

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam tugas akhir ini adalah studi kepustakaan, percobaan dan analisis.

3.1.1 Studi Kepustakaan

Awal mulanya, jaringan internet tidak didesain agar mampu membedakan perlakuan untuk setiap tipe trafik yang berbeda. Hal ini menjadi masalah ketika terdapat berbagai macam trafik yang mempunyai tingkat kebutuhan pelayanan yang berbeda-beda. Apabila ada aplikasi yang mengirimkan trafik dan menghabiskan sumber daya pada jaringan, maka trafik lain yang ingin masuk tentu saja akan dirugikan oleh trafik yang rakus sumber daya tersebut. Selain *Quality of Service* (QoS), dibutuhkan pula mekanisme *traffic engineering* untuk mengatasi permasalahan yang muncul ketika jalur mengalami keadaan kongesti. Salah satu protokol yang mampu menyediakan layanan *traffic engineering* dengan fleksibel adalah *Load balance*.

Load balance mempunyai banyak metode, salah satunya yang digunakan dalam penelitian ini yaitu PCC dan Nth. Kedua metode ini digunakan untuk menentukan metode *load balance* terbaik untuk aplikasi transfer data multimedia. Dalam pengujian pada tugas akhir ini, menggunakan parameter QoS (*Quality of Service*) antara lain *throughput*, *packet loss* dan *fairness index*. *Throughput* merupakan jumlah total kedatangan paket yang diamati pada *destination* selama

interval waktu tertentu dibagi oleh durasi interval waktu. *Packet loss* merupakan paket data yang hilang pada saat pengiriman. Dan yang terakhir *fairness index* untuk menentukan apakah *user* atau aplikasi telah menerima sumber daya yang adil.

3.1.2 Percobaan

Prosedur percobaan adalah prosedur dimana melakukan percobaan *system*, yaitu melakukan konfigurasi pada *router* dan melakukan percobaan pengiriman data dengan *protocol ICMP* antara *client* dengan *server* untuk mengetahui apakah *client* dengan *server* sudah terhubung. Setelah itu dilakukan konfigurasi *load balance* dan menguji *load balance* pada jaringan tersebut untuk mengetahui apakah *load balance* sudah siap pada *router* tersebut atau tidak.

3.1.3 Analisis

Prosedur analisis dilakukan untuk mengetahui sebab akibat dari suatu masalah yang ditemukan. Pada analisis ini, untuk mengetahui nilai *throughput*, *client* mengirimkan data video menuju ke *server*. Pada proses pengiriman tersebut, *packet* dicapture pada *router A* dan *router B*. Hal ini berbeda dengan penghitungan nilai *packet loss*, karena pada *packet loss*, *packet* dicapture pada *router S*. *Fairness index* didapatkan dari nilai *throughput* yang sudah dicapture dan dihitung dengan rumus *fairness index* tersebut.

3.2 Prosedur Penelitian

Prosedur ini menjelaskan tentang langkah – langkah yang akan dilakukan untuk membangun sistem ini, serta langkah – langkah apa saja yang akan dilakukan untuk menguji *system* tersebut. Berikut merupakan prosedur penelitian:

Tabel 3.1 Prosedur penelitian

1. Pengumpulan data dan parameter	Mikrotik, QoS, Load Balance
	Data <i>Video</i>
	Menentukan parameter analisis
2. Desain dan pembuatan topologi	Menentukan topologi <i>load Balance</i>
	Menentukan <i>server</i> dan <i>client</i>
3. Konfigurasi sistem	Konfigurasi <i>load balance</i> pada mikrotik
	Instalasi aplikasi Filezilla pada PC <i>server</i> dan PC <i>client</i>
	Instalasi wireshark pada PC <i>server</i> dan PC <i>client</i>
4. Menjalankan sistem	Mengirimkan data <i>video</i> melalui <i>client</i> ke <i>server</i> dengan menggunakan metode <i>load balance</i> PCC dan Nth secara bergantian
5. Mengolah data	Monitoring data dari wireshark
	Diolah menggunakan rumus pada Microsoft Excel
6. Pengujian system dan plotting	Menganalisis pengujian data
	Membuat hasil plotting dari analisis pengujian data berupa grafik

3.2.1 Pengumpulan Data dan Parameter Penelitian

Dalam tahap ini akan dilakukan pengumpulan data yang akan digunakan untuk melakukan pengujian. Data yang akan digunakan dalam pengujian *system* yaitu data *video*. Data tersebut didapatkan pada saat browsing di *internet*. Ada 3 jenis data *video* dengan ukuran yang berbeda – beda dan menggunakan aplikasi FTP salah satunya *FileZilla* yang akan digunakan untuk mengirimkan data dari *Client* ke *Server*. Setelah itu, PC *Server* dan *Client* menjalankan aplikasi *Wireshark*. *Wireshark* akan diset agar memonitoring paket data TCP dan hasil monitoring tersebut akan dilakukan pada sisi *Server* dan *Client*. Setelah itu hasil

dari monitoring tersebut akan diolah untuk mendapatkan nilai dari *packet loss*, *throughput* dan *fairness index*.

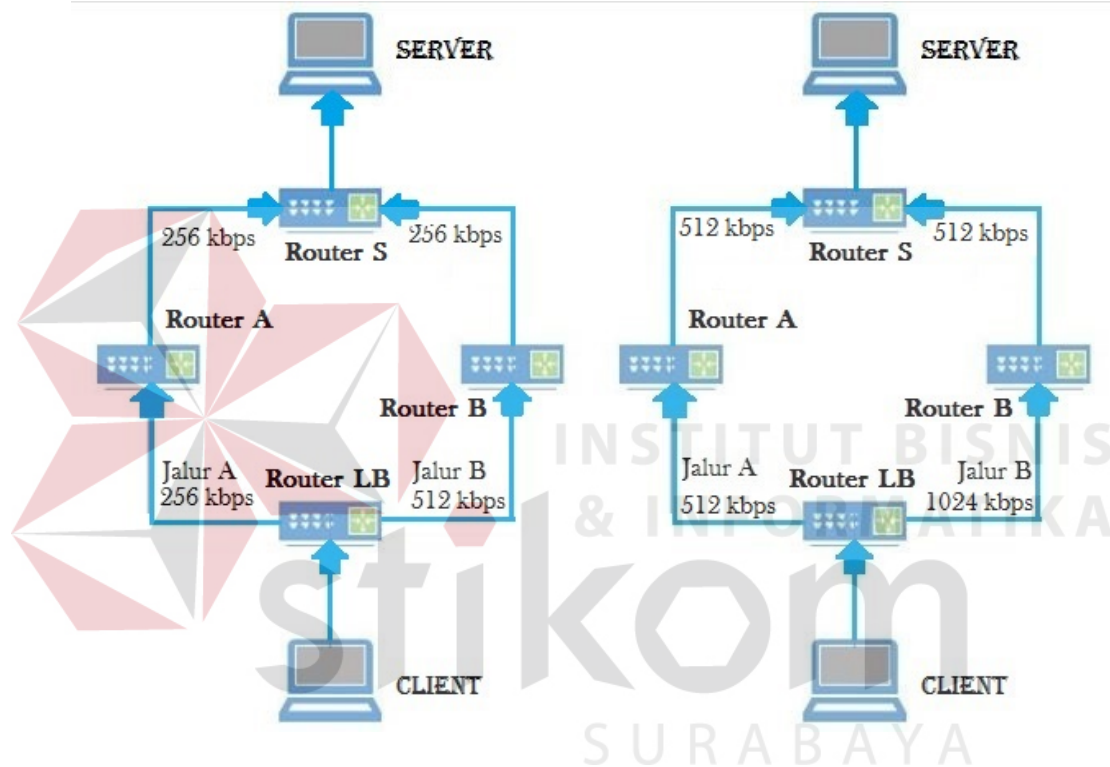
3.2.2 Desain dan Pembuatan Topologi

Gambar 3.1 menggambarkan topologi yang akan diimplementasi dan dianalisis dengan memanfaatkan jaringan *Load Balance* dengan protokol yang berbeda. Di dalam topologi terdapat empat *Router* Mikrotik dan dua *PC*. *Router* Mikrotik yang akan di gunakan berjumlah empat, satu sebagai *Router server*, satu sebagai *Router client*, dan dua *Router* sebagai penyedia jalur untuk *load balance*.

Kemudian keempat *Router* Mikrotik tersebut akan diberi alamat dan dikonfigurasi agar saling terhubung. Setelah semua terhubung, kemudian akan dikonfigurasi menggunakan metode *load balance* PCC dan Nth secara bergantian (tidak berjalan bersamaan). *Load balance* digunakan karena merupakan sistem untuk mengurangi beban *traffic* pada jaringan.

Topologi 1 menggunakan *bandwidth* sebesar 256 kbps pada jalur A dan *bandwidth* 512 kbps pada jalur B, kemudian untuk topologi 2 menggunakan *bandwidth* 512 kbps pada jalur A dan 1024 kbps pada jalur B. Kemudian ada dua *PC*, *PC server* untuk menerima data dan *PC client* untuk mengirim data. Untuk proses *transfer file* ini, menggunakan *software* Filezilla pada *PC server* dan Filezilla pada *PC client*. Proses *transfer* dibagi menjadi tiga sesi, pertama, satu *file video* berukuran ± 5 MB ditransfer dari *PC client* ke *PC server*. Kedua, satu *file video* berukuran ± 10 MB ditransfer dari *PC client* ke *PC server*. Ketiga, satu *file video* berukuran ± 15 MB ditransfer dari *PC client* ke *PC server*. Pada proses *monitoring* trafik penulis menggunakan aplikasi *Network Analysis Tools*, salah

satunya Wireshark. Wireshark akan digunakan untuk me-monitoring datangnya data dari PC *client* ke PC *server*. Pada saat proses *transfer* dimulai, maka secara otomatis *software* Wireshark ini me-monitoring semua aktifitas yang ada pada trafik jaringan. Setelah proses *transfer* selesai, maka data dari Wireshark ini akan diproses lebih lanjut untuk di analisis dan dapat diketahui hasilnya.



Gambar 3.1 Topogi 1 dan Topologi 2

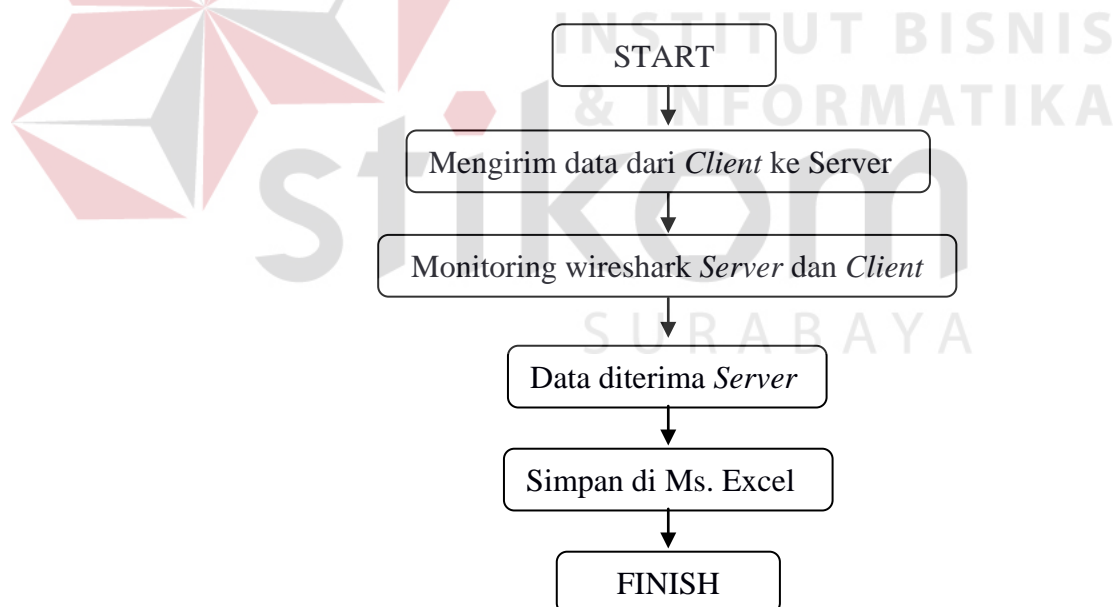
Berikut adalah alamat IP yang digunakan untuk menghubungkan antar *Router* dan PC (Personal Computer). Dapat dilihat pada tabel 3.2.

Tabel 3.2 IP Address pada Router

<i>Device</i>	<i>Interface</i>	IP Address <i>version 4</i>	IP Interface <i>Loopback</i>
Router LB	Ether 2	192.168.1.1/24	
	Ether 3	192.168.2.1/24	
	Ether 4	192.168.3.2/24	
	Loopback		10.10.10.1/24

Router A	Ether 2	192.168.3.1/24	
	Ether 3	192.168.5.2/30	
	Loopback		10.10.10.2/24
Router B	Ether 2	192.168.4.1/24	
	Ether 3	192.168.2.2/24	
	Loopback		10.10.10.3/24
Router S	Ether 2	192.168.4.2/24	
	Ether 3	192.168.6.1/24	
	Ether 4	191.168.5.1/24	
	Loopback		10.10.10.4/24
PC 1 (Server)	Port 2	192.168.1.2/30	
PC 2 (Client)	Port 3	192.168.6.2/30	

3.2.3 Proses Monitoring dan Pengambilan Data



Gambar 3.2 Flowchart proses pengambilan data

Pada gambar 3.2 dapat dilihat bahwa dari PC *Client* akan mengirimkan data ke PC *Server* melalui aplikasi *FileZilla* dan data dikirim melewati *Router* mikrotik. Sebelum data dikirim, aktifkan *wireshark* pada PC *Server* dan *Client*

yang akan me-*monitoring* pengiriman paket. Selanjutnya, data akan masuk pada PC *Server* dan *wireshark* akan memonitoring paket yang sampai dan setelah selesai, maka hasil dari monitoring tersebut disimpan ke dalam *ms.excel* untuk diproses lebih lanjut.

3.2.4 Konfigurasi Sistem

1. Konfigurasi Nama *Router*, *Interface* dan *IP Address*

Pada masing – masing *Router* diberi nama sesuai dengan topologi. Setelah itu beri nama masing – masing *interface Router* dan konfigurasi *IP Address* pada masing – masing *interface* sesuai dengan tabel 3.2.

2. Konfigurasi *Loopback Interface*

Konfigurasi *loopback interface* pada masing – masing *interface* yang ada pada *Router*.

3. Konfigurasi *Dynamic Routing*

Mengkonfigurasi *routing* dengan menggunakan *routing Open Shortest Path First (OSPF)* pada setiap *Router*.

4. Konfigurasi *Load Balance*

Konfigurasi *Load Balance* pada *Router Client (Router LB)* yang terhubung pada 2 *Router* penyedia jalur yaitu *Router A* dan *Router B*.

3.2.5 Uji Koneksi

Pada uji koneksi ini merupakan untuk melihat jaringan, apakah pada sisi *client* sudah terhubung dengan 4 *Router* dan *pc server* yang telah dikonfigurasi.

Pada gambar 3.3 uji koneksi melalui *client* ke *Router LB*, untuk mengetahui apakah *Router LB* sudah bisa terhubung dengan *client*.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=62
Reply from 192.168.1.1: bytes=32 time<1ms TTL=62
Reply from 192.168.1.1: bytes=32 time<1ms TTL=62
Reply from 192.168.1.1: bytes=32 time<1ms TTL=62

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\user>_
```

Gambar 3.3 Ping *Client* ke *Router LB*

Pada gambar 3.4 uji koneksi melalui *Client* ke *Router A* yang terhubung dengan *server* dan *Router LB*, untuk mengetahui apakah *Router A* sudah bisa terhubung dengan *client*.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>ping 192.168.3.1

Pinging 192.168.3.1 with 32 bytes of data:
Reply from 192.168.3.1: bytes=32 time<1ms TTL=63
Reply from 192.168.3.1: bytes=32 time<1ms TTL=63
Reply from 192.168.3.1: bytes=32 time<1ms TTL=63
Reply from 192.168.3.1: bytes=32 time<1ms TTL=63

Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\user>_
```

Gambar 3.4 Ping *Client* ke *Router A*

Pada gambar 3.5 uji koneksi melalui *client* ke *Router B* yang terhubung dengan *server* dan *Router LB*, untuk mengetahui apakah *Router B* sudah bisa terhubung dengan *client*.


```

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time<1ms TTL=63
Reply from 192.168.2.2: bytes=32 time<1ms TTL=63
Reply from 192.168.2.2: bytes=32 time<1ms TTL=63
Reply from 192.168.2.2: bytes=32 time<1ms TTL=63

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

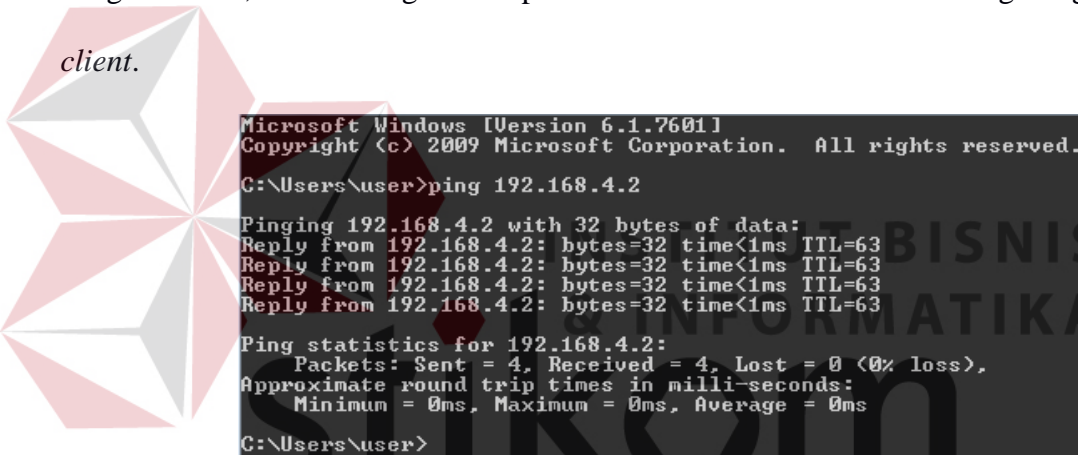
C:\Users\user>

```

Gambar 3.5 Ping *Client* ke *Router B*

Pada gambar 3.6 uji koneksi melalui *client* ke *Router S* yang terhubung dengan *server*, untuk mengetahui apakah *Router S* sudah bisa terhubung dengan

client.



```

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:
Reply from 192.168.4.2: bytes=32 time<1ms TTL=63
Reply from 192.168.4.2: bytes=32 time<1ms TTL=63
Reply from 192.168.4.2: bytes=32 time<1ms TTL=63
Reply from 192.168.4.2: bytes=32 time<1ms TTL=63

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\user>

```

Gambar 3.6 Ping *Client* ke *Router S*

Pada gambar 3.7 uji koneksi melalui *client* ke *server*, untuk mengetahui apakah *server* sudah bisa terhubung dengan *client*.

```

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>ping 192.168.6.2

Pinging 192.168.6.2 with 32 bytes of data:
Reply from 192.168.6.2: bytes=32 time<1ms TTL=64
Reply from 192.168.6.2: bytes=32 time<1ms TTL=64
Reply from 192.168.6.2: bytes=32 time<1ms TTL=64
Reply from 192.168.6.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.6.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\user>

```

Gambar 3.7 Ping *Client* ke *Server*

Setelah semua terhubung, selanjutnya adalah menyiapkan data *video* yang akan dikirim. Pada tabel 3.3 dan tabel 3.4 terdapat tiga data *video* dengan ukuran yang berbeda – beda serta *bandwidth* yang telah ditentukan, dimana nantinya akan digunakan dalam penelitian ini.

Tabel 3.3 Data *Video* dengan bandwidth 256k + 512k

Bandwidth	Ukuran <i>File Video</i>	Jumlah <i>File Video</i>
256Kbps pada jalur A 512Kbps pada jalur B	±5 MB	1 data
	±10 MB	1 data
	±15 MB	1 data

Tabel 3.4 Data *Video* dengan bandwidth 512k + 1024k

Bandwidth	Ukuran <i>File Video</i>	Jumlah <i>File Video</i>
512Kbps pada jalur A 1024Kbps pada jalur B	±5 MB	1 data
	±10 MB	1 data
	±15 MB	1 data

3.2.6 Pengolahan Data

Pada pengolahan data ini, akan menjelaskan yang dilakukan untuk mendapatkan nilai dari tiap parameter yang digunakan. Parameter tersebut meliputi *Packet Loss*, *Throughput* dan *Fairness Index*. Pertama, menjalankan aplikasi *FileZilla* pada *PC Server* dan *Client*, setelah itu menyiapkan data yang akan dikirimkan, setting *bandwidth Router*, setelah itu aktifkan *wireshark* untuk memonitoring jalannya suatu proses pengiriman data dari *Client* ke *Server*. Sebelumnya pada *wireshark* di *set* FTP-DATA agar yang tertangkap yaitu data FTP-DATA. Proses selanjutnya, menganalisa hasil yang tertangkap oleh aplikasi *wireshark* dan dihitung dengan menggunakan parameter - parameter QoS agar dapat memberikan kesimpulan pada hasil yang didapat.

3.2.7 Pengujian Sistem dan Plotting

Plotting ini menampilkan hasil dari analisis yang telah dilakukan. Berikut penjelasannya :

1. Pengujian pengiriman *Video* berdasarkan *bandwidth* yang digunakan dan ukuran *video* yang berbeda.

- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* PCC dengan *bandwidth* sebesar 256 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 512 kbps pada jalur B (*Router B*) dengan ukuran *video* 5 MB dan dilakukan plotting ke bentuk grafik.
- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* PCC dengan *bandwidth* sebesar 256 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 512 kbps pada jalur B (*Router B*) dengan ukuran *video* 10 MB dan dilakukan plotting ke bentuk grafik.
- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* PCC dengan *bandwidth* sebesar 256 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 512 kbps pada jalur B (*Router B*) dengan ukuran *video* 15 MB dan dilakukan plotting ke bentuk grafik.
- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* Nth dengan *bandwidth* sebesar 256 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 512 kbps pada jalur

B (*Router B*) dengan ukuran *video* 5 MB dan dilakukan plotting ke bentuk grafik.

- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* Nth dengan *bandwidth* sebesar 256 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 512 kbps pada jalur B (*Router B*) dengan ukuran *video* 10 MB dan dilakukan plotting ke bentuk grafik.
- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* Nth dengan *bandwidth* sebesar 256 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 512 kbps pada jalur B (*Router B*) dengan ukuran *video* 15 MB dan dilakukan plotting ke bentuk grafik.
- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* PCC dengan *bandwidth* sebesar 512 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 1024 kbps pada jalur B (*Router B*) dengan ukuran *video* 5 MB dan dilakukan plotting ke bentuk grafik.
- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* PCC dengan *bandwidth* sebesar 512 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 1024 kbps pada jalur B (*Router B*) dengan ukuran *video* 10 MB dan dilakukan plotting ke bentuk grafik..
- Melakukan analisa pada pengiriman data *video* pada *client* dan

server menggunakan metode *load balance* PCC dengan *bandwidth* sebesar 512 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 1024 kbps pada jalur B (*Router B*) dengan ukuran *video* 15 MB dan dilakukan plotting ke bentuk grafik.

- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* Nth dengan *bandwidth* sebesar 512 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 1024 kbps pada jalur B (*Router B*) dengan ukuran *video* 5 MB dan dilakukan plotting ke bentuk grafik.
- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* Nth dengan *bandwidth* sebesar 512 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 1024 kbps pada jalur B (*Router B*) dengan ukuran *video* 10 MB dan dilakukan plotting ke bentuk grafik.
- Melakukan analisa pada pengiriman data *video* pada *client* dan *server* menggunakan metode *load balance* Nth dengan *bandwidth* sebesar 512 kbps pada jalur A (*Router A*), dan *bandwidth* sebesar 1024 kbps pada jalur B (*Router B*) dengan ukuran *video* 15 MB dan dilakukan plotting ke bentuk grafik.