing protokol RIP. Berikut merupakan hasil pengujian *jitter* UDP:

Tabel 4. 10 *Jitter* UDP RIP

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15M B	0,281 ms	0,108 ms	0,239 ms	0,146 ms	0,239 ms	0,156 ms	0,72 ms	0,156 ms	0,109 ms	0,246 ms	0,240 0ms
30M B	0,157 ms	0,16 ms	0,353 ms	0,159 ms	0,215 ms	0,175 ms	0,219 ms	0,205 ms	0,212 ms	0,26 ms	0,211 5ms
50M B	0,172 ms	0,106 ms	0,162 ms	0,175 ms	0,311 ms	0,186 ms	0,211 ms	0,108 ms	0,869 ms	0,212 ms	0,251 2ms

Tabel 4.8 menunjukkan hasil pengujian *jitter* pada protokol UDP.



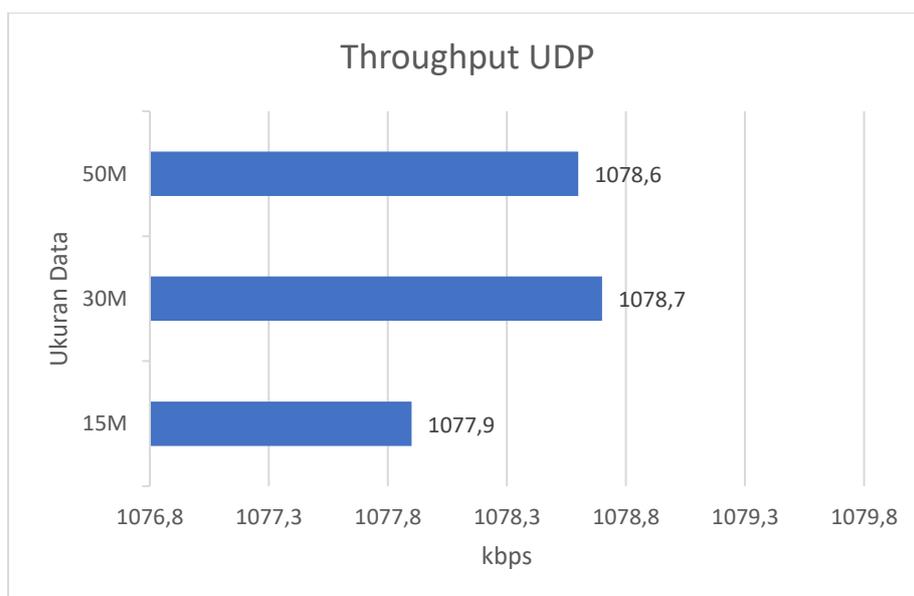
Gambar 4. 14 Jitter UDP

Jitter pada beban 30M memiliki ukuran terkecil yaitu 0,2115 dan nilai tertinggi berada pada beban 50M dengan nilai 0,2400. Jika dilihat dari hasil yang telah didapatkan ukuran jitter tidak ditentukan oleh ukuran beban yang dikirimkan. Karena pada hasil yang terdapat pada gambar 4.10 jitter pada beban 15M lebih tinggi dibandingkan dengan jitter pada beban 30M.

Tabel 4. 11 *Throughput* UDP RIP

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15MB	1087k	1059k	1082k	1079k	1080k	1079k	1078k	1078k	1079k	1078k	1077,9k
30MB	1078k	1078k	1078k	1079k	1078,7k						
50MB	1079k	1079k	1079k	1079k	1079k	1079k	1078k	1078k	1078k	1078k	1078,6k

Tabel 4.9 adalah hasil pengujian *throughput* pada protokol jaringan UDP. *Throughput* tidak memiliki perbedaan yang terlalu signifikan antar beban pengujian.

Gambar 4. 15 *Throughput* UDP

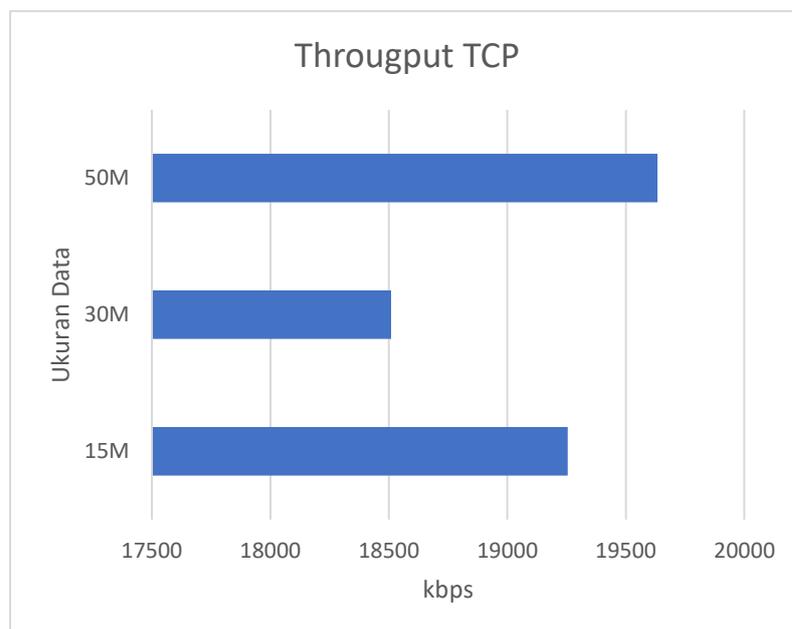
Ukuran *throughput* pada tiap beban tidak memiliki perbedaan yang terlalu besar. Pada beban 30M memiliki nilai *throughput* tertinggi dan nilai *throughput* terkecil terdapat pada beban 15M. Semakin tinggi *throughput* artinya semakin mendekati dengan *bandwidth*.

Selanjutnya *throughput* dari protokol jaringan TCP pada pengujian jaringan routing protokol RIP. Berikut merupakan hasil pengujian hasil *throughput* TCP:

Tabel 4. 12 Throughput TCP RIP

Beban	pengujian										rata-rata
	1	2	3	4	5	6	7	8	9	10	
15MB	1946 7k	1981 0k	1968 0k	1954 8k	1968 2k	1951 3k	1987 3k	1953 7k	1608 4k	1935 8k	19255, 2k
30MB	1357 6k	1893 0k	1981 4k	1836 9k	1767 5k	1899 3k	1937 6k	1913 6k	1930 7k	1991 6k	18509, 2k
50MB	2009 2k	2042 0k	2039 2k	1907 3k	1954 2k	2006 7k	2028 8k	1932 4k	1781 0k	1933 3k	19634, 1k

Berdasarkan pada tabel 4.10 diatas nilai *throughput* tertinggi terdapat pada beban 50M dengan nilai 19634,1 kbps. Dan nilai terendah pada beban 30M dengan nilai 18509,2 kbps. Selanjutnya pada beban 15M nilai *throughput* sebesar 19255,2 kbps.



Gambar 4. 16 Throughput TCP

Dilihat berdasarkan gambar 4.12 diatas *throughput* bahwa nilai *throughput* terendah terdapat pada ukuran beban 30M. Dan nilai terbesar terdapat pada beban 50M. Untuk ukuran beban 15M berada pada tengah dari kedua beban tersebut.

4.1.7. Analisis Data

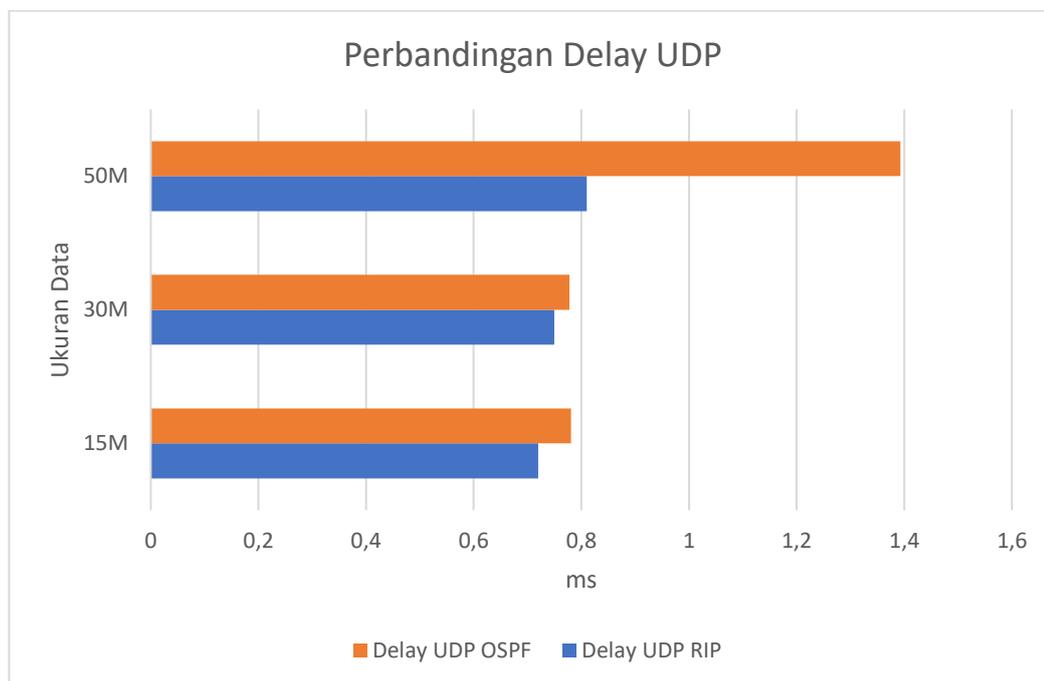
Pada sub-bab ini dilakukan perbandingan antara kedua routing protokol yaitu OSPF dan RIP guna mengetahui perbandingan kualitas jaringan. Dari perbandingan tersebut nantinya akan dilakukan analisis untuk memastikan jaringan mana yang lebih baik.

Perbandingan yang pertama dilakukan adalah perbedaan *delay* kedua routing protokol pada jaringan UDP. Pada tabel 4.11 merupakan perbandingan *delay* antara jaringan OSPF dan RIP :

Tabel 4. 13 Perbandingan Delay UDP

Ukuran Data	Delay UDP RIP	Delay UDP OSPF
15M	0,72ms	0,78ms
30M	0,75ms	0,78ms
50M	0,81ms	1,39ms
Rata-rata	0,76ms	0,98ms

Jika dilihat berdasarkan tabel 4.11 delay pada routing protokol RIP memiliki nilai yang lebih rendah dibanding dengan OSPF. Kedua routing protokol memiliki nilai yang stabil pada ukuran data yang relatif kecil, tetapi pada ukuran data 50M kedua routing protokol memiliki kenaikan delay yang cukup signifikan. Berikut merupakan grafik dari perbandingan delay antara RIP dan OSPF :



Gambar 4. 17 Perbandingan Delay UDP

Berdasarkan grafik diatas diatas dapat disimpulkan bahwa pada pengiriman dengan menggunakan UDP, protokol RIP memiliki nilai yang lebih baik dibandingkan dengan OSPF. Dan pada beban 50M pada OSPF memiliki lonjakan *delay* yang cukup besar.

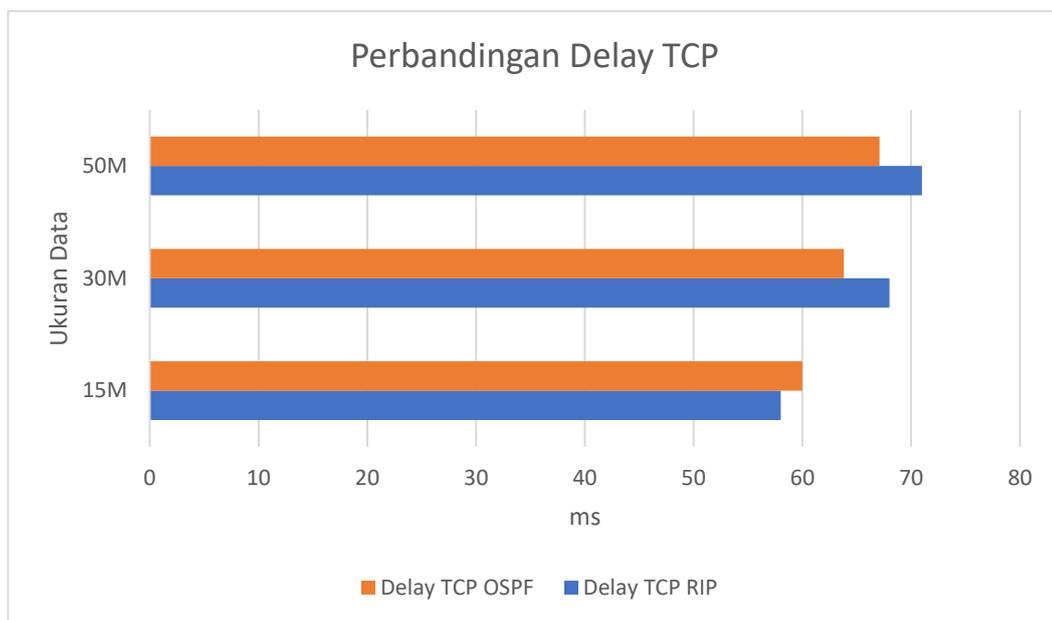
Parameter selanjutnya yang menjadi acuan adalah *delay* pada jaringan TCP. Berikut merupakan tabel hasil perbandingan antara kedua routing protokol :

Tabel 4. 14 Perbandingan Delay TCP

Ukuran Data	Delay TCP RIP	Delay TCP OSPF
15M	58ms	60ms
30M	68ms	63,8ms
50M	71ms	67,1ms
Rata-rata	65,6ms	63,6ms

Berdasarkan tabel diatas pada beban 15M delay pada protokol RIP

memiliki nilai yang lebih rendah, yang artinya protokol RIP mempunyai kualitas *delay* yang lebih baik. Akan tetapi pada beban yang lebih besar yaitu 30M, dan 50M OSPF memiliki nilai *delay* yang lebih rendah. Dengan begitu OSPF memiliki kualitas *delay* yang lebih baik pada pengiriman data yang lebih besar pada protokol TCP. Berikut merupakan grafik perbandingan *delay* TCP pada kedua routing protokol :



Gambar 4. 18 Perbandingan Delay TCP

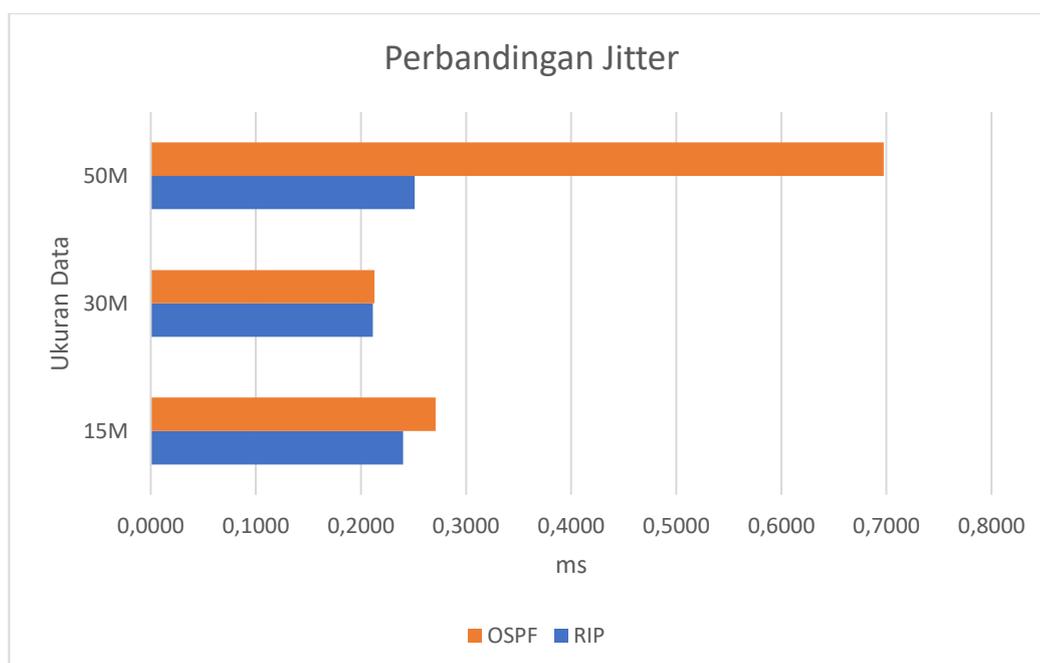
Berdasarkan gambar 4.14 dapat diambil kesimpulan bahwa secara keseluruhan OSPF memiliki nilai yang lebih baik dibandingkan dengan RIP. Namun pada beban 15M delay RIP menunjukkan nilai yang lebih baik.

Jitter menjadi tolak ukur berikutnya yang dijadikan sebagai acuan untuk menganalisa kualitas jaringan antara routing protokol RIP dan OSPF. Berikut adalah tabel perbandingan *jitter* UDP kedua jaringan:

Tabel 4. 15 Perbandingan Jitter

Ukuran Data	RIP	OSPF
15M	0,2400ms	0,2710ms
30M	0,2115ms	0,2128ms
50M	0,2512ms	0,69751ms
Rata-rata	0,2342ms	0,3937ms

Berdasarkan tabel 4.13 tersebut terlihat *jitter* pada RIP memiliki kinerja yang lebih baik. Perbedaan yang sangat terlihat terdapat pada ukuran data 50M, protokol OSPF memiliki nilai *jitter* yang meningkat drastis dari ukuran data sebelumnya. Berikut juga disediakan perbandingan *jitter* kedua jaringan tersebut dalam bentuk grafik:



Gambar 4. 19 Perbandingan Jitter

Dengan melihat hasil pada gambar 4.15 *jitter* yang lebih baik berada pada protokol RIP. Dan pada beban 50M protokol OSPF memiliki kenaikan nilai yang cukup tinggi.

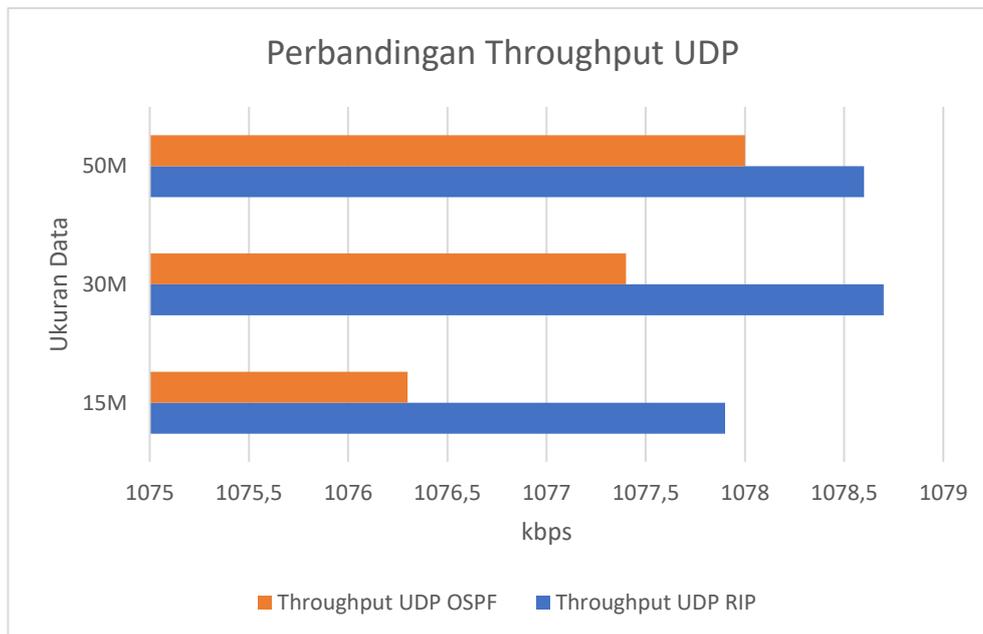
Kemudian adalah *throughput* yang menjadi acuan perbandingan kualitas kedua jaringan terakhir. *Throughput* yang dihitung adalah *throughput* pada protokol UDP dan TCP. Berikut adalah tabel perbandingan kinerja *throughput*

pada protokol UDP kedua jaringan:

Tabel 4. 16 Perbandingan *Throughput* UDP

Ukuran Data	<i>Throughput</i> UDP RIP	<i>Throughput</i> UDP OSPF
15M	1077,9kbps	1076,3kbps
30M	1078,7kbps	1077,4kbps
50M	1078,6kbps	1078kbps
Rata-rata	1.078,4kbps	1.077,2kbps

Nilai *throughput* kedua jaringan tidak memiliki perbedaan yang terlalu besar. Akan tetapi nilai *throughput* pada jaringan RIP sedikit memiliki nilai yang lebih baik dikarenakan jika nilai *throughput* semakin tinggi maka akan semakin dekat dengan *bandwidth*. Berikut merupakan grafik dari perbandingan *throughput* kedua jaringan :



Gambar 4. 20 Perbandingan *Throughput* UDP

Berdasarkan pada gambar 4.16 nilai *throughput* RIP lebih besar dibandingkan dengan OSPF. Hal ini menunjukkan bahwa *throughput* RIP lebih baik dibandingkan dengan OSPF dalam pengiriman data dengan menggunakan

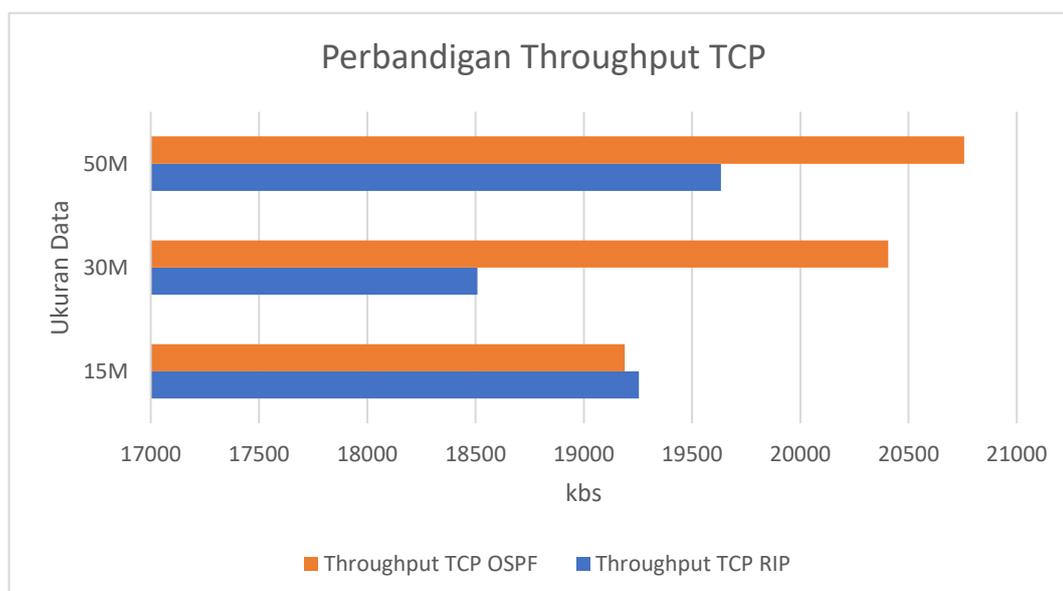
UDP. Nilai kenaikan *throughput* pada OSPF naik secara signifikan seiring bertambahnya nilai beban yang diuji, sedangkan pada RIP kenaikan nilai *throughput* pada beban 50M justru lebih kecil dibandingkan dengan pada beban 30M.

Selain *throughput* pada UDP diukur juga *throughput* pada jaringan TCP. Berikut merupakan tabel perbandingan *throughput* TCP pada kedua routing protokol:

Tabel 4. 17 Perbandingan *Throughput* TCP

Ukuran Data	<i>Throughput</i> TCP RIP	<i>Throughput</i> TCP OSPF
15M	19255,2 kbps	19189,2 kbps
30M	18509,2 kbps	20406,1 kbps
50M	19634,1 kbps	20757,7 kbps
Rata-rata	19.132,8 kbps	20.117,6 kbps

Pada jaringan TCP *throughput* pada OSPF memiliki kinerja yang lebih baik karena memiliki nilai yang lebih besar dibandingkan dengan RIP. Dibawah ini juga disediakan grafik dari perbandingan kedua routing protokol tersebut :



Gambar 4. 21 Perbandingan *Throughput* TCP

Untuk nilai throughput pada TCP sendiri nilai pada protokol OSPF memiliki nilai yang lebih besar, hal ini dapat disimpulkan bahwa secara keseluruhan throughput OSPF lebih baik dibandingkan dengan RIP. Akan tetapi pada beban 15M RIP sedikit lebih baik dibanding OSPF.

Dengan melihat hasil perbandingan yang telah dilakukan, maka dapat diambil kesimpulan bahwa. Kedua routing protokol memiliki kelebihan dan kekurangan jika dibandingkan satu sama lain. Pada *delay* dengan pengiriman UDP, RIP memiliki kinerja yang lebih baik dibandingkan OSPF. Akan tetapi pada pengiriman TCP *delay* OSPF memiliki kinerja yang lebih baik dibandingkan dengan RIP. Begitu pula pada *jitter* RIP memiliki kinerja yang lebih baik. Dan pada *throughput* TCP, OSPF memiliki kinerja yang lebih baik. Namun pada *throughput* UDP, RIP memiliki kinerja yang lebih baik. Dengan demikian dapat disimpulkan secara keseluruhan protokol RIP memiliki kinerja yang lebih baik dibandingkan dengan OSPF pada pengiriman dengan menggunakan protokol UDP. Sedangkan pada pengiriman dengan protokol TCP routing protokol OSPF memiliki kinerja yang lebih baik secara keseluruhan.