

# LOAD BALANCING PADA STATIC ROUTING GUNA MENINGKATKAN KINERJA ROUTER YANG SEBELUMNYA MENGUNAKAN ROUTING INFORMATION PROTOCOL (RIP)

Yusriel Ardian  
Wiji Setyaningsih

<sup>1</sup>Sistem Informasi, Universitas Kanjuruhan Malang, [acilnet@yahoo.com](mailto:acilnet@yahoo.com)

<sup>2</sup>Sistem Informasi, Universitas Kanjuruhan Malang, [wiji\\_setiya@yahoo.co.id](mailto:wiji_setiya@yahoo.co.id)

## ABSTRAK

*Routing Information Protocol (RIP) adalah salah satu protokol routing yang banyak digunakan pada proses routing. Protokol routing memiliki karakteristik yang sama dengan protokol routing lainnya yaitu melakukan advertise pada router terdekatnya untuk update database tabel routing-nya dalam interval waktu setiap 30 detik. Hal tersebut di atas dapat dilakukan oleh setiap router, apabila semakin banyak router yang digunakan maka waktu yang dibutuhkan untuk proses update juga semakin besar. NTH merupakan salah satu metode Load Balancing ber-algoritma Round Robin yang digunakan oleh sistem operasi Mikrotik, dimana metode ini menjamin pembagian beban paket data terhadap beberapa koneksi akan adil dan merata. Karena static routing tidak melakukan advertise seperti pada RIP, maka dengan ditambahkan metode NTH Load Balancing dapat meningkatkan kinerja jaringan dalam pengiriman datanya. Pengujian penelitian menggunakan metode pengukuran secara kuantitas tentang seberapa baik performance jaringan. Variabel pengukuran performance jaringan antara lain Troughput (Downstream dan Upstream) dan waktu (DownLoad dan UpLoad). Metode penelitian dilakukan menggunakan metode ekperimental kategori Pre Experimental, dengan melakukan pengujian langsung di laboratorium dengan membandingkan antara RIP dan Load Balancing dengan topologi dan kondisi yang sama.*

*Pada penelitian ini juga dihasilkan bahwa nilai troughput dan waktu baik downLoad dan upLoad dipengaruhi oleh ketepatan dalam menginisialisasi parameter Every, Counter dan Packet pada NTH, jumlah koneksi terhadap Router yang akan di-Load Balancing-kan, jumlah server beserta posisi server apakah berada pada satu Segment jaringan atau tidak.*

**Kata Kunci:** Network, RIP, NTH, Load Balancing, Every, Counter, Packet , Troughput, Downstream, Upstream.

## ABSTRACT:

*Routing Information Protocol (RIP) routing Protocol is one that is widely used in the routing process. Routing Protocols have the same characteristics with other routing Protocols that do advertise on the router closest to update its routing table database in the time interval every 30 seconds. Neither of the above can be performed by each router, if more and more routers are used then the time required for the update process will also increase. NTH is one method of Load Balancing her Round Robin algorithm is used by the operating system Mikrotik, where this method ensures the data*

*packet Load sharing to multiple connections to be fair and equitable. Because static routing does not perform as advertised in the RIP, then with added NTH Load Balancing methods can improve network performance in data delivery. Research testing using the method of measurement in quantities about how well the network performance. Network performance measurement variables such as Troughput and time.*

*Methods of research carried out using experimental methods Experimental Pre categories, with direct testing in the laboratory by comparing between RIP and Load Balancing with the same topology and conditions*

*In this research also produced that value troughput and both downLoad and upLoad time are influenced by the precision of the parameter initializing Every, and Packet Counter at NTH, the number of connections to the Router that will be the Load Balancing, the number of servers and their position is on a single server is network segment or not.*

**Keywords :** *Network, RIP, NTH, Load Balancing, Every, Counter, Packet , Troughput, Downstream, Upstream*

## 1. Pendahuluan

Pengetahuan tentang *Load Balancing* pada dasarnya adalah untuk mengetahui bagaimana memanfaatkan *server* secara efektif, sehingga jaringan-jaringan tersebut dapat bekerja dengan baik. Fungsi utama *Load Balancing* adalah bagaimana menyeimbangkan beban paket data yang lewat pada suatu *router* dapat dibagi menjadi ke beberapa koneksi yang tersedia secara adil dan merata, sehingga performance jaringan komputer dalam hal pengiriman data baik *downLoad* dan *upLoad* dapat lebih baik. Sedangkan NTH merupakan salah satu metode *Load Balancing* ber-algoritma *Round Robin* yang digunakan oleh sistem operasi *Mikrotik*, dimana metode ini menjamin pembagian beban paket data terhadap beberapa koneksi akan adil dan merata.

*Routing Information Protocol* (RIP) adalah salah satu protokol *routing* yang banyak digunakan pada proses *routing*. Protokol *routing* memiliki karakteristik yang sama dengan protokol *routing* lainnya seperti OSPF, EIGRP, BGP dan sebagainya. RIP juga melakukan *advertise* pada *router* terdekatnya untuk *update database* tabel *routing*-nya dalam interval waktu setiap 30 detik. Hal tersebut di atas dapat dilakukan oleh setiap *router*, apabila semakin banyak *router* yang digunakan maka waktu yang dibutuhkan untuk proses *update* juga semakin besar.

Hal ini juga yang menjadi ide dimana penelitian ini mencoba memanfaatkan metode *Load Balancing* yang digunakan pada statik *routing* guna meniadakan proses aktifitas *advertise* untuk meng-*update database* informasi jarak kepada tiap interface *router* seperti yang terjadi pada jaringan komputer dimana *router*-nya menggunakan RIP dengan tidak mengorbankan, *throughput* dan waktu *downLoad* maupun waktu *upLoad* yang diterima oleh *user*.

Dari penjelasan tentang *Load Balancing* dan RIP diatas penulis mencoba

melakukan penelitian bagaimana metode *Load Balancing* dijalankan pada *static routing* terhadap jaringan komputer yang berbasis *Protocol routing* RIP. Diharapkan dengan *resource* yang sama saat menggunakan RIP akan diganti dengan statik *routing* yang ditambahkan metode *Load Balancing* dengan harapan *Network Performance* akan meningkat sehingga kualitas *bandwidth* yang diterima oleh *end user* juga semakin baik.

## 2. Tinjauan Pustaka

### 2.1. Load Balancing

*Load balancing* atau penyeimbangan beban dalam jaringan sangat penting bila skala dalam jaringan komputer makin besar demikian juga *traffic data* yang ada dalam jaringan komputer makin lama makin tinggi. Layanan *Load Balancing* dimungkinkan pengaksesan sumber daya dalam jaringan didistribusikan ke beberapa host lainnya agar tidak terpusat sehingga unjuk kerja jaringan komputer secara keseluruhan bisa stabil. Ketika sebuah server sedang diakses oleh para pengguna, maka sebenarnya server tersebut sebenarnya sedang terbebani karena harus melakukan proses permintaan kepada para penggunanya. Jika penggunanya banyak maka prosesnyapun banyak. *Session-session* komunikasi dibuka oleh server tersebut untuk memungkinkan para pengguna menerima servis dari server tersebut. Jika satu server saja terbebani, tentu server tersebut tidak bisa banyak melayani para penggunanya karena kemampuan melakukan *processing* ada batasnya. Solusi yang paling ideal adalah dengan membagi-bagi beban yang datang ke beberapa server. Jadi yang melayani pengguna tidak hanya terpusat pada satu perangkat saja. Teknik ini disebut Teknik *Load Balancing*. [1]

Adapun manfaat dari *Load Balancing*:

- o Menjamin Reliabilitas layanan berarti kepercayaan terhadap sebuah sistem untuk dapat terus melayani pengguna

dengan sebaik-baiknya. Jaminan realibilitas memungkinkan pengguna dapat melakukan pekerjaan sebaik-baiknya dengan lancar melalui layanan tersebut.

- o Skalabilitas dan ketersediaan Jika dalam sebuah jaringan komputer jika hanya terdapat satu buah server mempunyai pengertian terdapat satu titik masalah. Seandainya tiba-tiba server itu mati maka layanan terhadap pengguna akan terganggu. Dengan melakukan penambahan server dan membentuk server farm maka skalabilitas akan meningkat dan selain itu faktor ketersediaan juga akan meningkat. [1]

## 2.2. Algoritma Round Robin

Konsep dasar dari algoritma ini adalah dengan menggunakan time-sharing. Pada dasarnya algoritma ini sama dengan FCFS, hanya saja bersifat preemptive. Setiap proses mendapatkan waktu CPU yang disebut dengan waktu quantum (quantum time) untuk membatasi waktu proses, biasanya 1-100 milidetik. Setelah waktu habis, proses ditunda dan ditambahkan pada ready queue. Jika suatu proses memiliki CPU burst lebih kecil dibandingkan dengan waktu quantum, maka proses tersebut akan melepaskan CPU jika telah selesai bekerja, sehingga CPU dapat segera digunakan oleh proses selanjutnya. Sebaliknya, jika suatu proses memiliki CPU burst yang lebih besar dibandingkan dengan waktu quantum, maka proses tersebut akan dihentikan sementara jika sudah mencapai waktu quantum, dan selanjutnya mengantri kembali pada posisi ekor dari ready queue, CPU kemudian menjalankan proses berikutnya. Jika terdapat  $n$  proses pada ready queue dan waktu quantum  $q$ , maka setiap proses mendapatkan  $1/n$  dari waktu CPU paling banyak  $q$  unit waktu pada sekali penjadwalan CPU. Tidak ada proses yang menunggu lebih dari  $(n-1)q$  unit

waktu. Performansi algoritma round robin dapat dijelaskan sebagai berikut, jika  $q$  besar, maka yang digunakan adalah algoritma FIFO, tetapi jika  $q$  kecil maka sering terjadi context switch. [2]

## 2.3. Routing Information Protocol (RIP)

Routing Information Protocol (RIP) mengirim routing table yang lengkap ke semua interface yang aktif setiap 30 detik. atau lebih cepat jika terdapat triggered updates. Jika dalam 180 detik sebuah route tidak diperbarui, router menghapus entry route tersebut dari forwarding table. RIP tidak memiliki informasi tentang subnet setiap route. Router harus menganggap setiap route yang diterima memiliki subnet yang sama dengan subnet pada router itu. Dengan demikian, RIP tidak mendukung Variable Length Subnet Masking (VLSM) RIP hanya menggunakan jumlah hop untuk menentukan cara terbaik ke sebuah network remote, tetapi RIP secara default memiliki sebuah nilai jumlah hop maksimum yg diizinkan, yaitu 15, berarti nilai 16 tidak terjangkau (unreachable). RIP bekerja baik pada jaringan kecil, tetapi RIP tidak efisien pada jaringan besar dengan link WAN atau jaringan yang menggunakan banyak router. [3]

RIP karakteristik:

1. Open Protocol, Stabil
2. Menggunakan Vektor Jarak/ Distance Vector Routing Protocol
3. Menghitung Jumlah Lompatan/ HOP
4. Jumlah Hop Maksimum sebanyak 15 selebihnya akan diabaikan
5. Waktu update 30 detik
6. Menggunakan UDP port 520
7. Administrative distance adalah 120.
8. Tidak selalu memilih rute tercepat
9. Menghasilkan Traffict jaringan dengan update [3]

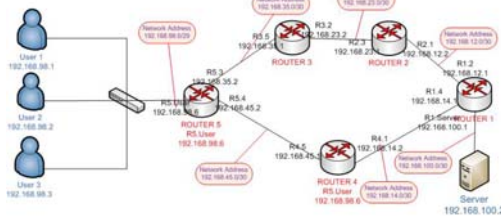
## 3. Pembahasan

Pengujian dilakukan dengan tidak mengubah kondisi lingkungan pada saat pengujian RIP yang telah dilakukan

sebelumnya, baik itu lokasi, topologi, jenis/ tipe dan spesifikasi *router router*, Sistem operasi, perangkat lunak yang digunakan serta sisi computer client dan server baik dari sisi jumlah dan spesifikasinya, sehingga diharapkan hasil dari percobaan ini dapat menjadi tolak ukur yang relatif sama apabila diterapkan dimana saja.

Tolak ukur dari keberhasilan penelitian ini yaitu minimal salah satu dari variable yang dibandingkan mendapatkan hasil sesuai dengan hipotesa yang telah ditetapkan. Adapun variable yang menjadi tolak ukur keberhasilan penelitian ini adalah waktu *DownLoad* dan *UpLoad* satuannya *second* (ms) dan *Troughtput Downstream* dan *Upstream* satuannya KBps.

Pengujian penelitian ini menggunakan 4 topologi yang berbeda, perbedaan topologi dipengaruhi oleh jumlah koneksi yang digunakan oleh *Load Balancing*, jumlah *Router*, dan jumlah dan posisi server dalam jaringan.



**Gambar 3.1**  
Topologi pengujian ke 2

Gambar 3.1 merupakan Topologi yang akan digunakan sebagai obyek penelitian NTH *Load Balancing* pada *Static Routing* Untuk Meningkatkan Kinerja Jaringan Berbasis *Routing Information Protocol* (RIP) menggunakan Sistem Operasi *Mikrotik*.

Langkah ini dilakukan pengujian terhadap metode *Routing Protocol* RIP yang digunakan. Jika dilihat pada topologi Gambar 5.4 diatas, berdasarkan teori RIP, paket yang menuju ke computer *server*

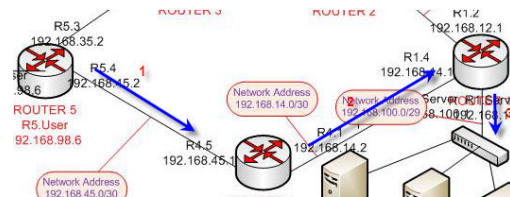
(*Router 1*) akan berjalan dari *Router 5* → *Router 4* → *Router 1* → *Server*.

**Tabel 3.1**  
**Script RIP pada Router 1**

```
(1) /Routing rip interface add interface=all receive=v2 send=v2
(2) ip address add address=192.168.12.0/30 interface=R1.2
    ip address add address=192.168.14.0/30 interface=R1.4
    ip address add address=192.168.12.100/30 interface=R1.Server1
    ip address add address=192.168.12.101/30 interface=R1.Server2
```

Tabel 3.1 adalah *script* dari inialisasi RIP *Routing*, penjelasannya sebagai berikut :

- Pada baris ke 1 **“interface add interface=all”** menentukan *interface* yang akan menggunakan RIP, sedangkan **“receive=v2 send=v2”** menentukan versi RIP yang akan digunakan.
- Pada baris ke 2, **“ip address add address”** menentukan *Network Address* yang akan menggunakan metode RIP untuk menentukan *Routing table*-nya.



**Gambar 3.2**  
Visualisasi aliran paket dari *Router 1* – *Server2*

Gambar 3.2 menunjukkan arah jalur paket yang telah ditentukan oleh RIP, yaitu mengambil jalur yang terpendek (2 hop), daripada jalur yang melalui *Router 3* – *Router 2* – *Router 5* (3 hop).

```
Tracing route to 192.168.101.2 over a maximum of 30 hops:
  0  0 ms  <1 ms  <1 ms  192.168.98.6
  1  5 ms  <1 ms  <1 ms  192.168.45.1
  2 10 ms  2 ms   4 ms  192.168.14.1
  3  5 ms  4 ms   2 ms  192.168.101.2
  4  6 ms  3 ms   4 ms  192.168.101.2
Trace complete.
```

**Gambar 3.3**  
Hasil *tracert* dari masing-masing *user*

Dari gambar 3.3 membuktikan bahwa semua komputer *user* (*user1* dan *user2*) koneksi/ jalur paket dari *Router 5* –

Router 1 melalui Router 4, jadi jalur Routingnya adalah : Router 5 (192.168.45.2) – Router 4 (192.168.45.1) – Router 1 (192.168.14.1), sesuai dengan visualisasi diatas yaitu melakukan lompatan sebanyak 2.

### 3.1 Pengujian Metode NTH Load Balancing

Langkah ini dilakukan pengujian terhadap metode NTH Load Balancing pada Static Routing yang digunakan.

Tabel 3.2 dibawah ini menjelaskan script dari inisialisasi RIP Routing,

**Tabel 3.2**  
**Script NTH Load Balancing**

```
(1) add chain=preRouting src-address-list=R1 in-interface=Local
action=mark-connection new-connection-mark=R1 passthrough=yes

(2) add chain=preRouting src-address-list=R2 in-interface=Local
action=mark-connection new-connection-mark=R2 passthrough=yes

(3) add chain=preRouting in-interface=Local connection-state=new nth=2,1
action=mark-connection new-connection-mark=R1 passthrough=yes

(4) add chain=preRouting in-interface=Local connection-state=new nth=2,2
action=mark-connection new-connection-mark=R2 passthrough=yes

(5) add chain=srcnat connection-mark=odd action=src-nat to-
addresses=192.168.53.2 tc-ports=0-65535

(6) add chain=srcnat connection-mark=even action=src-nat to-
addresses=192.168.45.2 tc-ports=0-65535

(7) add dst-address=0.0.0.0 gateway=192.168.53.1 scope=30 target-scope=10
Routing-mark=R1

(8) add dst-address=0.0.0.0 gateway=192.168.45.1 scope=30 target-scope=10
Routing-mark=R2
```

Untuk NTH Load Balancing pada Static Routing jalur data akan melalui dua koneksi sekaligus (koneksi terpakai semua) dengan pembagian paket secara merata.

```
Tracing route to 192.168.100.2 over a maximum of 30 hops
  0  4 ms  <1 ms  <1 ms  192.168.98.6
  1  12 ms  1 ms  1 ms  192.168.35.1
  2  6 ms  1 ms  2 ms  192.168.23.1
  3  7 ms  2 ms  2 ms  192.168.12.1
  4  7 ms  4 ms  4 ms  192.168.100.2
Trace complete.
```

**Gambar 3.4**  
**Hasil tracert dari user1**

Dari gambar 3.4 membuktikan bahwa komputer user user1 jalur paket dari Router 5 – Router 1 melalui Router 3 dan Router 2, jadi jalur Routingnya adalah : Router 5 (192.168.98.6) – Router 3

(192.168.35.1) – Router 2 (192.168.23.1) – Router 1 (192.168.12.1).

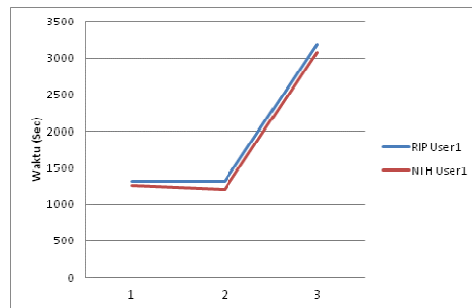
```
Tracing route to 192.168.101.2 over a maximum
  0  5 ms  <1 ms  <1 ms  192.168.98.6
  1  10 ms  2 ms  4 ms  192.168.45.1
  2  5 ms  4 ms  2 ms  192.168.14.1
  3  6 ms  3 ms  4 ms  192.168.101.2
Trace complete.
```

**Gambar 3.5**  
**Hasil tracert dari user2**

Dari gambar 3.5 membuktikan bahwa komputer user user2 jalur paket dari Router R5 – Router R1 melalui Router R4, jadi jalur Routingnya adalah : Router R5 (192.168.98.6) – Router 4 (192.168.45.1) – Router R1 (192.168.14.1).

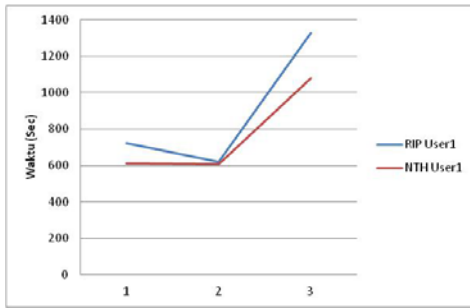
### 3.2 Pengujian Waktu

Pengujian yang dilakukan untuk mendapatkan nilai waktu dari waktu download yang dilakukan oleh dua metode yang berbeda yaitu antara Routing Protocol RIP dan NTH Load Balancing pada Static Routing, dapat dihasilkan sebagai berikut :



**Gambar 3.6**  
**Grafik perbandingan waktu download pada user2**

Dari gambar 3.6 waktu yang diperlukan untuk aktifitas Download antara Router yang hanya menggunakan RIP dan NTH Load Balancing pada Static Routing di komputer user2 menunjukkan rata-rata waktu download yang lebih baik pada Router yang menggunakan NTH Load Balancing pada Static Routing.



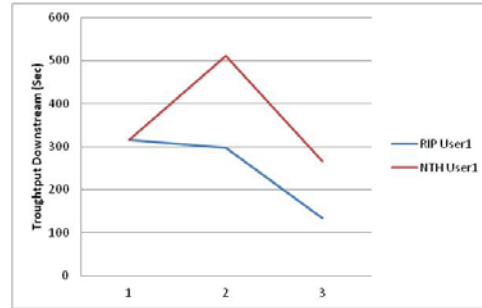
**Gambar 3.7**  
**Grafik perbandingan Waktu Upload pada user1**

Dari gambar 7 waktu yang diperlukan untuk aktifitas *Upload* antara Router yang hanya menggunakan RIP dan NTH Load Balancing pada *Static Routing* di komputer *user1* menunjukkan rata-rata waktu *downLoad* yang lebih baik pada Router yang menggunakan NTH Load Balancing pada *Static Routing*

### 3.3 Pengujian Troughput

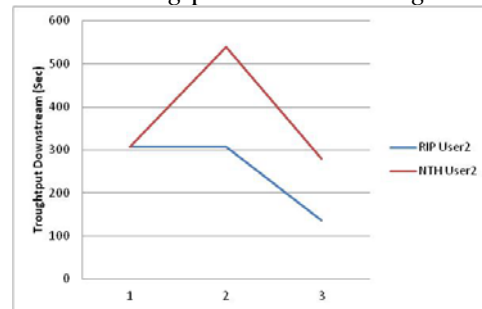
Pengujian ini untuk mendapatkan hasil *Throughput* dari kegiatan *downLoad* (*downstream*) dan *upLoad* (*Upstream*) yang dilakukan oleh masing-masing user (*User1* dan *User2*) dengan variasi beban file terhadap dua *Server* secara bersamaan. Satuan dari variable *Throughput* ini adalah *byte per second* (bps).

Dari beberapa pengujian yang dilakukan untuk mendapatkan nilai *Throughput downstream* pada aktifitas *downLoad* yang dilakukan oleh dua metode yang berbeda yaitu antara *Routing Protocol RIP* dan *NTH Load Balancing* pada *Static Routing*, dapat dihasilkan table sebagai berikut



**Gambar 3.8**  
**Grafik perbandingan Downstream pada user1**

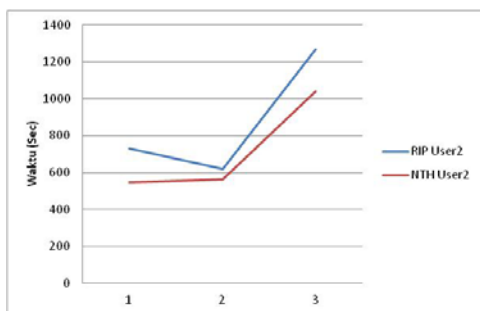
Dari gambar 3. 8 nilai *downstream* antara Router yang hanya menggunakan RIP dan NTH Load Balancing pada *Static Routing* di komputer *user1* menunjukkan rata-rata nilai *downstream* yang lebih baik pada Router yang menggunakan NTH Load Balancing pada *Static Routing*



**Gambar 3.9**  
**Grafik perbandingan Downstream pada user2**

Dari gambar 9 nilai *downstream* antara Router yang hanya menggunakan RIP dan NTH Load Balancing pada *Static Routing* di komputer *user1* menunjukkan rata-rata nilai *downstream* yang lebih baik pada Router yang menggunakan NTH Load Balancing pada *Static Routing*





**Gambar 3.10**  
**Grafik perbandingan Waktu**  
**Upstream pada user2**

Dari gambar 3.10 waktu yang diperlukan untuk aktifitas *UpLoad* antara Router yang hanya menggunakan RIP dan NTH Load Balancing pada *Static Routing* di komputer *user2* menunjukkan rata-rata waktu *downLoad* yang lebih baik pada Router yang menggunakan NTH Load Balancing pada *Static Routing*.

Sedangkan pengujian yang dilakukan untuk mendapatkan nilai *Throughput Upstream* pada aktifitas *upLoad* yang dilakukan oleh dua metode yang berbeda yaitu antara *Routing Protocol RIP* dan NTH Load Balancing pada *Static Routing*, dapat dihasilkan table sebagai berikut :

**Tabel 6**  
**Summary Pengujian Upstream**

#Pengujian	RIP (KBps)		NTH (KBps)	
	User1	User2	User1	User2
1	697.9	709.4	707.9	701.6
2	651	671.9	750	755.3
3	690.4	714.2	820.1	856.3

Tabel 6 diatas menunjukkan nilai perbandingan *throughput Upstream* antara NTH Load Balancing pada *static routing*.

#### 4. Kesimpulan

Dari beberapa pengujian waktu dan *Throughput* dari aktifitas *DownLoad* dan *UpLoad* terhadap dua metode antara RIP

dan NTH Load Balancing pada *Static Routing* pada Router menghasilkan :

1. Pada kondisi lingkungan yang sama (Topologi, jumlah koneksi, jumlah router dan jumlah server), perbandingan antara RIP dan NTH Load Balancing pada *static routing* menghasilkan :
  - a. Waktu *DownLoad* NTH Load Balancing pada *Static Routing* lebih baik 5% dari pada RIP
  - b. Waktu *UpLoad* NTH Load Balancing pada *Static Routing* lebih baik 5% dari pada RIP
  - c. *Throughput Downstream* NTH Load Balancing pada *Static Routing* lebih baik 35% dari pada RIP
  - d. Waktu *Upstream* NTH Load Balancing pada *Static Routing* lebih baik 9% dari pada RIP
2. NTH Load Balancing pada *static routing* akan menghasilkan peningkatan performance terhadap waktu dan *Throughput* apabila nilai *integer* pada parameter NTH adalah *Every* = jumlah koneksi -1, *Counter* = antara bilangan *integer* 0 – 15 dan *Packet* dimulai dengan bilangan 0 sampai jumlah sama dengan *Every*.
3. NTH Load Balancing pada *static routing* akan menghasilkan peningkatan performance terhadap waktu dan *Throughput* apabila jumlah jalur koneksi semakin banyak.
4. NTH Load Balancing pada *static routing* akan menghasilkan peningkatan performance terhadap waktu dan *Throughput* apabila jumlah server yang digunakan lebih dari satu dan posisi router berada pada router jalur router yang berbeda.

#### Daftar Pustaka

1. Iwan Rijayana, *Teknologi Load Balancing Untuk Mengatasi Beban Server*, Jurusan Teknik Informatika, Universitas Widyatama Jalan Cikutra 204 A Bandung., Seminar Nasional



Aplikasi Teknologi Informasi 2005  
(SNATI 2005), Yogyakarta, 18 Juni  
2005, ISBN: 979-756-061

2. Irfan Darmawan, Kuspriyanto, Yoga Priyana (2009), Perancangan Algoritma Load Balancing pada Topologi Dynamic Tree Jaringan Grid Computing, SNATI, Yogyakarta, ISSN: 1907-5022
3. Sandra Sendra, Pablo A. Fernández, Miguel A. Quilez and Jaime Lloret. 2010. Study and Performance of Interior Gateway IP *Routing Protocols*. Dong (Don) Xu. <http://www.sfu.ca/~donx/>