

Meningkatkan *Troughput Bandwidth* sekaligus sebagai Jalur *Failover* dengan menggunakan Metode Bonding pada MikroTik

## MENINGKATKAN TROUGHPUT BANDWIDTH SEKALIGUS SEBAGAI JALUR FAILOVER DENGAN MENGGUNAKAN METODE BONDING PADA MIKROTIK

**Khoirotun Nikmah**

D3 Manajemen Informatika, Fakultas Teknik, Universitas Negeri Surabaya, khoirotunnikmah20@gmail.com

**Agus Prihanto**

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya, agusprihanto@unesa.ac.id

### Abstrak

Banyaknya pengguna jaringan LAN yang semakin meningkat, ketika jaringan mengalami kendala bisa membuat pengguna tidak nyaman, selain itu bisa mempengaruhi *throughput bandwidth* yang akan didapat, apalagi jika *bandwidth*nya kecil maka koneksi akan menjadi lambat, dan terkadang jaringan mengalami kendala. Untuk mengatasi hal tersebut, diperlukan adanya suatu metode untuk meningkatkan *throughput bandwidth*, sehingga koneksi menjadi lebih cepat dan *traffic* merata.

Dari permasalahan tersebut penulis melakukan penelitian dengan mengimplementasikan teknologi bridge untuk jaringan *failover* dengan menggunakan protokol RSTP (*Rapid Spanning Tree Protocol*) dan mengimplementasikan teknologi bonding pada MikroTik untuk meningkatkan *throughput bandwidth*. Untuk pengujian *throughput bandwidth* menggunakan tool *BTest*. Pada pengujian ini dilakukan dengan menggunakan skenario yang berbeda-beda.

Hasil pengujian menunjukkan bahwa dengan menerapkan teknologi bridge dan protokol RSTP (*Rapid Spanning Tree Protocol*) pada jaringan, maka jaringan dapat menjadi *failover*, artinya apabila *link* utama terputus maka koneksi tetap berjalan menggunakan *link* yang masih aktif (*backup link*), selain itu protokol ini juga mencegah terjadinya *bridge loop*, dan dapat menentukan jalur alternatif untuk mengakomodasi jaringan agar tetap berjalan. Sedangkan pengujian dengan menerapkan teknologi bonding menunjukkan dapat meningkatkan *throughput bandwidth* dan *bandwidth* tersebar ke seluruh *link/node* sehingga menjadi *load balance*, selain itu dapat juga untuk *failover*, ketika jalur utama terputus, maka koneksi akan tetap berjalan dengan menggunakan jalur yang masih aktif (jalur alternatif) untuk menggantikan jalur utama sampai normal kembali.

**Kata kunci** : Bonding, *Bandwidth*, Bridge, *Failover*, *BTest*, RSTP (*Rapid Spanning Tree Protocol*)

### Abstract

The increasing number of LAN network users, when the network is experiencing constraints can make the user uncomfortable, but it can affect the bandwidth throughput that will be obtained, especially if the bandwidth is small then the connection will be slow, and sometimes the network has constraints. To overcome this, we need a method to increase bandwidth throughput, so the connection becomes faster and traffic is evenly distributed.

From these problems the authors conducted research by implementing bridge technology for failover network using RSTP protocol (*Rapid Spanning Tree Protocol*) and implement bonding technology on MikroTik to increase bandwidth throughput. For bandwidth throughput testing using *BTest* tool. In this test is done using different scenarios.

The test results show that by applying the bridge technology and RSTP (*Rapid Spanning Tree Protocol*) protocol on the network, the network can become failover, meaning that if the main link is disconnected then the connection still running using the link is still active (*backup link*), besides this protocol also prevent the occurrence of a bridge loop, and can determine alternate paths to accommodate the network to keep it running. While testing by applying bonding technology shows can increase bandwidth throughput and spread bandwidth to all link / node so that become load balance, besides can also for failover, when the main line is lost, then connection will still run by using still active path (alternative path) to replace the main line until normal again.

**Keywords**: Bonding, *Bandwidth*, Bridge, *Failover*, *BTest*, RSTP (*Rapid Spanning Tree Protocol*)



## PENDAHULUAN

Teknologi saat ini sudah menjadi salah satu kebutuhan pokok bagi masyarakat, terutama dalam bidang teknologi jaringan komputer. Seiring dengan kebutuhan masyarakat, jaringan saat ini memiliki manfaat tersendiri bagi setiap orang yang menggunakan jaringan karena kebutuhan bisa terfasilitasi lebih cepat. Hampir semua orang memanfaatkan teknologi jaringan komputer, hal ini disebabkan karena adanya pergeseran pola hidup masyarakat ke arah yang lebih berkembang. Sehingga semua dapat teratasi dengan maksimal.

Salah satu permasalahan yang biasanya kerap terjadi di masyarakat adalah masalah *bandwidth* yang diperoleh, karena banyak sedikitnya *bandwidth* akan mempengaruhi akses bagi pengguna, baik itu untuk *browsing*, *streaming*, *download* dan *upload*. Jika mendapatkan *bandwidth* sedikit maka koneksi akan menjadi lambat, sehingga pengguna merasa tidak puas dalam penggunaan internet. Oleh karena itu diperlukan adanya suatu metode untuk meningkatkan kapasitas *bandwidth*, atau disebut dengan *Link Agregation* yaitu penggabungan lebih dari satu interface. Pada mikrotik menggunakan metode Bonding untuk menggabungkan lebih dari satu interface kedalam satu *link* virtual sehingga akan mendapatkan *throughputbandwidth* yang lebih besar.

Ketika mengakses internet maka akan mendapatkan *throughput bandwidth* yang lebih besar dari penggabungan dua atau lebih *link/node* dengan menggunakan metode bonding. Disamping itu, ada sisi kelemahan terkadang jaringan mengalami kendala dan akhirnya terputus. Maka dari itu diperlukan adanya jalur alternatif sebagai jalur cadangan apabila jalur utama mengalami kendala, hal ini dilakukan agar koneksi tetap berjalan, yaitu dengan memanfaatkan *failover* dengan menggunakan *protocol* RSTP (*Rapid Spanning Tree Protocol*). Dari percobaan yang pernah ada sebelumnya yang berjudul “Implementasi Failover Layer Dua menggunakan Protocol RSTP dengan Router MikroTik” yaitu hanya menjelaskan tentang proses *failover* pada *layer 2* saja dengan menggunakan beberapa percobaan yang berbeda-beda, tidak membahas *throughput bandwidth*. Sehingga disini dikembangkan menjadi proses *failover* pada *layer 2* sekaligus meningkatkan *throughput bandwidth* dengan menggunakan metode bonding. Jadi metode bonding ini sendiri selain bisa meningkatkan *throughputbandwidth* (*load balancing*), juga bisa sebagai jalur *failover*, apabila *link* yang dibonding terputus, maka koneksi akan tetap berjalan dengan lancar karena adanya *link* cadangan sebagai jalur alternatif untuk menggantikan *link* tersebut. Jadi koneksi

tetap berlangsung karena masih terdapat *link backup* yang akan mengakomodasi proses komunikasi.

## KAJIAN PUSTAKA

### Bonding

Bonding merupakan sebuah teknologi untuk menggabungkan lebih dari satu *interface* dan menggabungkan kedalam satu *link* virtual sehingga akan mendapatkan *throughputbandwidth* yang lebih besar. Selain itu bisa digunakan untuk *failover*. Pada contoh implementasinya, bisa menghubungkan dua buah Router Mikrotik yang masing-masing *interface* pada router telah disetting bonding dan saling dikoneksikan. Atau bisa juga dari keempat *interface* tersebut dihubungkan ke sebuah *switch*, sehingga dapat berfungsi sebagai sebuah *backbone* penghubung jaringan yang memiliki *bandwidth* besar.

Menurut Rigar Yudistyo dan Indrastanti R. Widiari pada judul penelitiannya yaitu “Perbandingan Ethernet Bonding pada Linux CentOS dan RouterOS Guna Pemaksimalan Transfer Data”, bonding disebut *trunkinglink* agregasi yang artinya penggabungan beberapa *port* jaringan menjadi satu, sehingga bisa mendapatkan *bandwidth* yang lebih besar dari beberapa antarmuka, jadi misalnya dua ethernet 50 Mbps jika dilakukan bonding akan mendapatkan *bandwidth* maksimal 100 Mbps.

### Mode balance\_rr

Jika mode ini diset, paket dikirim secara berurutan dari *slaves* pertama yang tersedia sampai yang terakhir. Balance-rr adalah satu-satunya mode yang akan mengirim paket di beberapa antarmuka yang termasuk dalam koneksi TCP/IP yang sama. Saat menggunakan beberapa pengiriman dan banyak menerima *link*, paket sering kali diterima tanpa urutan, yang menghasilkan transmisi ulang segmen, untuk protokol lain seperti UDP, tidak masalah jika *client software* dapat mentolerir paket *out of order*. Jika switch digunakan untuk menggabungkan keseluruhan link, diperlukan konfigurasi *port* switch yang sesuai, namun banyak switch tidak mendukung balance-rr. Panduan penyiapan cepat menunjukkan penggunaan bonding mode balance-rr. Seperti yang bisa dilihat, cukup mudah untuk mengaturnya. Balance-rr juga berguna untuk bonding beberapa *wireless link*, namun memerlukan *bandwidth* yang sama untuk semua *link* yang terikat. Jika *bandwidth* dari satu *link* turun, maka total *bandwidth* yang di *bond* akan sama dengan *bandwidth* dari *linkbond* terlemah.

### Failover

Pada penelitian Nurul Fadilah Zamzami yang berjudul “Implementasi Load balancing Dan Failover Menggunakan Mikrotik Router Os Berdasarkan Multihomed Gateway Pada Warung Internet Diga”, *failover* dalam istilah computer *internetworking* adalah kemampuan sistem untuk dapat berpindah secara otomatis jika salah satu sistem mengalami

kegagalan sehingga menjadi *backup* untuk sistem yang mengalami kegagalan. Jika *link* satu mengalami *disconnect* (putus) maka *linkbackup*akan menggantikan *link* yang terputus. Jika *link* tersebut sudah kembali normal maka jalur koneksi yang digunakan kembali menjadi *link* tersebut. Dengan begitu dapat disimpulkan bahwa tujuan dari *failover* adalah digunakan untuk menggantikan atau sistem *backup* koneksi ISP (*Internet Service Provider*) yang terputus dengan koneksi ISP (*Internet Service Provider*) yang lainnya

**Bandwidth**

Pada jaringan komputer *bandwidth* merupakan kecepatan transfer data dari sebuah titik ke titik lain dalam jangka waktu tertentu. *Bandwidth* biasanya dinyatakan dalam ukuran bps (*bits per second*), adakalanya juga dinyatakan dalam Bps (*bytes per second*). Koneksi dengan *bandwidth* yang besar memungkinkan pengiriman informasi yang besar. Seberapa besar kecepatan transfer data sambungan internet tergantung pada jenis sambungan ke ISP (*Internet Service Provider*) yang digunakan. Jika menggunakan *dial-up*, maka kecepatan maksimal koneksinya adalah 56 kbps. Secara teoritis dapat mengirim.mengambil data dari internet dengan kecepatan tersebut. (Istina & Maryono, 2008).

**Tabel 1. Tabel *Bandwidth***

JenisMedia	Panjang Maksimal	Kecepatan Maksimal
Kabel Coaxial 10Base2	200m	10-100Mbps
Kabel Coaxial 10Base2	500m	10-100Mbps
UTP 10BaseT	100m	10Mbps
UTP100BaseTX	100m	100Mbps
Multimode 100BaseFX	2Km	100Mbps
Singlemode100BaseLX	3Km	1Gbps
Wireless	100m	2Mbps
InfraRed	1 m	4 Mbps

(Sumber : Imam Riadi, 2010)

**RSTP (*Rapid Spanning Tree Protocol*)**

RSTP merupakan penyempurnaan dari STP (*Spanning Tree Protocol*). Baik STP maupun RSTP akan melakukan “*blocking*” terhadap salah satu interface yang

telah di *bridge*. Interface yang berada dalam status *blocking* ini tidak akan meneruskan *traffic*, hal inilah yang mencegah terjadinya *looping*. *Root bridge* dalam suatu jaringan yang menerapkan *Root Bridge* dan router-router MikroTik telah mengaktifkan STP ataupun RSTP, maka dipilih sebuah router yang akan menjadi *Root Bridge*. *Root Bridge* merupakan Router/Bridge yang tidak memiliki *interface* dalam status *blocking*. Seperti yang sudah dijelaskan sebelumnya bahwa RSTP/STP nantinya akan membuat satu atau beberapa *interface* berada dalam status *blocking*. (Towidjojo, 2013).

Berikut merupakan Port Role dari RSTP :

1. **Root Port** ini berfungsi untuk menentukan langkah awal jalur mana yang memiliki jalur terbaik akan dilalui, dan setiap *switch* hanya memiliki satu *Root Port* yang mempunyai status *forwarding*.
2. **Designated Port** ini adalah port yang berfungsi sebagai menerima *frame* yang menuju *Root Bridge*. Port ini ada pada setiap *switch Root Bridge* maupun *non Root Bridge* yang memiliki status *forwarding*.
3. **Alternatife Port** ini adalah sebuah *port* yang berada pada *non designated switch* yang berupa jalur alternatif menuju *Root Bridge*.
4. **Backup Port** ini adalah sebuah *port* yang berada pada *designated switch* menuju segmen dimana *switch* berperan sebagai *designatedswitch*.

**Load Balancing**

RSTP merupakan penyempurnaan dari STP (*Spanning Tree Protocol*). Baik STP maupun RSTP akan melakukan “*blocking*” terhadap salah satu interface yang telah di *bridge*. Interface yang berada dalam status *blocking* ini tidak akan meneruskan *traffic*, hal inilah yang mencegah terjadinya *looping*. *Root bridge* dalam suatu jaringan yang menerapkan *Root Bridge* dan router-router MikroTik telah mengaktifkan STP ataupun RSTP, maka dipilih sebuah router yang akan menjadi *Root Bridge*. *Root Bridge* merupakan Router/Bridge yang tidak memiliki *interface* dalam status *blocking*. Seperti yang sudah dijelaskan sebelumnya bahwa RSTP/STP nantinya akan membuat satu atau beberapa *interface* berada dalam status *blocking*. (Towidjojo, 2013).

**Simple Queue**

*Simple Queue* merupakan metode *bandwidthmanagement* termudah yang ada di Mikrotik. Menu dan konfigurasi yang dilakukan untuk menerapkan *simple queue* cukup sederhana dan mudah dipahami. Walaupun namanya *simple queue* sebenarnya parameter yang ada pada *simple queue* sangat banyak, bisa disesuaikan dengan kebutuhan yang ingin diterapkan pada jaringan. Parameter dasar dari *simple queue*

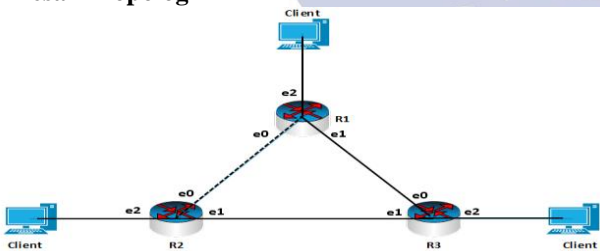
adalah Target dan Max-limit. Target dapat berupa *IP address*, *network address*, dan bisa juga *interface* yang akan diatur *bandwidth*nya. Max limit *Upload/Download* digunakan untuk memberikan batas maksimal bandwidth untuk target.

**METODE**

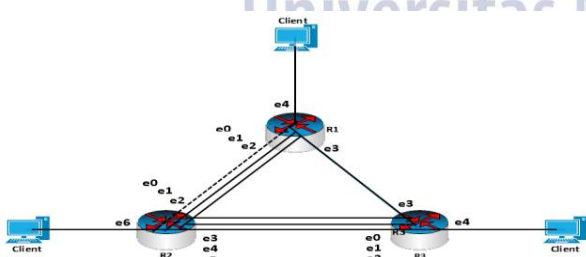
**Analisis Sistem**

Sistem yang akan dibuat adalah meningkatkan *throughput bandwidth* sekaligus sebagai jalur *failover* dengan menggunakan metode *bonding* pada MikroTik. Bonding merupakan suatu metode pada MikroTik dengan cara menggabungkan lebih dari satu *interface* menjadi satu virtual untuk meningkatkan *throughput bandwidth* sehingga menjadi *balancing*, sehingga keseimbangan *traffic* menjadi lebih cepat dan merata. Pada metode ini, *throughput bandwidth*kan disebarakan melalui beberapa *link* yang terhubung. Sekaligus sebagai jalur *failover* dengan menggunakan RSTP bridge, berfungsi sebagai *link* cadangan apabila terjadi kendala pada *link* utama. Ketika sedang mengakses, maka *bonding* akan memberikan *throughput bandwidth* dengan jumlah yang lebih besar melalui tiga *link* tersebut. Dan *failover* berfungsi apabila pada koneksi tiba-tiba *link* utama terputus maka koneksi akan tetap berjalan dengan lancar dengan adanya *backup link*, karena *link* tersebut sebagai *link* cadangan untuk menggantikan *link* utama sampai *link* utama berjalan normal kembali.

**Desain Topologi**



Gambar 1. Topologi failover dengan Bridge



Gambar 2. Topologi failover dengan Bonding

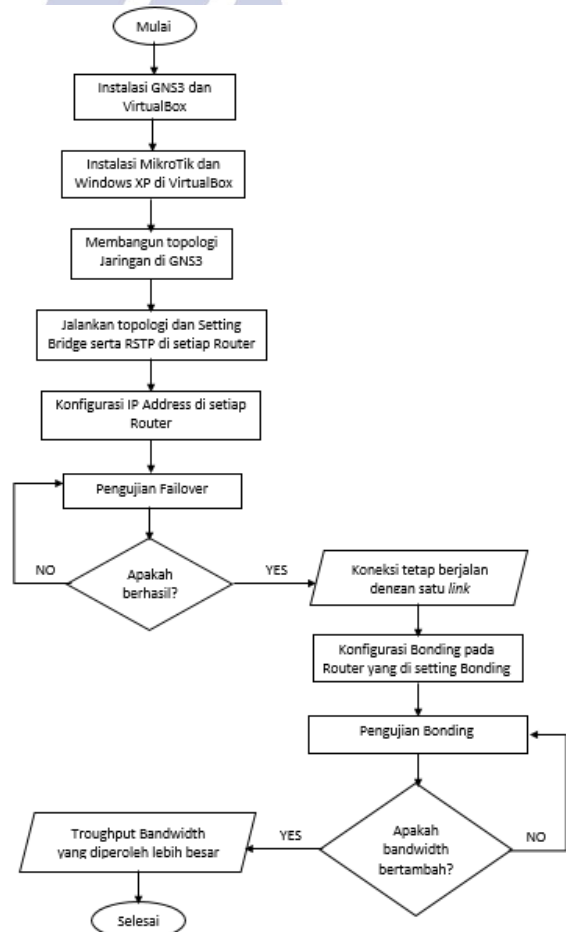
Pada Gambar 1 menunjukkan topologi *failover* dengan menggunakan RSTP bridge dan pada Gambar 2 menunjukkan topologi Bonding sekaligus sebagai *failover*.

Untuk penjelasan pada Gambar 1 yaitu ketika *Client R2* melakukan koneksi ke *Client R1* dan apabila *link* antara R1 dan R2 terputus maka koneksi akan tetap berjalan lancar dengan melalui *link* antara R2 dan R3 kemudian ke *link* antara R3 dan R1.

Sedangkan penjelasan Gambar 2 yaitu ketika *Client R3* melakukan koneksi ke *Client R1* melalui *link* yang sudah disetting bonding maka otomatis *throughput bandwidth* yang diperoleh akan bertambah.

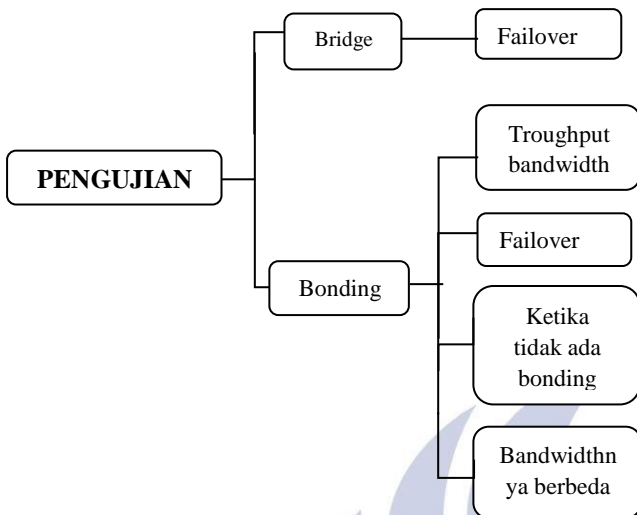
Apabila jalur antara R3 dan R2 tersebut mengalami kendala (terputus), maka koneksi akan tetap berjalan lancar dengan menggunakan jalur cadangan yaitu jalur/*link* antara R3 dan R1 sampai jalur utamakembali berjalan normal, hal ini disebut dengan *failover*.

**Perancangan Sistem**



Gambar 3. Perancangan Sistem

### Skenario Pengujian



Gambar 4. Skenario Pengujian

Penjelasan Gambar 4 tentang skenario pengujian adalah sebagai tahap menganalisa hasil dari uji coba. Tahapan pengujian penelitian sebagai berikut:

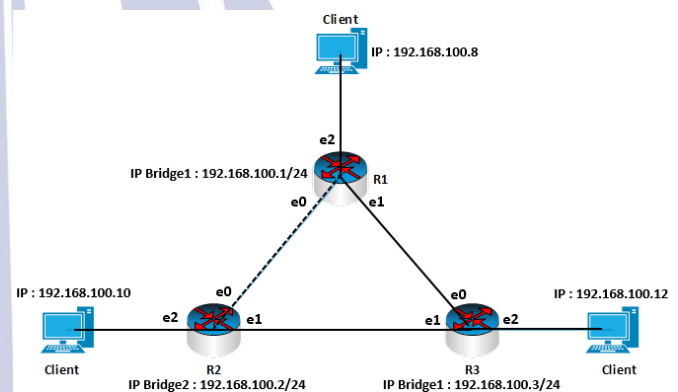
- 1) Pengujian satu *link*, yaitu untuk failover. Pengujian ini dilakukan untuk melihat proses terjadinya *failover* dan untuk membuktikan bahwa dengan menggunakan satu *link* saja koneksi akan tetap berjalan.
- 2) Pengujian tiga *link* (bonding), penulis akan melakukan pengujian untuk melihat proses *failover* dan berapa besar *troughputbandwidth* yang diperoleh.
  - a. Pengujian yang pertama yaitu untuk melihat *troughput bandwidth* yang diperoleh.
  - b. Pengujian yang kedua yaitu untuk *failover*, pengujian ini dilakukan untuk membuktikan bahwa meskipun satu *link* terputus maka koneksi tetap berjalan.
  - c. Pengujian yang ketiga yaitu ketika tidak ada bonding. Pengujian ini dilakukan untuk melihat hasil *troughput* dari percobaan yang dilakukan apabila tidak ada bonding.
  - d. Sedangkan pengujian yang keempat dilakukan untuk melihat hasil *bandwidthnya* apabila alokasi *bandwidthnya* berbeda.

### HASIL DAN PEMBAHASAN

Pada tahap ini merupakan penerapan dari hasil analisa dan desain sistem yang telah dibuat pada bab

sebelumnya. Penelitian ini bertujuan untuk implementasi meningkatkan *troughputbandwidth* dan proses *failover*. Untuk mendapatkan *troughput* dalam jumlah besar dilakukan dengan menggunakan metode bonding pada MikroTik, ketika melakukan Tes *Bandwidth* ke IP bonding maka otomatis *bandwidth* yang diperoleh akan bertambah. Untuk proses *failover* terjadi apabila salah satu *link* terputus maka akan berpindah ke *link* yang lainnya. Dalam penelitian ini terdapat beberapa konfigurasi jaringan yang akan dijelaskan lebih detail pada bab ini.

Di bawah ini adalah Desain arsitektur jaringan untuk proses *failover* dan bonding. Seperti pada bab sebelumnya, Gambar 5 merupakan desain sistem untuk *Failover* dengan Bridge serta pembagian IP Address. Pada sistem ini dijelaskan bahwa koneksi akan tetap berlangsung melalui *link* yang lainnya meskipun satu *link* terputus.

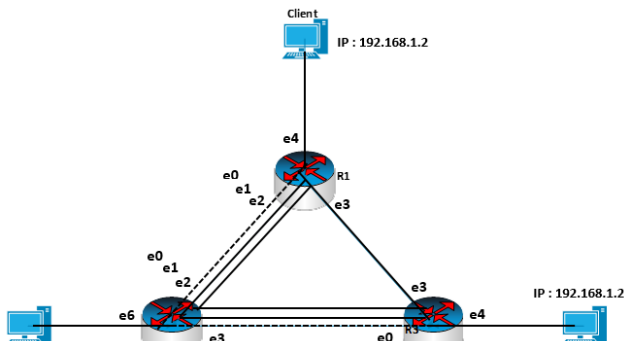


Gambar 5. Topologi *Failover* dengan Bridge

Berikut merupakan pembagian IP yang terdapat pada Gambar 4.1 :

1. Router1  
Bridge1 : 192.168.100.1/24
2. Router2  
Bridge2 : 192.168.100.2/24
3. Router3  
Bridge3 : 192.168.100.3/24
4. Client R1  
IP Address : 192.168.100.8/24
5. Client R2  
IP Address : 192.168.100.10/24
6. Client R3  
IP Address : 192.168.100.12/24

Sedangkan pada Gambar 6 merupakan desain sistem untuk *Failover* dengan menggunakan Bonding. Pada sistem ini dijelaskan tentang *troughput bandwidth* yang diperoleh dan proses *failover*.



Gambar 6. Topologi Failover dengan Bonding

Berikut merupakan pembagian IP yang terdapat pada Gambar 6 :

1. Client R1  
IP Address : 192.168.1.1/24
2. Client R3  
IP Address : 192.168.1.2/24

### Hasil Implementasi

Berikut adalah konfigurasi yang dilakukan pada Router maupun Client

1. Hasil Implementasi Failover dengan Bridge pada Router1, Router2, dan Router3 :
  - a. Setting bridge dengan menambahkan interface bridge
  - b. Setting RSTP
  - c. Penambahan port-port yang di bridge
  - d. Setting IP Address
2. Hasil implementasi Failover dengan Bonding Konfigurasi pada Router1 dan Router3 :
  - a. Setting bonding dengan menambahkan interface bonding
  - b. Pilih Mode balance\_rr
  - c. Setting bridge dan port-port yang akan di bridge
  - d. Aktifkan RSTP
  - e. Centang Use Firewall
  - f. Setting Simple Queue dan alokasi bandwidth
  - g. Setting IP Address

Konfigurasi pada Router2 :

- a. Setting bridge dan port-port yang akan di bridge
- b. Aktifkan STP/RSTP
- c. Centang Use Firewall
- d. Setting Simple Queue dan alokasi bandwidth

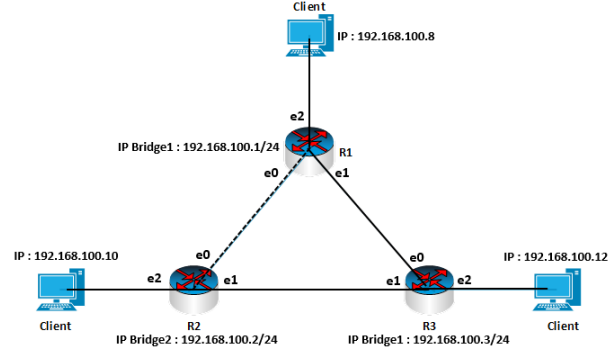
### Pengujian dan Pembahasan

#### 1. Pengujian Failover dengan Bridge

Pengujian dengan menggunakan satu link adalah pengujian untuk failover. Pengujian ini dilakukan dengan melakukan percobaan memutus salah satu dari diantara

tiga link antar router, untuk melihat hasilnya yaitu dengan melakukan test ping antar router maupun antar Client.

- a. Pengujian Failover dengan memutus link antara Router1 dan Router2



Gambar 7. Memutus link antara R1 dan R2

- 1) Pada Gambar 8 Test ping dari Client R2 ke Client R1 yang memiliki alamat IP 192.168.100.1 dan terlihat bahwa sudah berhasil ping ke Client R1. Pengujian ini dilakukan sebelum link antara R1 dan R2 terputus.

```

C:\WINDOWS\system32\cmd.exe - ping 192.168.100.8 -t
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\nikmah>ping 192.168.100.8 -t
Pinging 192.168.100.8 with 32 bytes of data:
Reply from 192.168.100.8: bytes=32 time=3ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=4ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=3ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=3ms TTL=128
Reply from 192.168.100.8: bytes=32 time=3ms TTL=128
    
```

Gambar 8. Ping dari Client R2 ke Client R1

- 2) Gambar 9 berikut adalah hasil ping setelah link antara R1 dan R2 diputus, terlihat bahwa ada jeda sebentar atau timeout, tetapi koneksi kembali lagi normal seperti semula meskipun jalur yang menuju ke Root Bridge terputus, karena otomatis root port yang awalnya menuju ke Root Bridge akan berpindah ke link alternatif sebagai jalur cadangan untuk mengakomodasi koneksi.

```

Command Prompt - ping 192.168.100.8 -t
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\nikmah>ping 192.168.100.8 -t
Pinging 192.168.100.8 with 32 bytes of data:
Reply from 192.168.100.8: bytes=32 time=9ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=2ms TTL=128
Reply from 192.168.100.8: bytes=32 time=3ms TTL=128
Request timed out.
Reply from 192.168.100.8: bytes=32 time=4ms TTL=128
Reply from 192.168.100.8: bytes=32 time=3ms TTL=128
Reply from 192.168.100.8: bytes=32 time=3ms TTL=128
Reply from 192.168.100.8: bytes=32 time=3ms TTL=128
    
```

Gambar 9. Hasil ping setelah link antara R1 dan R2 terputus Pengujian Failover dengan Bonding

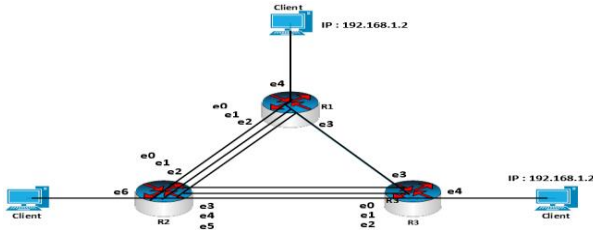
Pengujian dengan menggunakan tiga link adalah pengujian untuk meningkatkan throughput bandwidth sekaligus untuk failover. Pengujian ini dilakukan dengan melakukan beberapa percobaan untuk melihat berapa besar throughput bandwidth yang diperoleh ketika tiga link sudah di

Meningkatkan *Troughput Bandwidth* sekaligus sebagai Jalur *Failover* dengan menggunakan Metode Bonding pada MikroTik

setting bonding. Untuk melihat hasilnya yaitu dengan melakukan test *bandwidth* dengan menggunakan Btest yang sudah di instal di Client/PC.

Hal ini bisa juga terjadi *failover* apabila salah satu *link* di putus maka koneksi tetap berjalan menggunakan dua *link* yang masih aktif

- a. Pengujian *bandwidth* dengan test *bandwidth* dari Client R3 ke Client R1



Gambar 10. Pengujian *Bandwidth* dengan Bonding

- 1) Test *bandwidth* ke Client Router1 dengan IP 192.168.1.1, terlihat pada Gambar 11 bahwa *bandwidth* yang diperoleh yaitu kurang lebih 3Mb, karena setiap *link* dialokasikan *bandwidth* sebesar 1Mb pada Simple Queue, dan *traffic* tersebar merata sehingga tidak *overload* pada masing-masing *link*, karena pada mode *balance rr* dan metode Round Robin sendiri mendukung teknik *load balancing*.



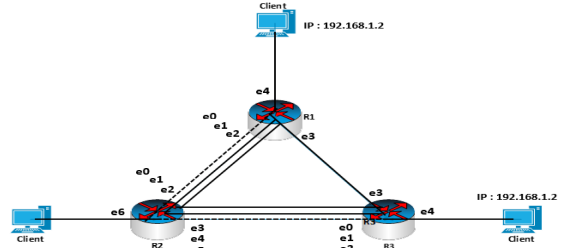
Gambar 11. Hasil Test *Bandwidth* dengan Bonding

- 2) Dikarenakan *bandwidth* yang diperoleh sesuai dengan yang dialokasikan, maka otomatis masuk ke Queue sehingga berwarna merah. Seperti yang terlihat pada Gambar 12.

#	Name	Target Ad...	Rx Max Limit	Tx Max Limit	Packet ...
0	queue1	1M	1M		
1	queue2	1M	1M		
2	queue3	1M	1M		

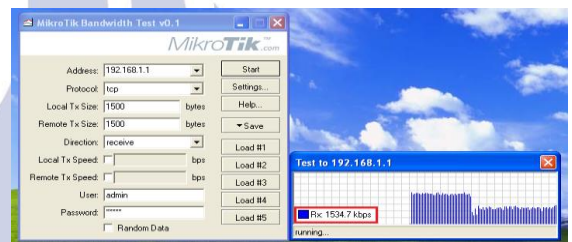
Gambar 12. Tampilan Queue apabila test *bandwidth* stabil

- b. Pengujian *bandwidth* dengan test *bandwidth* dengan disable salah satu *port*



Gambar 13. Pengujian *bandwidth* dengan memutus salah satu *interface/link*

- 1) Test *bandwidth* dari Client R3 ke Client R1, pada Gambar 14 adalah tampilan dari *traffic* setelah salah satu *interface/link* dimatikan. Terlihat bahwa mendapatkan *bandwidth* kurang lebih 2Mb yang sebelumnya mendapatkan *bandwidth* kurang lebih 3Mb, dan *traffic* nya tersebar merata sehingga tidak *overload* pada masing-masing *link*.



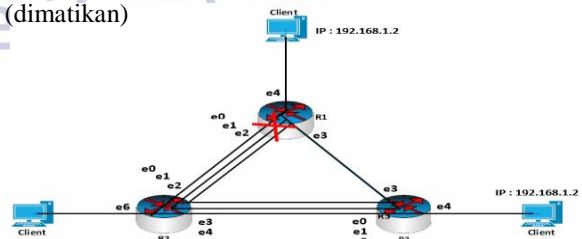
Gambar 14. Tampilan *traffic* setelah salah satu

- 2) Gambar 15 adalah tampilan Queue bahwa berhasil melalui *queue2* dan *queue3*, karena *interface* pada *queue* yang pertama sudah diputus.

#	Name	Target Ad...	Rx Max Limit	Tx Max Limit	Packet ...
0	queue1	1M	1M		
1	queue2	1M	1M		
2	queue3	1M	1M		

Gambar 15. Tampilan *queue* setelah salah satu *interface* disable

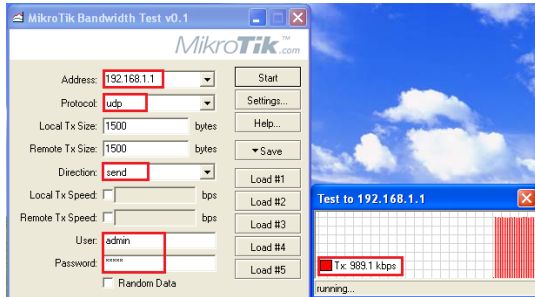
- c. Pengujian *Failover* dengan *interface* bonding di *disable* (dimatikan)



Gambar 16. Pengujian *Failover* dengan *disable interface* bonding

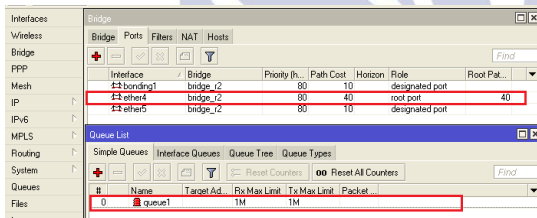


- 1) *Testbandwidth* dari *Client R3* ke *Client R1*. Terlihat pada Gambar 17 bahwa *throughput bandwidth* yang diperoleh adalah kurang lebih 1Mb, karena otomatis jalur yang dilalui berpindah ke *ether4* pada Router3, dan sudah dialokasikan *bandwidth* pada Simple Queue sebesar 1Mb



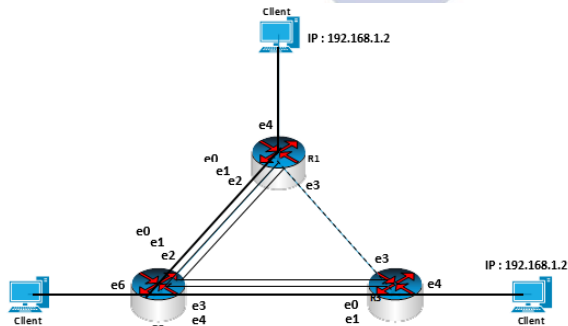
Gambar 17. Tampilan *traffic* ketika interface bonding di disable

- 2) Gambar 18 adalah tampilan bahwa jalur berpindah ke *ether4*. Dan berhasil melalui *queue1* dengan *bandwidth* kurang lebih 1Mb.



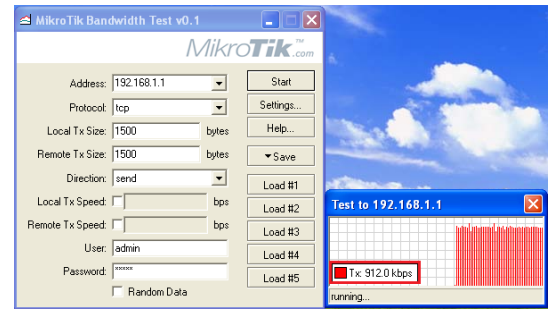
Gambar 18. Tampilan *queue* ketika interface bonding di disable

- d. Pengujian *bandwidth* jika tidak ada bonding



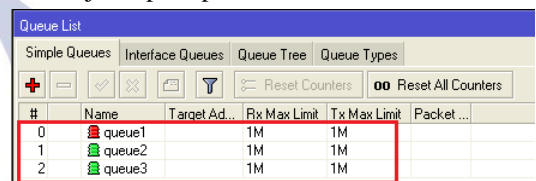
Gambar 19. Pengujian *bandwidth* tanpa bonding

- 1) *Testbandwidth* dari *Client R3* ke *Client R1*, pada Gambar 20 adalah hasil *test bandwidth* ke *Client R1*. Terlihat bahwa *throughput bandwidth* yang diperoleh kurang lebih 1Mb, karena hanya melalui salah satu *link* dari diantara tiga *link* yang hanya murni menggunakan *setting bridge* dan setiap *link* dialokasikan *bandwidth* sebesar 1Mb.



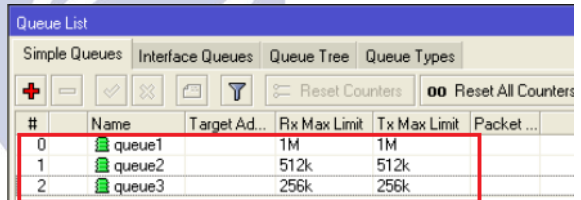
Gambar 20. Tampilan *traffic* jika tanpa bonding

- 2) Terlihat pada *queue* bahwa hanya melalui *queue1* dan hanya mendapatkan *bandwidth* kurang lebih 1Mb saja. Seperti pada Gambar 21 berikut



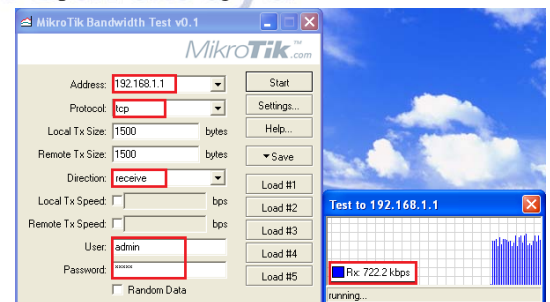
Gambar 21. Tampilan *queue* jika tanpa bonding

- e. Pengujian dengan alokasi *bandwidth* yang berbeda



Gambar 22. Alokasi *bandwidth* berbeda

- 1) Lalu lakukan *testbandwidth* dari *Client R3* ke *Client R1*, Gambar 23 berikut adalah hasil *testbandwidth* setelah mengubah alokasi *bandwidth* seperti yang terlihat pada Gambar 22. *Throughputbandwidth* yang diperoleh kurang lebih 750k, karena mengikuti *bandwidth* terendah kemudian ketiga *link* dijumlahkan, dan *traffic* tersebar merata sehingga menjadi *balance*. Pada mode *balance rr* dijelaskan bahwa *bandwidth* yang didapat adalah mengikuti *bandwidth* terendah.

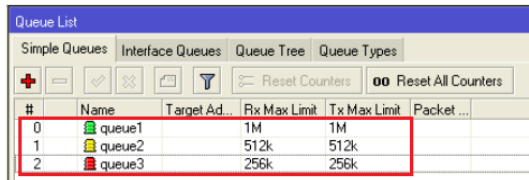


Gambar 23. Hasil *testbandwidth* berbeda

- 2) Di atas terlihat bahwa *traffic* berjalan kurang lebih 750k, hal ini dikarenakan bonding pada MikroTik

Meningkatkan *Troughput Bandwidth* sekaligus sebagai Jalur *Failover* dengan menggunakan Metode Bonding pada MikroTik

berlaku setiap *queue* harus *balance*, artinya beban setiap *link/queue* sama dengan *link/queue* yang lain, sehingga ketika dijumlahkan *bandwidth* yang diperoleh adalah sebesar diantara kisaran 750k, dan mengikuti *minimumbandwidth* yang disediakan, jadi otomatis *queue* yang dilalui seperti pada Gambar 24 berikut.



Gambar 24. Tampilan *queue* jika alokasi *bandwidth* berbeda

Tabel 2. Tabel jika Alokasi *bandwidth* berbeda

Link ke-	Bandwidth	Hasil
1	1M	256k
2	512k	256k
3	256k	256k

**KESIMPULAN DAN SARAN**

**Simpulan**

- Dari Hasil Pengujian dan implemantasi yang dilakukan pada penelitian dapat disimpulkan bahwa dengan memanfaatkan *protocol RSTP (Rapid Spanning Tree Protocol)* bisa digunakan untuk *failover*, pada pengujian dapat dilihat ketika *link* utama terputus, maka koneksi tetap terhubung melalui *link* yang lainnya (*backup link*). Ketika *link* utama kembali normal maka koneksi akan kembali ke *link* utama. Sedangkan pada pengujian failover dengan bonding dapat dilihat bahwa dengan menggunakan teknologi bonding bisa juga sebagai *failover*, apabila jalur bonding terputus maka bisa melalui *link* cadangan dan menggunakan *protocol RSTP* agar tidak terjadi *looping*.
- Dengan menggunakan metode bonding pada MikroTik, maka dapat disimpulkan bahwa:
  - Pada pengujian menunjukkan bahwa dengan menggunakan teknologi bonding dapat meningkatkan *troughputbandwidth* kurang lebih 3Mb sesuai *bandwidth* yang dialokasikan yaitu sebesar 1Mb setiap *link* sehingga bisa *load balance (bandwidth* merata).

- Ketika tidak ada bonding maka jalur yang bisa dilalui hanya salah satu *link* saja diantara tiga *link*, jadi murni *setting* bridge.
- Pada pengujian yang telah dilakukan apabila alokasi *bandwidth*nya berbeda maka *troughput* yang didapat sesuai dengan penjumlahan tiga *link* tersebut, akan tetapi *bandwidth* pada tiga *link* tersebut besarnya rata, dan mengikuti *bandwidth* yang terendah.

**Saran**

Diharapkan pada penelitian selanjutnya mampu mengimplementasikan teknologi bonding maupun jaringan *failover* dengan menggunakan perangkat secara Real, ataupun dikombinasikan dengan metode lainnya, karena pada penelitian ini menggunakan metode bonding dengan mode *balance rr*.

**DAFTAR PUSTAKA**

Istiana, B. Patmi dan Y. Maryono. (2008). *Teknologi Informasi dan Komunikasi 3*. Quadra

Riadi, Imam. (2010). *Optimasi Bandwidth menggunakan Traffic Shapping*. Universitas Ahmad Dahlan Yogyakarta. (Jurnal). (<http://download.portalgaruda.org/article.php?article=123348&val=5541>), diakses pada tanggal 5 Maret 2017

Towidjojo, Rendra. (2013). *MikroTik Kung Fu Kitab 2 RSTP*. Palu: Jasakom

Towidjojo, Rendra. (2016). *MikroTik Kung Fu Kitab 4 Load Balancing*. Palu: Jasakom

Website. (2005-2007). *MikroTik Indonesia Konfigurasi Dasar Bonding Interface*. ([http://www.mikrotik.co.id/artikel\\_lihat.php?id=161](http://www.mikrotik.co.id/artikel_lihat.php?id=161)), diakses pada tanggal 5 Maret 2017

Website. (2005-2017). *MikroTik Indonesia Simple Queue VS QueueTree*. ([http://www.mikrotik.co.id/artikel\\_lihat.php?id=251](http://www.mikrotik.co.id/artikel_lihat.php?id=251)), diakses pada 11 September 2017

Website. (2017). *Wiki MikroTik Interfaace/Bonding*. (<https://wiki.mikrotik.com/wiki/Manual:Interface/Bonding>), diakses pada 11 September 2017

Yudistyo, Riger dan Indrastanti R. Widiarsari. (2012). *Perbandingan Ethernet Bonding pada Linux CentOS dan RouterOS Guna Pemaksimalan Transfer Data*. Salatiga. (Jurnal). ([http://repository.uksw.edu/bitstream/123456789/2846/2/ART\\_Rigar%20Yudistyo,%20Indrastanti%20RW\\_Perbandingan%20Ethernet%20Bonding%20pada%20Linux\\_Full%20text.pdf](http://repository.uksw.edu/bitstream/123456789/2846/2/ART_Rigar%20Yudistyo,%20Indrastanti%20RW_Perbandingan%20Ethernet%20Bonding%20pada%20Linux_Full%20text.pdf)), diakses pada 5 Maret 2017

Zamzami, Nurul Fadilah. (2013). *Implementasi Load Balancing dan Failover menggunakan MikroTik*

*Router OS berdasarkan Multihomed Gateway pada Warung "DIGA". Bandung. (Jurnal). ([http://eprints.dinus.ac.id/12153/1/jurnal\\_12049.pdf](http://eprints.dinus.ac.id/12153/1/jurnal_12049.pdf)), diakses pada 7 Maret 2017*

