
IMPLEMENTASI METODE *NTH* DAN *FAILOVER* DENGAN FITUR NOTIFIKASI

Ade Kurnia Saputra¹, Indra Setyabudi², Dedy Hermanto³
STMIK GI MDP, Jl. Rajawali No. 14 Palembang, 0711-376400
Jurusan Teknik Informatika, STMIK GI MDP, Palembang
e-mail: ade241193@mhs.mdp.ac.id, indra.setyabudi@mhs.mdp.ac.id
dedy@mdp.ac.id

Abstrak

Dalam sebuah perusahaan tentunya harus mempunyai koneksi jaringan yang stabil agar dapat menunjang kinerja dalam sebuah perusahaan itu sendiri. *NTH* adalah sebuah fitur pada *firewall* yang digunakan sebagai penghitung (*counter*) dari paket data atau koneksi (*packet new*). *Failover* adalah teknik untuk membagi beban *traffic* di jaringan yang terlalu besar dan memindahkannya secara otomatis apabila pada jaringan tersebut mengalami kendala dalam koneksi. Penelitian ini menggunakan metode penelitian *Study Literature* dan *Action Research*. *Action Research* memiliki tahapan yaitu *Planning*, *Acting*, *Observing*, dan *Reflecting*. Penelitian ini menggunakan tiga *Internet Service Provider* dan menggabungkan teknik *load balancing* menggunakan metode *NTH*, teknik *failover*, dan menggunakan fitur notifikasi *SMS* bila salah satu koneksi terputus. Hasil pengujian *QoS* untuk *load balancing* didapat rata-rata *Delay* adalah 6.66 ms, rata-rata *Throughput* 739.123 Kbps, rata-rata *Packet loss* 8.99 %, dan hasil semua pengujian *Jitter* adalah 0 ms.

Kata Kunci : *Load Balancing*, *NTH*, *Failover*, notifikasi *SMS*, Mikrotik.

Abstract

In a company must had a stable network connection in order to support performance within a company it self. *NTH* was a feature of *firewall* used as counter of packet data or connections(*packet new*). *Failover* was a technique to divide the burden of *traffic* on a network that was too large and move it automatically if the network was experiencing constraints in the connection. This research used method of *Study Literature* and *Action Research*. *Action Research* had stages of planning, Acting, Observing, and Reflecting. This study used three *Internet Service Providers* and incorporates load balancing techniques using the method *NTH*, *failover techniques*, and used the *SMS* notification feature when one of a connection was lost. The results of *QoS* test for *load balancing* obtained average *Delay* was 6.66 ms, average *Throughput* was 739.123 Kbps, average *Packet loss* was 8.99%, and the results of all *Jitter* testing was 0 ms.

Keyword : *Load Balancing*, *NTH*, *Failover*, notification *SMS*, MikroTik.

1. PENDAHULUAN

Semakin meningkatnya kebutuhan dari penggunaan internet, tuntutan terhadap akses internet yang lebih cepat dan stabil pun semakin meningkat. Tidak menutup kemungkinan suatu sistem jaringan komputer pada suatu perusahaan menggunakan lebih dari satu ISP (*Internet Service Provider*) agar tetap bisa menjaga kualitas dari layanan internetnya. Akan tetapi, tingginya beban trafik dan *request* yang dapat ditangani oleh sistem jaringan komputer tidak jarang menimbulkan berbagai permasalahan seperti terjadinya kemacetan pada jalur koneksi internet. Permasalahan ini dapat diatasi dengan penggunaan suatu teknik jaringan yang disebut *load balancing*.

Load balance adalah teknik membagi beban (*load*) ke dalam beberapa jalur atau *link*. Penggunaan *load balancing* ini bisa dimaksimalkan dengan menggunakan teknik *Failover*. *Failover* adalah teknik untuk membagi beban *traffic* di jaringan yang terlalu besar dan memindahkannya secara otomatis apabila pada jaringan tersebut mengalami kendala dalam koneksi, dalam hal ini jaringan internet bisa terputus koneksi. Pada penelitian ini metode yang digunakan yaitu *NTH*. *NTH* sendiri merupakan suatu teknik *load balancing* yang membentuk suatu deret tertentu, yang nantinya akan digunakan sebagai suatu sistem antrian di dalam *mangle rule* yang akan dibentuk

Pada penelitian sebelumnya telah dilakukan penelitian oleh Risandy Wiratman dan Septian Ari Purnomo dengan menggunakan teknik *load balancing* *PCC* (*Per Connection Classifier*), terdapat fitur notifikasi dengan menggunakan *email*. Sebagai pembanding dari teknik *load balancing* yang akan dipakai, kali ini peneliti akan menggunakan metode yang berbeda yaitu *NTH*, dan menggunakan fitur notifikasi SMS sebagai notifikasi apabila salah satu jalur koneksi terputus atau mengalami gangguan.

Berdasarkan uraian diatas maka penulis tertarik untuk meneliti “IMPLEMENTASI METODE *NTH* DAN *FAILOVER* DENGAN FITUR NOTIFIKASI”.

2. METODE PENELITIAN

Adapun tahapan-tahapan yang dilakukan didalam penelitian ini, meliputi :

2.1 Study Literature

Pada tahap ini, peneliti mencari jurnal dan buku yang berkaitan dengan implementasi metode *NTH*, *failover*, dan fitur notifikasi *SMS*.

2.2 Action Research

Selain menggunakan metode pengumpulan data, penelitian ini juga menggunakan metode Action Research yang meliputi Planning, Acting, Observing, dan Reflecting[1].

2.3 Load Balance

Load balance dalam jaringan komputer adalah teknik membagi beban (*load*) ke dalam beberapa jalur atau *link*. Teknik ini hanya bisa dilakukan apabila network memiliki lebih dari beberapa *link*. Tujuan dari *load balance* adalah agar tidak ada *link* yang

mendapatkan beban yang berlebihan dari *link* yang lain. Teknik ini digunakan agar keseimbangan (*balance*) penggunaan *link-link* tersebut dapat tercapai[2].

2.4 Failover

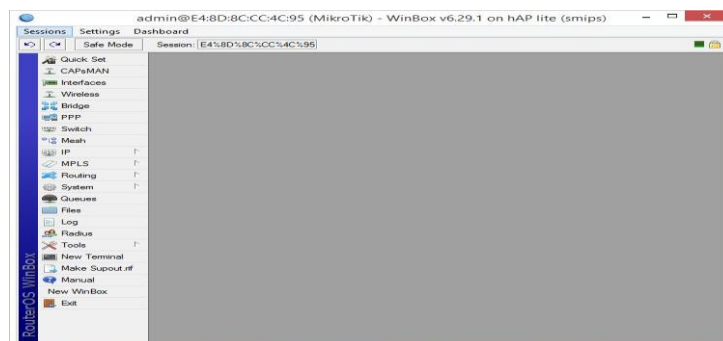
Failover adalah teknik untuk membagi beban *traffic* di jaringan yang terlalu besar dan memindahkannya secara otomatis apabila pada jaringan tersebut mengalami kendala dalam koneksi, dalam hal ini jaringan internet bisa terputus koneksi[2].

2.5 NTH

NTH adalah sebuah fitur pada *firewall* yang digunakan sebagai penghitung (*counter*) dari paket data atau koneksi (*packet new*). Untuk membagi packet sehingga dapat dilewatkan melalui beberapa link *ISP*, maka *NTH* menggunakan dua parameter, yaitu parameter *every* dan parameter *packet*. Kedua parameter ini inilah yang berfungsi sebagai penghitung dan akan digunakan *NTH* untuk menangkap sejumlah packet untuk kemudian menentukan packet beberapa yang akan di-marking dengan *action=mark-routing*[3].

2.6 Winbox

Winbox merupakan perangkat lunak yang bisa melakukan *remote* ke *router* mikrotik pada mode GUI. Untuk melakukan konfigurasi, *router* mikrotik pada umumnya menggunakan sebuah PC. Apabila menggunakan winbox, maka kita bisa melakukan konfigurasi menggunakan GUI ataupun *text*. Tampilan Winbox dapat dilihat pada Gambar 2.1 dibawah ini[4] :



Gambar 2.1 Tampilan Winbox

2.7 Wireshark

Wireshark merupakan perangkat lunak yang digunakan untuk melakukan pengawasan terhadap paket data jaringan dan menampilkan data yang ditangkap secara menyeluruh. Penggunaan wireshark bersifat gratis pada sistem operasi windows, linux dan mac karena bersifat *open source*. Tampilan Wireshark dapat dilihat pada gambar 2.2 dibawah ini[5] :



Gambar 2.2 Tampilan Wireshark

2.8 Quality of Service (QoS)

Quality of Service digunakan untuk menghitung kualitas jaringan ataupun jaminan terhadap layanan yang akan diberikan sehingga dapat diketahui kualitas layanan yang baik untuk pengguna. Parameter yang dikendalikan pada saat implementasi *Quality of Service* meliputi *Delay*, *Packet loss*, *Jitter*, dan *Trougpath*[6].

3. HASIL DAN PEMBAHASAN

3.1 Delay

Delay adalah tertundanya paket data yang pada saat dikirim ke tujuan. *Delay* dapat dilihat pada menu *Statistics*, lalu klik *Summary*. *Delay* Versi *Telecommunications and Internet Protocol Harmonization Over Network (TIPHON)* dikelompokkan menjadi empat kategori seperti pada Tabel 3.1 dibawah ini[7].

Tabel 3.1 Kategori Delay

Kategori Latency	Besar Delay
Sangat Bagus	<150ms
Bagus	150ms – 299ms
Sedang	300ms – 450ms
Jelek	>450ms

Untuk menghitung rata-rata delay dapat menggunakan rumus:

$$\text{Rata - rata Delay} = \frac{\text{total delay}}{\text{banyak pengujian}}$$

Hasil pengujian *delay* dapat dilihat pada Tabel 3.2 dibawah ini.

Tabel 3.2 Hasil Pengujian Delay

Percobaan ke-	Besar Delay	Kategori
1	5.87 ms	Sangat Bagus

2	6.53 ms	Sangat Bagus
3	6.64 ms	Sangat Bagus
4	7.03 ms	Sangat Bagus
5	7.05 ms	Sangat Bagus
6	6.73 ms	Sangat Bagus
7	6.92 ms	Sangat Bagus
8	6.89 ms	Sangat Bagus
9	6.50 ms	Sangat Bagus
10	6.44 ms	Sangat Bagus
Rata-rata Delay	6.66 ms	Sangat Bagus

3.2 Pengujian *Packet loss*

Packet loss merupakan kondisi di mana paket yang dikirimkan tidak sampai ke tujuan dikarenakan ada beberapa paket yang terbuang (*drop*), ukuran queue yang kecil, atau pun kondisi link yang kurang baik. *Packet loss* Versi *Telecommunications and Internet Protocol Harmonization Over Network (TIPHON)* dikelompokkan menjadi empat kategori seperti pada Tabel 3.3 dibawah ini[7].

Tabel 3.3. Kategori *Packet loss*

Kategori Degradasi	Persentase <i>Packet Loss</i>
Sangat Bagus	0% - 2%
Bagus	3% - 14%
Sedang	15% - 24%
Jelek	>25%

rumus untuk menghitung nilai rata-rata *packet loss*:

$$\text{Rata – rata Packet Loss} = \frac{\text{total packet loss}}{\text{banyak pengujian}}$$

Hasil pengujian berdasarkan *packet loss* dapat dilihat pada Tabel 3.4 dibawah ini.

Tabel 3.4 Hasil Pengujian *Packet Loss*

Percobaan ke-	Besar <i>Packet Loss</i>	Kategori
1	0 %	Sangat Bagus
2	0 %	Sangat Bagus
3	0 %	Sangat Bagus
4	0 %	Sangat Bagus
5	89.9 %	Jelek
6	0 %	Sangat Bagus
7	0 %	Sangat Bagus
8	0 %	Sangat Bagus
9	0 %	Sangat Bagus
10	0 %	Sangat Bagus
Rata-rata <i>packet loss</i>	8.99 %	Bagus

3.3 Pengujian *Jitter*

Jitter adalah variasi nilai *delay* yang dapat dipengaruhi oleh banyak faktor, salah satunya yaitu peningkatan trafik secara tiba-tiba. *Jitter* versi *Telecommunications and Internet Protocol Harmonization Over Network* (TIPHON) dikelompokkan menjadi empat kategori seperti pada Tabel 3.5 dibawah ini[7].

Tabel 3.5 Kategori Degradasi *Jitter*

Kategori Degradasi	Peak <i>Jitter</i>
Sangat Bagus	0ms
Bagus	1ms – 75ms
Sedang	76ms – 125ms
Jelek	126ms – 225ms

Menghitung nilai rata-rata *jitter* menggunakan rumus:

$$\text{Rata – rata } \mathit{Jitter} = \frac{\text{total } \mathit{jitter}}{\text{banyak pengujian}}$$

Hasil pengujian berdasarkan *jitter* dapat dilihat pada Tabel 3.6dibawah ini.

Tabel 3.6 Hasil Pengujian *Jitter*

Percobaan ke-	Besar <i>Jitter</i>	Kategori
1	0 ms	Sangat Bagus
2	0 ms	Sangat Bagus
3	0 ms	Sangat Bagus
4	0 ms	Sangat Bagus
5	0 ms	Sangat Bagus
6	0 ms	Sangat Bagus
7	0 ms	Sangat Bagus
8	0 ms	Sangat Bagus
9	0 ms	Sangat Bagus
10	0 ms	Sangat Bagus
Rata-rata <i>Jitter</i>	0 ms	Sangat Bagus

3.4 Pengujian *Throughput*

Throughput adalah kemampuan sebuah jaringan untuk melakukan pengiriman data. Untuk mengetahui nilai *throughput*, dapat dilihat pada menu *Statistics* lalu klik menu *Summary*[7].

Berikut rumus untuk mengkonversi nilai *throughput* dari Mbps menjadi Kbps

$$\text{Throughput(Kbps)} = \text{Throughput(Mbps)} \times 1024$$

Untuk menghitung nilai rata-rata *throughput* digunakan rumus:

$$\text{Rata – rata } \mathit{Throughput} = \frac{\text{total } \mathit{throughput}}{\text{banyak pengujian}}$$

Hasil pengujian berdasarkan *throughput* dapat dilihat pada Tabel 4.7.

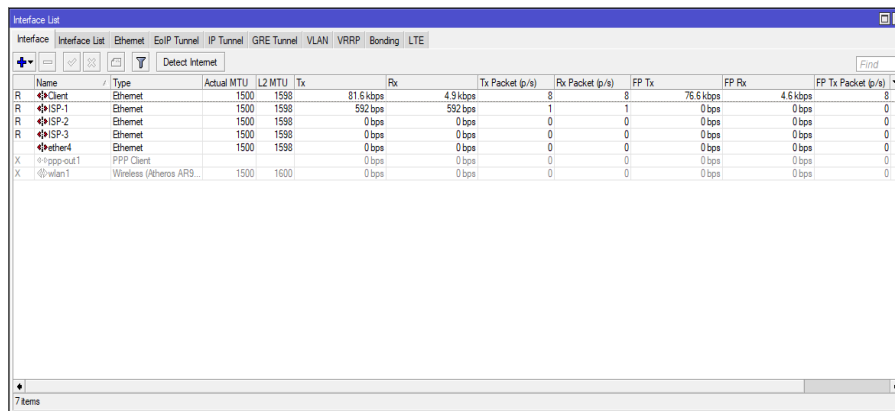
Tabel 4.7 Hasil Pengujian *Throughput*

Percobaan ke-	Besar <i>Throughput</i>
1	776.192 Kbps
2	781.312 Kbps
3	765.952 Kbps
4	708.608 Kbps
5	707.584 Kbps
6	683.008 Kbps
7	697.344 Kbps
8	714.752 Kbps
9	779.264 Kbps
10	777.216 Kbps
Rata-rata <i>Throughput</i>	739.123 Kbps

3.5 Hasil Pengujian Failover Notifikasi SMS

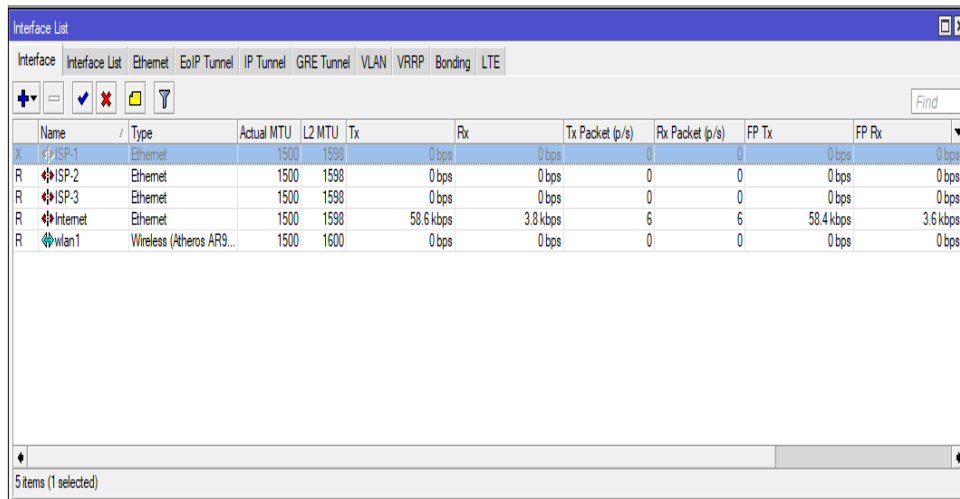
Pengujian failover dan notifikasi SMS dilakukan dengan memutus *interface* yang dianggap sebagai salah satu ISP di routerboard RB951Ui-2HnD. Berikut langkah-langkah pengujian failover notifikasi SMS.

1. Routerboard B951Ui-2HnD masih terhubung ke *Internet Service Provider*, di setiap *interface* masih dengan status *running*. Gambar 4.10 dibawah ini daftar *Interface* pada Routerboard RB941-2HnD Dalam Kondisi Semua *Interface* Belum Terputus



Gambar 4.10 Tampilan *Interface* pada Routerboard RB941-2HnD pada saat Semua *Interface* Belum Terputus

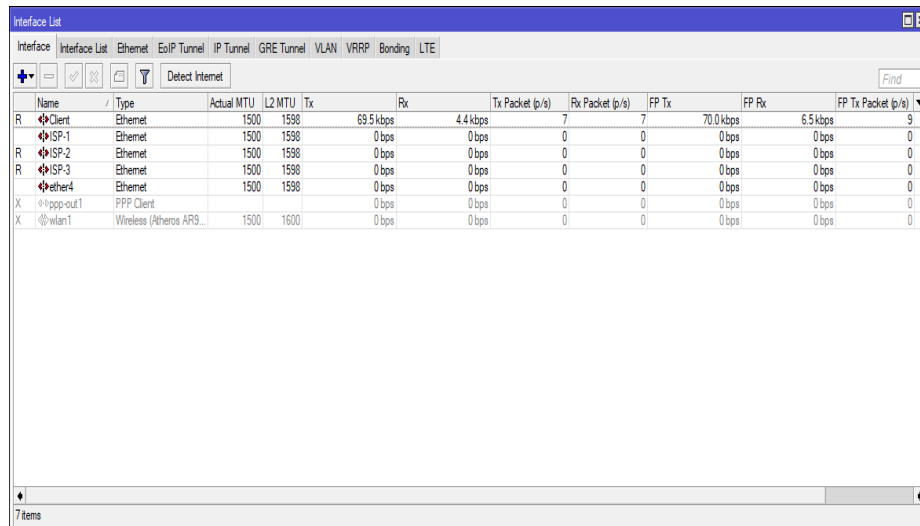
2. Selanjutnya *login* ke routerboard RB941-2nD (hAP *Lite*) klik menu *interfaces*, pilih salah satu *interface* yang dianggap sebagai salah satu *Internet Service Provider* dan putuskan koneksinya.



Name	Type	Actual MTU	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)	FP Tx	FP Rx
X ISP-1	Ethernet	1500	1538	0 bps	0 bps	0	0	0 bps	0 bps
R ISP-2	Ethernet	1500	1538	0 bps	0 bps	0	0	0 bps	0 bps
R ISP-3	Ethernet	1500	1538	0 bps	0 bps	0	0	0 bps	0 bps
R Internet	Ethernet	1500	1538	58.6 kbps	3.8 kbps	6	6	58.4 kbps	3.6 kbps
R wlan1	Wireless (Atheros AR9...	1500	1600	0 bps	0 bps	0	0	0 bps	0 bps

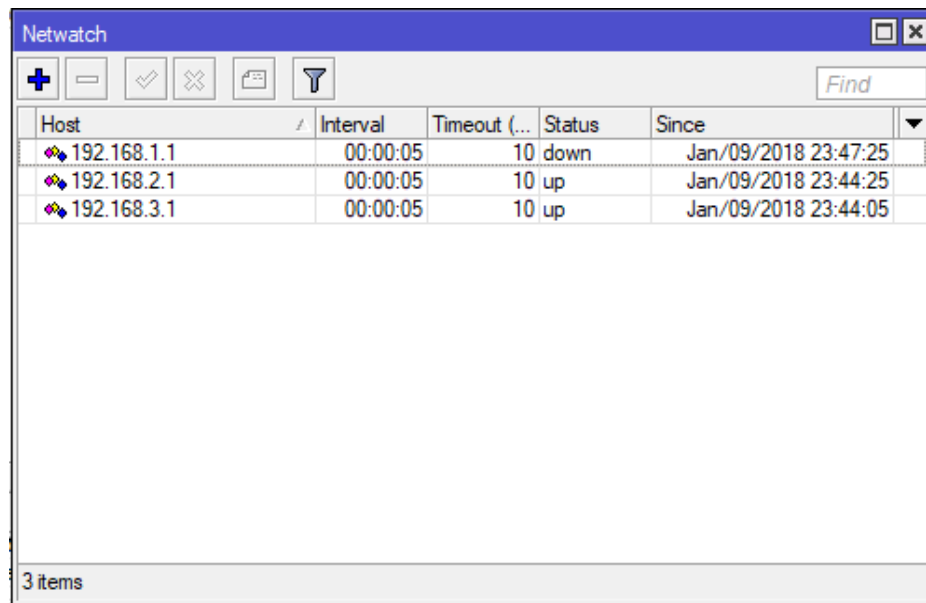
Gambar 4.11 Tampilan *Interface* Dalam Kondisi *ISP-1* Terputus pada Routerboard RB941-2Nd

3. Tampilan *interface* pada routerboard RB941-2HnD. Di kondisi ini *netwatch* menjalankan *script* notifikasi SMS seperti pada Gambar 4.12 di bawah ini.



Name	Type	Actual MTU	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)	FP Tx	FP Rx
R Client	Ethernet	1500	1538	69.5 kbps	4.4 kbps	7	7	70.0 kbps	6.5 kbps
X ISP-1	Ethernet	1500	1538	0 bps	0 bps	0	0	0 bps	0 bps
R ISP-2	Ethernet	1500	1538	0 bps	0 bps	0	0	0 bps	0 bps
R ISP-3	Ethernet	1500	1538	0 bps	0 bps	0	0	0 bps	0 bps
R ether4	Ethernet	1500	1538	0 bps	0 bps	0	0	0 bps	0 bps
X ppp-out1	PPP Client	0 bps	0 bps	0 bps	0 bps	0	0	0 bps	0 bps
X wlan1	Wireless (Atheros AR9...	1500	1600	0 bps	0 bps	0	0	0 bps	0 bps

Gambar 4.12 Tampilan *Interface* pada Routerboard RB951Ui-2HnD Saat *ISP-1* Terputus



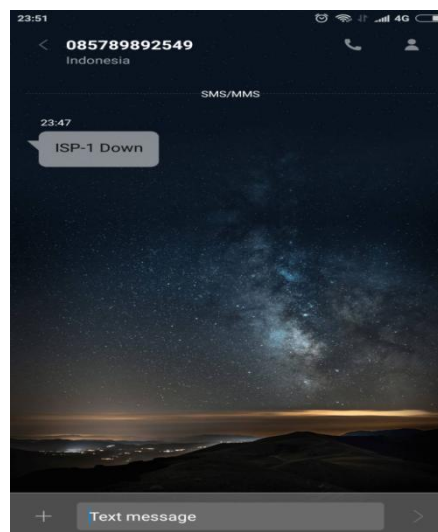
The screenshot shows the Netwatch application window. The title bar is blue and contains the text 'Netwatch'. Below the title bar is a toolbar with several icons: a plus sign, a minus sign, a checkmark, a cross, a document, and a funnel. To the right of the toolbar is a search box labeled 'Find'. The main area of the window contains a table with the following data:

Host	Interval	Timeout (...)	Status	Since
192.168.1.1	00:00:05	10	down	Jan/09/2018 23:47:25
192.168.2.1	00:00:05	10	up	Jan/09/2018 23:44:25
192.168.3.1	00:00:05	10	up	Jan/09/2018 23:44:05

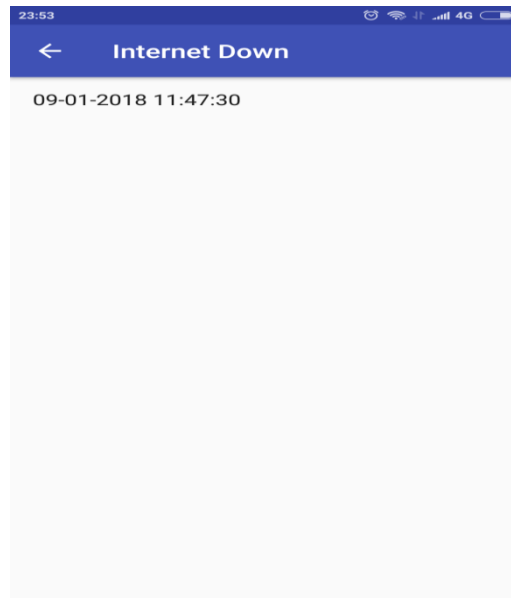
At the bottom left of the window, it says '3 items'.

Gambar 4.13 Kondisi *Netwatch* pada Saat *ISP-1* Terputus

4. Cek Pesan pada *Handphone* yang telah dijadikan sebagai tempat menampung notifikasi.



Gambar 4.14 Tampilan *Inbox SMS* pada *Handphone* Ketika Salah Satu Koneksi Terputus



Gambar 4.15 Tampilan Netlink pada Aplikasi Android Ketika Salah Satu Koneksi Terputus

5. Karena notifikasi *SMS* telah masuk. Maka pengujian *failover* notifikasi *SMS* telah berhasil.

3 KESIMPULAN

Berdasarkan hasil penelitian “Implementasi metode *NTH* dan *Failover* dengan fitur notifikasi” yang telah dibahas pada bab sebelumnya. Sehingga, dapat di tarik kesimpulan sebagai berikut:

1. Penerapan metode *NTH* dapat membagi beban *traffic* dengan seimbang pada ISP 1, ISP 2, ISP 3, dan menghindari *overload*. Dari hasil pengujian QoS untuk *load balancing* didapat rata-rata *delay* sebesar 6.66 ms, rata-rata *Throughput* sebesar 739.123 Kbps, rata-rata *Packet loss* sebesar 8.99 %, dan rata-rata *Jitter* sebesar 0 ms.
2. Dengan diterapkannya metode *failover*, terputusnya koneksi internet dapat di hindari. Karena metode *failover* dapat memindahkan ISP yang mengalami gangguan ke ISP lain secara otomatis dengan waktu *delay* yang pendek.
3. Admin dapat mengetahui dan memantau koneksi ISP yang terputus dari notifikasi yang masuk.

5. SARAN

Berdasarkan kesimpulan yang telah diuraikan dari hasil pengujian, maka diberikan saran untuk pengembangan sistem ini :

1. Penelitian dilakukan menggunakan metode *load balancing* dan *failover* yang berbeda dengan penelitian yang sebelumnya dilakukan.
-

DAFTAR PUSTAKA

- [1] Arikunto, Suharsimi 2010, *Prosedur Penelitian Suatu Pendekatan Praktik*, PT Rineka Cipta, Jakarta.
 - [2] Towidjojo, Rendra 2013, *Mikrotik Kung Fu Kitab 2*, Jasakom, Jakarta.
 - [3] Towidjojo, Rendra 2016, *Mikrotik Kung Fu Kitab 4*, Jasakom, Jakarta.
 - [4] Delhendro.com 2012, *Pengertian dan Fungsi Winbox*, Diakses 11 Maret 2016, dari <http://www.delhendro.com/>
 - [5] Kurniawan, A. 2012, *Network Forensics : Panduan Analisis & Investigasi Paket Data Jaringan Menggunakan Wireshark*, Andi, Yogyakarta
 - [6] Towidjojo, Rendra 2014, *MikroTik Kung Fu Kitab Manajemen Bandwidth Kitab 1, 2, 3*, Jasakom, Jakarta.
 - [7] ETSI 1999-06, *Telecommunications and Internet Protocol Harmonization Over Network(TIPHON) General aspects of Quality of Service(QoS)101 329 V2.1.1*, ETSI, France
-

