

BAB II

LANDASAN TEORI

2.1. Jaringan Komputer

Jaringan Komputer terdiri dari dua suku kata yaitu jaringan dan komputer, jaringan komputer adalah jaringan telekomunikasi yang memungkinkan antar komputer untuk saling bertukar data (Chandra dan Kosdiana, 2018). Jaringan memiliki media transmisi data yang terbagi menjadi dua yaitu jaringan kabel dan jaringan nirkabel. Jaringan kabel memiliki keunggulan dalam keamanan dan gangguan yang lebih sedikit serta kecepatan transfer data yang relatif lebih cepat dari nirkabel serta lebih murah, sedangkan jaringan nirkabel memiliki keunggulan mobilitas, fleksibilitas, dan mempermudah penambahan perangkat yang ingin terhubung dengan jaringan.

2.2. Internet

Interconnection-Networking atau Internet merupakan jaringan yang saling terhubung secara global dengan menggunakan paket *protocol* (TCP/IP). Internet dikembangkan pertama kali pada tahun 1969 oleh salah satu bagian dari departemen pertahanan Amerika Serikat yaitu ARPA (*Automatic Radar Plotting Aids*), saat itu internet masih dikenal dengan ARPANET (*Advance Research Project Agency Network*). ARPANET yaitu suatu badan penelitian yang dibentuk oleh DARPA (*Defence Advanced Research Project Agency*), dan merupakan “*grand-daddy of packet switching*” (Sukaridhoto, 2014). Proyek yang difungsikan sebagai sarana percobaan teknologi jaringan komputer terbaru pada zamannya, seperti teknologi *packet-switching* untuk mengirim dan menerima data.

Penggunaan teknologi *packet-switching* ini memungkinkan bit informasi dapat mengambil data dari hampir semua rute informasi dan tetap bisa berakhir di tujuan akhir yang telah di tentukan.

2.3. MikroTik Router Board

MikroTik *router board* adalah sebuah *hardware* atau perangkat keras yang sudah terpasang sistem operasi MikroTik berbasis Linux. MikroTik didesain khusus untuk memudahkan berbagai keperluan yang berkaitan dengan jaringan komputer. Terdapat banyak fitur-fitur yang dimiliki oleh MikroTik *router board* yaitu fitur *firewall*, *routing*, *QoS (Quality of Service)*, *hotspot*, *wireless*, dan masih banyak lagi fitur lainnya. Salah satu fitur yang akan menjadi penelitian ini adalah fitur *load balancing*. Awalnya MikroTik merupakan sistem operasi *router* yang di rilis dengan nama MikroTik RouterOS, kelebihan dari RouterOS ini adalah mampu dipasang pada komputer biasa, tidak seperti sistem operasi *router* lainnya yang hanya bisa dipasang pada *hardware* tertentu (Towijdojo, 2014).

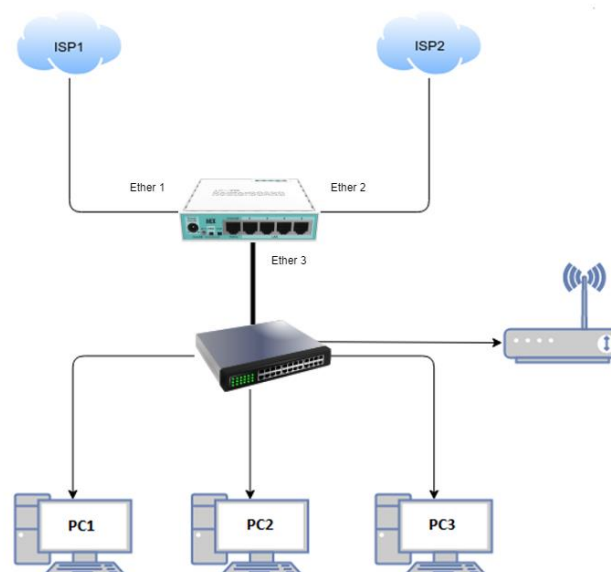
2.4. Bridge

Mode *bridge* merupakan fitur yang tersedia pada MikroTik yang berfungsi untuk menggabungkan 2 atau lebih *port ethernet* kedalam satu *network* yang sama. Mengaktifkan *bridge* pada *interface* akan menonaktifkan fungsi *routing* pada *interface* tersebut. Cara kerja mode ini adalah mengubah fungsi pada *port ethernet* atau jenis *port* lain pada *router board* yang semulanya fungsi awalnya adalah untuk *routing* dirubah fungsi menjadi *bridge* untuk menghubungkan fungsi pada *port* dan menghubungkan perangkat jaringan kedalam satu *network* yang sama. Kelemahan penggunaan mode *bridge* yaitu:

1. Sulit untuk mengatur trafik *broadcast*.
2. Permasalahan pada satu segment akan membuat masalah di semua segmen pada *bridge* yang sama.
3. Sulit untuk membuat *failover system*.
4. Sulit untuk melihat kualitas link pada tiap segmen.
5. Beban trafik pada setiap perangkat yang dilalui akan berat, karena terjadi akumulasi traffic.

2.5. Load Balancing

Load balancing adalah proses pendistribusian beban terhadap sebuah servis yang ada pada sekumpulan server atau perangkat jaringan ketika ada permintaan dari pengguna (Oktivasari dan Sanjaya, 2015). *Load balancing* diperlukan jika akses data atau arus data pada internet yang sangat padat maka server tersebut harus memiliki dua atau lebih akses internet agar dapat membagi jalur data dan tidak terbebani ataupun terfokus pada satu jalur akses internet saja.



Gambar 2.1 Topologi jaringan dengan *load balancing*

Load balancing tidak bekerja seperti perhitungan matematika biasa, dengan menggunakan *load balancing* dengan dua *link* koneksi, maka besarnya *bandwith* yang didapat bukan menjadi 2 kali lipat atau penambahan dari *bandwith* itu, dengan 2 *link* koneksi internet dari yang besarnya misalnya masing-masing 10 Mbps bukan berarti setelah diterakan *load balance bandwith* menjadi $10 + 10 = 20$, tetapi *bandwith* yang diterima menjadi $10 + 10 = 10 + 10$ atau $10 + 10 = 5 + 5 + 5 + 5$ bisa juga $10 + 10 = 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2$.

2.5.1. Metode Load Balancing

2.5.2. Per Connection Classifier

Per Connection Classifier (PCC) merupakan metode yang mengelompokkan *traffic* koneksi ke *router* menjadi beberapa kelompok. Pengelompokan ini bisa dibedakan berdasarkan *srcaddress*, *dst-address*, *src-port* dan atau *dst-port* (Pujo Dewobroto, n.d.). Dengan pengelompokan koneksi, *traffic* menggunakan atau dilewatkan dari semua *link* yang ada, router akan menyimpan informasi tentang *link gateway* yang dilewati data di setiap *traffic* koneksi, sehingga pada paket-paket selanjutnya yang masih berkaitan dengan paket data sebelumnya akan dilewatkan pada jalur *gateway* yang sama juga.

2.5.3. Nth

Nth merupakan sebuah bilangan integer (bilangan ke-n) yang diterapkan dalam suatu deret yang terdiri dari *every* dan *packet* yang akan direalisasikan dalam suatu deret *integer*, deret ini yang nantinya akan digunakan sebagai sistem antrian dalam *mangle rule* yang ditetapkan pada MikroTik. Nth dapat bekerja

dalam prinsip *per packet load balance* atau dengan *per connection load balnce* (Towijdojo, 2016).

2.6. *Failover*

Failover merupakan kemampuan suatu sistem untuk berpindah ke sistem cadangan jika sistem utama mengalami kegagalan koneksi (Dani dan Suryawan, 2017). Kemampuan *failover* ini diperlukan jika *load balancing* yang sudah di terapkan mengalami kegagalan koneksi sehingga sistem akan tetap berjalan tanpa gangguan.

2.7. *TIPHON*

TIPHON adalah standar penilaian yang dikeluarkan oleh badan standar ETSI (*European Telecommunications Standards Institue*). Standar yang menjadi pedoman *QoS* terdapat pada versi ETSI TR 101 329-7 v2.1.1. ETSI adalah Organisasi Standar Eropa (ESO), bekerja pada bidang standar regional yang bergerak dalam telekomunikasi, penyiaran, jaringan dan layanan komunikasi elektronik.

Standar TIPHON berdasarkan nilai indeks untuk parameter *troughput* tertera pada tabel 2.1.

Tabel 2.1 Parameter *Troughput* Tiphon

Nilai	<i>Troughput</i>	Indeks
Sangat Buruk	0-337kbps	0
Buruk	338-700kbps	1
Sedang	700kbps – 1,2Mbps	2
Bagus	1,2-2Mbps	3
Sangat Bagus	2,1Mbps	4

Standar TIPHON berdasarkan nilai indeks untuk parameter *delay* tertera pada tabel 2.2.

Tabel 2.2 Parameter *Delay* TIPHON

Nilai	<i>Delay</i> (ms)	Indeks
Jelek	450 >	1
Sedang	350	2
Bagus	250	3
Sangat Bagus	< 150	4

Standar TIPHON berdasarkan nilai indeks untuk parameter *jitter* tertera pada tabel 2.3.

Tabel 2.3 Parameter *Jitter* TIPHON

Nilai	<i>Jitter</i> (ms)	Indeks
Jelek	225	1
Sedang	125	2
Bagus	75	3
Sangat Bagus	0	4

Standar TIPHON berdasarkan nilai indeks untuk parameter *packet loss* tertera pada tabel 2.4.

Tabel 2.4 Parameter *Packet Loss* Tiphon

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

2.8. *Quality Of Services*

Quality of Service (QoS) adalah mengukur suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwidth*, mengatasi *jitter*

dan *delay*. Parameter *QoS* adalah *latency*, *jitter*, *packet loss*, *throughput*, *MOS*, *echo cancellation* dan *PDD*. *QoS* sangat ditentukan oleh kualitas jaringan yang digunakan.

2.8.1. *Throughput*

Throughput, adalah jumlah data yang dapat dikirim dan diterima dalam jangka waktu tertentu. Berbeda dengan *bandwidth* walaupun satuannya sama *bits per second (bps)*, tapi *throughput* lebih menggambarkan *bandwidth* yang sebenarnya pada suatu waktu dan pada kondisi dan jaringan tertentu yang digunakan untuk mengunduh suatu *file* dengan ukuran tertentu (Fahmi, 2018).

Bandwidth dan *throughput* merupakan kata yang berbeda, analogi yang paling umum digunakan untuk menggambarkan keduanya adalah menganggap *bandwidth* sebagai pipa dan *throughput* sebagai air, semakin besar pipa atau *bandwidth*, semakin banyak air atau data yang dapat mengalir melaluinya sekaligus.

Menghitung *throughput* dapat dilakukan dengan cara tp adalah *throughput*, dz adalah ukuran data yang dikirim, dan t adalah waktu yang dibutuhkan maka di dapatkan rumus berikut :

$$tp = \frac{dz}{t} \dots (1)$$

Keterangan :

tp : *Throughput*

dz : Ukuran data

t : Waktu (*time*)

2.8.2. Delay

Delay atau *latency* atau *round trip time delay*, adalah waktu yang dibutuhkan untuk sebuah paket yang dikirimkan dari suatu komputer ke komputer yang dituju. *Delay* dalam sebuah proses transmisi paket dalam sebuah jaringan komputer disebabkan karena adanya antrian yang panjang, atau mengambil rute lain untuk menghindari kemacetan pada routing (Fahmi, 2018). Penghitungan *delay* dapat dilakukan dengan aplikasi Wireshark.

2.8.3. Jitter

Jitter atau variasi *delay*, adalah variasi dari *delay* atau selisih antara *delay* pertama dengan *delay* selanjutnya. Jika variasi *delay* dalam transmisi terlalu lebar, maka akan mempengaruhi kualitas data yang ditransmisikan. Contoh dari *jitter*, misalnya hasil ping menunjukkan *delay* dengan rentang 1ms, 3ms, 5ms. Maka *jitter* dapat dihitung dengan mengurangi *delay* akhir dengan *delay* sebelumnya, seperti contoh tersebut maka hasil *jitter* adalah 5ms-2ms=3ms.

2.8.4. Packet Loss

Packet lost, adalah persentase paket yang hilang selama mentransmisikan data. Disebabkan oleh banyak faktor seperti penurunan sinyal dalam media jaringan, kesalahan perangkat keras jaringan, atau juga radiasi dari lingkungan sekitar. Cara menghitung persentase *packet loss* dapat menggunakan rumus berikut: *pl* adalah *packet lost*, *pt* adalah paket yang dikirim, dan *pr* adalah paket yang diterima, maka rumus untuk menghitung *packet lost*, adalah sebagai berikut:

$$pl = \left(\frac{pt - pr}{pt} \right) \times 100\% \dots (2)$$

Keterangan :

pl = *Packet Loss*

pt = Paket yang dikirim (*packet transfer*)

pr = Paket yang diterima (*packet recieved*)

2.9. Penelitian Terkait

Penelitian sebelumnya dengan judul kombinasi *load balancing* metode Nth dan PCC serta penerapan *failover* pada *Cloud Core Router* (Herdiana, 2017), membahas mengenai perbedaan perbandingan metode *load balancing* dengan metode Nth dan PCC serta kombinasi dengan menggabungkan metode Nth dengan PCC, parameter yang diuji yaitu pengujian *download*, *Accebility Website*, deteksi IP Address, *failover*, hasil dari penelitian tersebut adalah pertama implementasi *load balancing* metode Nth dan PCC secara bersamaan lebih baik ketimbang implementasi hanya satu metode saja baik itu metode Nth saja maupun metode PCC saja, kedua pendistribusian trafik dapat menyebar ke semua jalur koneksi, ketiga dapat mengakses website dengan intoleransi terhadap perubahan IP Address dan *session*, yang terakhir teknik *failover* dapat menjaga koneksi tetap stabil walaupun salah satu jalur koneksi bermasalah.

Penelitian sebelumnya yang berkaitan dengan *load balancing*, *failover* yang telah dilakukan dan dapat membantu dalam penelitian ini ditampilkan pada tabel 2.1. Penelitian terdekat yaitu dilakukan oleh Herdiana (2017), menjelaskan mengenai penggabungan dari dua metode *load balancing* yaitu metode PCC dan Nth, kedua metode ini mempunyai kelebihan dan kelemahan.

Tabel 2.5 Penelitian Terkait *Load Balancing*

No	Peneliti	Judul	Metode Penelitian	Parameter
1	Herdiana, 2017	Kombinasi <i>Load Balancing</i> Metode Nth dan PCC Serta Penerapan <i>Failover</i> Pada <i>cloud core router</i>	a. Merancang topologi jaringan b. Konfigurasi <i>Firewall</i> c. Pengujian	1. Aktifitas Download 2. Accessibility Website 3. Deteksi IP Address 4. <i>Failover</i>
2	Sukendar, 2017	Keseimbangan <i>Bandwidth</i> Dengan Menggunakan Dua ISP Melalui Metode Nth <i>Load Balancing</i> Berbasiskan Mikrotik	a. Studi literatur b. Observasi Eksperimental c. Instrumentasi	1. Pengujian dengan speedtes.net 2. Pengujian dengan <i>tracert</i> 3. Pengujian <i>failover</i> 4. <i>Download</i> 5. <i>Upload</i>
3	Anwar dan Nurhaida, 2019	Implementasi <i>Load Balancing</i> Menggunakan Metode Equal Cost Multi Path (ECMP) Pada Interkoneksi Jaringan	a. Studi lapangan b. Studi Pustaka c. Analisa kebutuhan d. Perancangan dan implementasi e. Eksperimen	1. <i>LOAD BALANCING</i> 2. <i>Failover</i> 3. <i>QUALITY OF SERVICE</i> (<i>Throughput, Delay, Jitter</i>)
4	Nasser dan Witono, 2016	Analisis Algoritma Round Robin, Least Connection, dan Ratio pada Load Balancng Menggunakan Opnet Modeler	a. Analisis topologi jaringan b. Simulasi skenario c. Simulasi jaringan data	1. Perbandingan Algoritma <i>Round Robin, Least Connection, Ratio</i> 2. Besaran uji da ta 1 Mbps, 5 Mbps, 7 Mbps 3. Setiap pengujian algoritma dengan parameter <i>Throughput, delay, jitter</i>

Tabel 2.5 Penelitian Terkait *Load Balancing* (Lanjutan 1)

No	Peneliti	Judul	Metode Penelitian	Parameter
5	Leisubun dkk., 2013	Analisis Kinerja <i>Load Balancing</i> Menggunakan Metode Nth Dengan Dua Internet Service Provider Pada Router Mikrotik	a. Metode pengembangan sistem (<i>analysis, design, simulation prototype, implementation, management</i>) b. Flowchart	-
6	Hakim dan Yahyo, 2019	Implementasi <i>Load Balancing</i> Menggunakan Jaringan Indihome Dan Telkomsel Pada Mikrotik Router Dengan Metode Nth	<i>mixed method</i> (Kualitatif dan kuantitatif)	1. <i>Throughput</i> 2. <i>Packet loss</i> 3. <i>Delay</i> 4. Rata-rata 2 independen <i>Throughput</i> 5. Rata-rata 2 independen <i>Packet loss</i> 6. Rata-rata 2 independen <i>Delay</i>
7	Oktivasari dan Sanjaya, 2015	Implementasi Sistem <i>Load Balancing</i> Dua ISP Menggunakan Mikrotik dengan Metode <i>Per Connection Classifier</i>	a. Deskripsi b. Cara kerja program c. Rancangan program	1. <i>Bandwith Download</i> 2. Pengujian pada sisi mikrotik 3. <i>Failover</i>
8	Leman, 2019	<i>Load Balancing</i> 2 Jalur Internet Menggunakan MikroTik Round Robin	a. Kebutuhan Analisa b. Strategi pemecahan masalah	1. <i>Throughput</i>

Tabel 2.5 Penelitian Terkait *Load Balancing* (Lanjutan 2)

No	Peneliti	Judul	Metode Penelitian	Parameter
9	Pangestu dkk., 2018	Metode Per Connection Classifier Untuk Implementasi <i>Load Balancing</i> Jaringan Internet	<ul style="list-style-type: none"> a. <i>Analysis</i> b. <i>Design</i> c. <i>Simulation prototyping</i> d. <i>Implementation</i> e. <i>Monitoring</i> f. <i>Management</i> 	<ul style="list-style-type: none"> 1. <i>Download</i> 2. <i>Upload</i> 3. <i>Failover</i>
10	Frayogi dkk., 2018	Perbandingan Kinerja RouterOS Mikrotik dan Zeroshell pada Mekanisme <i>Load Balancing</i> Serta Failover	<ul style="list-style-type: none"> a. Perancangan b. Pengujian 	<ul style="list-style-type: none"> 1. Failover 2. Delay 3. Speedtest 4. Streaming 4 film bersamaan 5. Download dengan <i>Google Chrome</i> 6. Download dengan <i>Internet Download Manager</i>
11	Syaputra & Assegaff, 2017	Analisis Dan Implementasi <i>Load Balancing</i> Dengan Metode Nth Pada Jaringan Dinas Pendidikan Provinsi Jambi	<ul style="list-style-type: none"> a. Identifikasi masalah b. Pengumpulan data c. Analisis Sistem d. Perancangan Sistem e. Pembuat Laporan Hasil Penelitian 	<ul style="list-style-type: none"> a. Ping b. Download (Mbps) c. Upload (Mbps) d. Grade
12	Wahanani et al., 2018	<i>Analysis Load Balancing With Nth Method on Web Video Streaming Using Backup Failover</i>		<ul style="list-style-type: none"> a. Download b. Video Streaming c. Failover

Tabel 2.5 Penelitian Terkait *Load Balancing* (Lanjutan 3)

No	Peneliti	Judul	Metode Penelitian	Parameter
13	Bhayangkara dan Riadi, 2014	Implementasi Proxy Server dan <i>Load Balancing</i> Menggunakan metode Per Connection Classifier (PCC) Berbasis Mikrotik (studi kasus : shmily.net)	<ul style="list-style-type: none"> a. Pengumpulan Data b. Alat Penelitian c. Analisis Data d. Perancangan e. Uji Kelayakan 	<ul style="list-style-type: none"> a. Pengujian Proxy
14	Rasna & Ashari, 2019	<i>Application Load Balancing With Nth Method Multiple Gateway Internet Networks</i>	<ul style="list-style-type: none"> a. Identifikasi b. Studi Literatur c. Merancang Topologi 	<ul style="list-style-type: none"> a. Download b. Upload c. Ping d. Rx e. Tx f. Average Throughput g. Packet Loss h. Delay
15	Warman & Andrian, 2017	Analisis kinerja <i>Load Balancing</i> dua line koneksi Dengan metode Nth (Studi Kasus : Laboratorium Teknik Informatika Institut Teknologi Padang)	NDLC (<i>Network Depelopment Life Cycle</i>) : <ul style="list-style-type: none"> a. Analysis b. Design c. Simulation Prototyping d. Implementation e. Monitoring f. Management 	<ul style="list-style-type: none"> a. Tracert

Tabel 2.6 Matriks Penelitian

No	Judul	Metode			Parameter			Keterangan
		ECMP	PCC	Nth	Implementasi	QoS	Lainnya	
1	Kombinasi <i>Load Balancing</i> Metode Nth dan PCC Serta Penerapan <i>Failover</i> Pada <i>cloud core router</i>		✓	✓			✓	Parameter yang diuji : 1. Aktifitas Download 2. <i>Accessibility Website</i> 3. Deteksi IP Address 4. <i>Failover</i>
2	Keseimbangan <i>Bandwidth</i> Dengan Menggunakan Dua ISP Melalui Metode Nth <i>Load Balancing</i> Berbasiskan Mikrotik			✓			✓	Parameter yang diuji : 1. Tracert 2. <i>Download</i> 3. <i>Upload</i>
3	Implementasi <i>Load Balancing</i> Menggunakan Metode <i>Equal Cost Multi Path</i> (ECMP) Pada Interkoneksi Jaringan	✓			✓			-
4	Analisis Algoritma <i>Round Robin</i> , <i>Least Connection</i> , dan <i>Ratio</i> pada <i>Load Balancing</i> Menggunakan Opnet Modeler					✓		Penelitian menggunakan OPNET <i>Network Simulator</i>
5	Analisis Kinerja <i>Load Balancing</i> Menggunakan Metode Nth Dengan Dua Internet Service Provider Pada Router Mikrotik			✓				Penerapan Nth pada jaringan sekolah
6	Implementasi <i>Load Balancing</i> Menggunakan Jaringan Indihome Dan Telkomsel Pada Mikrotik Router Dengan Metode Nth			✓	✓	✓		-

Tabel 2.6 Matriks Penelitian (Lanjutan 1)

No	Judul	Metode			Parameter			Keterangan
		ECMP	PCC	Nth	Implementasi	QoS	Lainnya	
7	Implementasi Sistem <i>Load Balancing</i> Dua ISP Menggunakan Mikrotik dengan Metode <i>Per Connection Classifier</i>		✓		✓			Melakukan pengujian <i>download</i>
8	<i>Load Balancing</i> 2 Jalur Internet Menggunakan MikroTik <i>Round Robin</i>			✓	✓			Menggunakan sistem LVS (Linux Virtual Server)
9	Metode <i>Per Connection Classifier</i> Untuk Implementasi <i>Load Balancing</i> Jaringan Internet	✓			✓			Parameter yang diuji : 1. <i>Tracert</i> 2. <i>Download</i> 3. <i>Upload</i>
10	Perbandingan Kinerja RouterOS Mikrotik dan Zeroshell pada Mekanisme <i>Load Balancing</i> Serta <i>Failover</i>	✓			✓			Menggunakan algoritma <i>round robin</i>
11	Analisis Dan Implementasi <i>Load Balancing</i> Dengan Metode Nth Pada Jaringan Dinas Pendidikan Provinsi Jambi			✓	✓			Parameter yang diuji : 1. <i>Packet data</i> 2. <i>Download</i> 3. <i>Upload</i>
12	<i>Analysis Load Balancing With Nth Method on Web Video Streaming Using Backup Failover</i>			✓			✓	Parameter yang diuji : 1. <i>Download</i> 2. <i>Video Stream</i>
13	Implementasi Proxy Server dan <i>Load Balancing</i> Menggunakan Metode <i>Per Connection Classifier</i> (PCC) Berbasis Mikrotik (studi kasus : <i>shmily.net</i>)		✓				✓	Perbandingan sebelum menerapkan proxy server dan setelah menerapkan proxy server

Tabel 2.6 Matriks Penelitian (Lanjutan 2)

No	Judul	Metode			Parameter			Keterangan
		ECMP	PCC	Nth	Implementasi	QoS	Lainnya	
14	<i>Application Load Balancing With Nth Method Multiple Gateway Internet Networks</i>			✓		✓		Perbandingan <i>QoS</i> sebelum diterapkan <i>Load Balancing</i> dan setelah diterapkan <i>Load Balancing</i>
15	Analisis kinerja <i>Load Balancing</i> dua line koneksi Dengan metode Nth (Studi Kasus : Laboratorium Teknik Informatika Institut Teknologi Padang)			✓	✓			-
16	<i>Quality Of Service</i> Pada Kombinasi Metode <i>Load Balancing</i> PCC dan Nth serta penerapan Faiover Pada Mikrotik		✓	✓		✓		-

Keterangan: Parameter *QoS* meliputi *Throughput*, *Delay*, *Jitter*, *Packet Loss*.