



**TEKNOLOGI PENGGABUNGAN BANDWIDTH  
DOWNSTREAM MENGGUNAKAN OPERATING SYSTEM  
BERBASIS UNIX**

**PROYEK AKHIR**

**Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta  
Untuk Memenuhi Sebagian Persyaratan Guna Memperoleh  
Gelar Ahli Madya  
Program Studi Teknik Elektronika**

**Disusun Oleh :**

**YUDHA CHRISTIANTO FIRMANSYAH  
NIM. 06507134018**

**PROGRAM STUDI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA**

**DESEMBER 2009**

**LEMBAR PERSETUJUAN**

**PROYEK AKHIR**

**TEKNOLOGI PENGGABUNGAN BANDWIDTH DOWNSTREAM  
MENGUNAKAN OPERATING SYSTEM BERBASIS UNIX**

Oleh :

Yudha Christianto Firmansyah

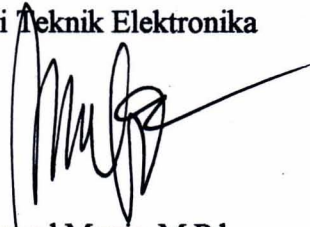
06507134018

Telah Diperiksa Dan Disetujui Oleh Pembimbing Untuk Diuji

Yogyakarta, 15 Juni 2011

Mengetahui,

Kaprodi Teknik Elektronika



Muhammad Munir, M.Pd.  
NIP. 19630512 198901 1 001

Menyetujui,

Pembimbing



Totok Sukardiyono, MT  
NIP. 19670930 199303 1 005

**LEMBAR PENGESAHAN**  
**PROYEK AKHIR**  
**TEKNOLOGI PENGGABUNGAN BANDWIDTH DOWNSTREAM**  
**MENGGUNAKAN OPERATING SYSTEM BERBASIS UNIX**

Dipersiapkan Dan Disusun Oleh :

Yudha Christianto Firmansyah  
06507134018

Telah Dipertahankan di Depan Dewan Penguji Proyek Akhir  
Fakultas Teknik Universitas Negeri Yogyakarta  
Pada Tanggal 28 Juni 2011  
Dan Dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar  
Ahli Madya

**SUSUNAN PANITIA PENGUJI**

<b>Jabatan</b>	<b>Nama Lengkap</b>	<b>Tanda Tangan</b>
1. Ketua Penguji	: Totok Sukardiyono, MT	.....
2. Sekretaris	: Drs. Kadarisman Tejo Yuwono	.....
3. Penguji Utama	: Herman Dwi Surjono, Ph.D	.....

Yogyakarta, 28 Juni 2011

Dekan FT UNY



Wardan Suyanto, Ed.D

NIP. 19540810 197803 1 001

## LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Yudha Christianto Firmansyah

Nim : 06507134018

Jurusan : Teknik Elektronika

Judul : Teknologi Penggabungan Bandwidth Downstream Menggunakan  
Operating System Berbasis Unix

Menyatakan bahwa dalam pembuatan Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini merupakan karya asli dari penulis dalam pembuatan proyek akhir ini. Sepanjang pengetahuan penulis belum menjumpai karya yang pernah diajukan untuk memperoleh Gelar Ahli Madya di Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta atau Perguruan Tinggi lainnya. Kalaupun itu ada bentuk mekanik, rangkaian kendali (*hardware*) dan program (*software*) pasti berbeda. Jika ternyata terbukti pernyataan ini tidak benar sepenuhnya menjadi tanggung jawab penulis.

Yogyakarta, 28 Desember 2009

## MOTTO

Sesungguhnya semua urusan (perintah) apabila Allah menghendaki segala  
sesuatunya,

Allah hanya berkata “Jadi” maka jadilah.

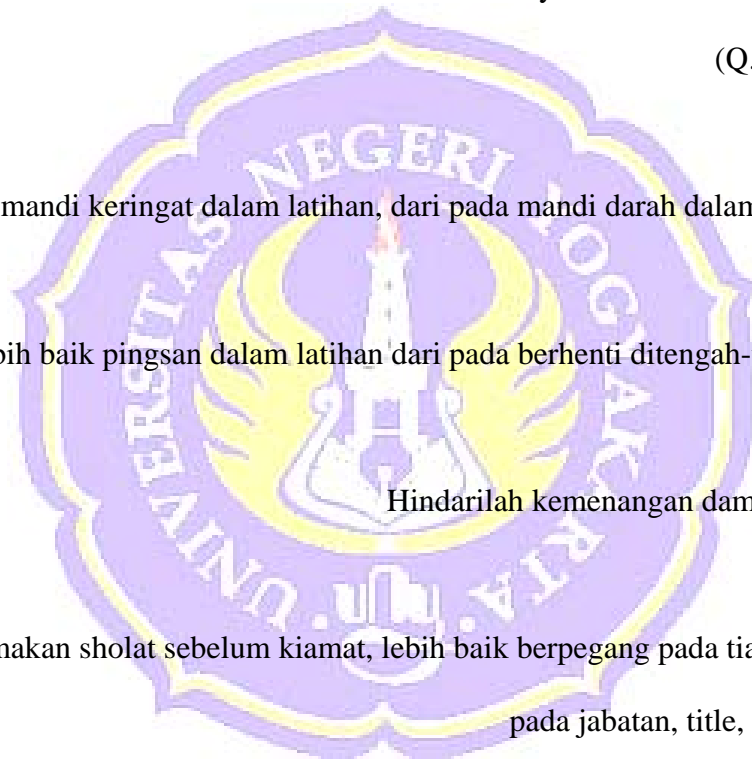
(Q.S Yaasiin :82)

Lebih mandi keringat dalam latihan, dari pada mandi darah dalam pertempuran.

Lebih baik pingsan dalam latihan dari pada berhenti ditengah-tengah latihan.

Hindarilah kemenangan demi keselamatan.

Utamakan sholat sebelum kiamat, lebih baik berpegang pada tiang agama dari  
pada jabatan, title, atau yang lain.



# **Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix**

*Oleh :*  
*Yudha Christianto Firmansyah*  
*06507134018*

## **ABSTRAK**

Tujuan dari pembuatan proyek akhir ini adalah untuk mengetahui dan merealisasikan susunan perangkat keras (*hardware*), perangkat lunak (*software*) dan pembuatan mekanik dari *software* yang akan digunakan, serta untuk mengembangkan *software* MikroTik yang digunakan dalam Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix.

Metode yang digunakan dalam membuat Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini menggunakan metode rancang bangun yang terdiri dari beberapa tahap yaitu: (1) Identifikasi kebutuhan, (2) Analisis kebutuhan, (3) Konsep rancangan, (4) Pembuatan dan (5) Pengujian. Sehingga didapatkan model Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix yang terdiri dari (1) Unit komputer, yang berfungsi untuk memproses data masukan dan data keluaran dan sebagai *interface*. (2) MikroTik, berfungsi untuk Mengkonfigurasi jaringan Internet masuk (*input*) dan hasil penggabungan jaringan Internet keluar (*output*). (3) MODEM, yang berfungsi untuk merubah sinyal masukan yang masih dalam bentuk *analog* ke bentuk *digital* supaya dapat terbaca oleh komputer. (4) *LAN card*, sebagai *interface* antara unit komputer dengan MODEM dan sebagai pendistribusi hasil keluaran proses (*output*).

Cara pengoperasian alat ini adalah dengan menyalakan unit komputer, dan MODEM, kemudian kabel *UTP* yang terhubung ke modem dan ke unit komputer klien dipasangkan pada *LAN card* komputer, dengan *software* MikroTik yang telah dikonfigurasi sedemikian rupa, maka secara otomatis jaringan Internet yang masuk ke unit komputer akan diproses sehingga keluarannya akan menjadi satu jaringan yang utuh.

## KATA PENGANTAR

اَشْكُرُكُمْ وَرَحْمَةُ اللهِ وَبَرَكَاتُهُ

Puji dan syukur penulis ucapkan kehadiran Allah SWT, yang telah melimpahkan rahmat, dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan proyek akhir yang merupakan syarat kelulusan program Diploma di Fakultas Teknik Universitas Negeri Yogyakarta.

Proyek akhir yang berjudul Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix merupakan karya akhir mahasiswa Diploma III dalam bentuk karya nyata.

Segala kemampuan serta upaya maksimal telah diusahakan untuk menyelesaikan proyek akhir ini. Berbagai pihak telah banyak membantu penulis, oleh karena itu disampaikan penghargaan dan rasa terima kasih yang sebesar-besarnya kepada :

1. Bapak Wardan Suyanto, Ed.D selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
2. Bapak Masduki Zakarijah, MT selaku Ketua Jurusan sekaligus Penasehat Akademik Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta.
3. Bapak Muhammad Munir, M.Pd. selaku Ketua Program Studi Diploma Teknik Elektronika Universitas Negeri Yogyakarta.
4. Bapak Drs. Djoko Santoso, M.Pd. selaku Koordinator Proyek Akhir Teknik Elektronika Universitas Negeri Yogyakarta.
5. Bapak Totok Sukardiyono, MT selaku pembimbing Proyek Akhir yang banyak memberikan bantuan dan pengarahan kepada penulis.
6. Almarhum papa tercinta yang selalu memberi bimbingan, semangat dan dorongan dalam berbagai hal kepada penulis.
7. Mama dan semua keluarga yang telah mendoakan serta memberikan dukungan dalam pembuatan Proyek Akhir.
8. Robby Firmansyah & Bobby Ardian yang telah banyak membantu dan memfasilitasi dalam pengerjaan Proyek Akhir ini.

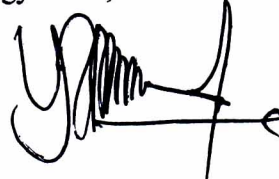
9. Teman-teman mahasiswa Kelas C tahun angkatan 2006 yang telah memberikan semangat dan memberikan pengetahuan untuk pembuatan proyek akhir.
10. Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu menyelesaikan proyek akhir.

Menyadari bahwa laporan proyek akhir ini masih terdapat banyak **kekurangan**, oleh karena itu segala kritik dan saran yang konstruktif selalu **diharapkan** demi kesempurnaan laporan Proyek Akhir.

Semoga laporan Proyek Akhir ini dapat bermanfaat bagi penulis dan **berbagai** pihak yang membutuhkan.

وَالسَّلَامُ عَلَيْكُمْ وَرَحْمَةُ اللَّهِ وَبَرَكَاتُهُ

Yogyakarta, 28 Desember 2009



Yudha Christianto Firmansyah



## DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
HALAMAN PERSETUJUAN .....	ii
HALAMAN PENGESAHAN .....	iii
HALAMAN PERNYATAAN KEASLIAN .....	iv
MOTTO .....	v
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN .....	xiv
<b>BAB I PENDAHULUAN</b>	
A. Latar Belakang Masalah .....	1
B. Identifikasi Masalah .....	3
C. Batasan Masalah .....	4
D. Rumusan Masalah .....	4
E. Tujuan .....	5
F. Manfaat .....	5
G. Keaslian Gagasan .....	6
<b>BAB II KAJIAN TEORI</b>	
A. Load Balancing .....	7
1. <i>Grouping</i> .....	8
2. <i>Fail Over</i> .....	9
3. <i>PCC</i> .....	10
B. Perangkat Keras ( <i>Hardware</i> ).....	11
1. Komputer .....	11
2. MODEM ADSL .....	13
3. Kelebihan ADSL .....	16
3. Kekurangan ADSL .....	17

4. LAN CARD .....	20
C. Perangkat Lunak ( <i>Software</i> ) .....	21
MikroTik .....	21
- Sejarah MikroTik .....	21
- Pengertian Dasar .....	22
- Metode Konfigurasi .....	23
<b>BAB III PERANCANGAN ALAT</b>	
A. Identifikasi Kebutuhan <i>Hardware</i> dan <i>Software</i> .....	27
B. Analisis Kebutuhan <i>Hardware</i> dan <i>Software</i> .....	28
C. Alasan Pemilihan Komponen .....	29
D. Konsep Rancangan .....	31
1. Perancangan Perangkat Keras ( <i>Hardware</i> ) .....	31
a. Koneksi Internet .....	32
b. MODEM ADSL .....	32
c. CPU / Komputer <i>Server</i> .....	32
d. Laptop / Komputer Klien .....	33
e. Switch HUB .....	33
f. PC ( <i>Personal Computer</i> ) .....	33
2. Perancangan Perangkat Lunak ( <i>Software</i> ) .....	33
a. Program Utama .....	34
1 ) <i>Algoritma</i> .....	34
2 ) <i>Flow Chart</i> .....	35
E. Pembuatan Alat .....	36
a. Langkah Instalasi MikroTik .....	37
b. <i>Script Load Balancing</i> .....	40
c. Konfigurasi <i>Load Balancing</i> dengan WinBox .....	42
F. Spesifikasi Alat .....	51
G. Pengoperasian Alat .....	51
<b>BAB IV HASIL DAN PEMBAHASAN</b>	
A. Hasil Pengujian .....	53
1. Pengujian Fungsi Per bagian .....	53

a. Pengujian Unit MODEM ADSL .....	53
b. Pengujian Unit komputer .....	53
c. Pengujian Jaringan Internet .....	54
2. Pengujian Kinerja Alat Keseluruhan .....	54
B. Pembahasan .....	64
1. MODEM ADSL .....	64
2. Komputer Server .....	64
3. Kinerja Alat .....	64
<b>BAB V KESIMPULAN DAN SARAN</b>	
A. Kesimpulan .....	66
B. Keterbatasan Alat .....	67
C. Saran .....	67
<b>DAFTAR PUSTAKA .....</b>	<b>69</b>
<b>LAMPIRAN .....</b>	<b>70</b>

## DAFTAR GAMBAR

	Halaman
Gambar 1. Ilustrasi Metode <i>Grouping</i> .....	8
Gambar 2. Unit Komputer .....	11
Gambar 3. MODEM ADSL .....	13
Gambar 4. <i>Back panel</i> MODEM ADSL .....	18
Gambar 5. LAN CARD .....	20
Gambar 6. PuTTY .....	23
Gambar 7. Tampilan WinBox .....	25
Gambar 8. Telnet .....	26
Gambar 9. Blok Perancangan <i>Hardware</i> .....	31
Gambar 10. <i>Flow Chart</i> .....	35
Gambar 11. Menu Utama Instalasi MikroTik .....	37
Gambar 12. Tampilan Awal <i>Console</i> MikroTik .....	39
Gambar 13. Perintah Konfigurasi Pada MikroTik .....	40
Gambar 14. Tampilan Awal Winbox .....	43
Gambar 15. Tampilan Winbox Setelah Login .....	43
Gambar 16. Cara Mendeklarasikan IP pada MikroTik. ....	44
Gambar 17. Cara Mengisi DNS pada MikroTik. ....	45
Gambar 18. Cara NAT dari <i>Firewall</i> pada MikroTik .....	46
Gambar 19. Cara Mengisi NAT dari <i>Firewall</i> pada MikroTik.....	47
Gambar 20. Cara Mendeklarasikan <i>Routes</i> pada MikroTik.....	49
Gambar 21. Pengujian Jaringan <i>Speedy</i> .....	55
Gambar 22. Hasil Penggabungan 2 Jaringan <i>Speedy</i> .....	55
Gambar 23. Pengujian <i>Speedy</i> dan TELKOMSEL FLASH ...	56
Gambar 24. Hasil Penggabungan <i>Speedy</i> dengan TELKOMSEL FLASH ....	57
Gambar 25. Pengujian Jaringan GSM 3 .....	58
Gambar 26. Hasil Penggabungan Jaringan GSM 3 .....	58
Gambar 27. Pengujian GSM 3 dan FLASH .....	59
Gambar 28. Hasil Penggabungan GSM 3 dan FLASH .....	59

Gambar 29. Pengujian GSM 3 dan CDMA <i>Smartfren</i> ... ..	60
Gambar 30. Hasil Penggabungan GSM 3 dan CDMA <i>Smartfren</i> ... ..	60
Gambar 31. Pengujian Jaringan GSM 3 .....	62
Gambar 32. Pengujian Jaringan CDMA <i>Smartfren</i> ... ..	62
Gambar 33. Pengujian Jaringan GSM TELKOMSEL FLASH ... ..	63

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1. <i>Datasheet</i> MODEM ADSL .....	71
Lampiran 2. <i>Installing RouterOS</i> .....	74
Lampiran 3. <i>Terminal Console MikroTik</i> .....	78
Lampiran 4. <i>Winbox</i> .....	86
Lampiran 5. <i>Mangle</i> .....	92
Lampiran 6. <i>NAT</i> .....	100

## BAB I

### PENDAHULUAN

#### A. Latar Belakang Masalah

Perkembangan teknologi dibidang jaringan Internet secara global saat ini sangatlah pesat, berbagai macam layanan yang ditawarkan oleh para provider cukup membingungkan para konsumen. Pada dasarnya dalam berbagai hal manusia menginginkan untuk mendapatkan suatu barang yang berkualitas tinggi namun dengan harga yang semurah-murahnya, sama halnya dengan para pengguna jasa layanan Internet, mereka rela mengeluarkan uang lebih banyak demi mendapatkan akses Internet dengan akselerasi tinggi. Beberapa orang memilih untuk meng-*upgrade bandwidth* akses yang mereka gunakan, namun sebagian besar orang beranggapan bahwa hal tersebut akan mengeluarkan biaya lebih mahal dibandingkan dengan pemasangan jaringan baru dari *provider* yang sama ataupun dari *provider* lain.

Dengan menambah jaringan dari *provider* yang sama maupun dari *provider* lain menimbulkan pemikiran untuk dapat menggunakan kedua jaringan tersebut secara optimal dan maksimal. Saat ini banyak yang menggunakan metode *grouping*, dan memikirkan bagaimana cara menggabungkan kedua jalur tersebut. Ada pula yang sudah mulai melirik metode penggabungan bandwidth (*Load Balancing*). Akan tetapi berdasarkan data observasi yang penulis lakukan pada tanggal 16 Maret 2009 di beberapa tempat antara lain warnet LK.net 1 pada jam 20.00 WIB, SMP N 1 pada jam 08.00 , dan perusahaan PLANT jam 09.00 WIB di Kecamatan Purwodadi Grobogan, rata-rata belum bisa menggabungkan *bandwidth*

dengan hasil maksimal. Sebagai gambaran metode yang digunakan saat ini, dapat penulis gambarkan seperti berikut ini :

Semisal sebuah WARNET memanfaatkan 2 (dua) jaringan Internet, kita namai dengan jaringan A dan jaringan B yang mana masing-masing memiliki *bandwidth* sebesar 1 Mb/s. WARNET tersebut memiliki 10 (sepuluh) unit perangkat komputer client, maka untuk pemerataan *bandwidth* sang pemilik warnet membagi 10 (sepuluh) komputer client tersebut menjadi 2 (dua) blok yang mana masing-masing blok terdiri dari 5 (lima) unit komputer client, maka pada blok A diberikan 1 (satu) jaringan Internet yaitu jaringan A dan untuk blok B mendapatkan jaringan B. Dari sini kita dapatkan asumsi yaitu untuk 1 (satu) jaringan yang memiliki *bandwidth* sebesar 1 Mb/s dipergunakan untuk 5 (lima) unit komputer. Semisal ketika blok A terisi oleh 2 (dua) pengguna dan blok B terisi oleh 5 (lima) pengguna, secara tidak langsung telah terjadi ketidak seimbangan kecepatan akses, dimana pada blok A yang dialokasikan *bandwidth* sebesar 1 Mb/s hanya digunakan oleh 2 orang ( $1 \text{ Mb/s} : 2$ ) sedangkan untuk blok B yang dialokasikan *bandwidth* sama seperti blok A yaitu 1 Mb/s digunakan oleh 5 orang ( $1 \text{ Mb/s} : 5$ ), kita dapatkan pada blok A masing-masing pengguna dapat menikmati kecepatan akses sebesar 512 Kb/s dan untuk blok B masing-masing pengguna dapat menikmati kecepatan akses sebesar 204,8 Kb/s. Hal seperti yang penulis contohkan tersebut lebih dikenal dengan metode *grouping*.



Berdasarkan pada hal tersebut, penulis membuat **Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix** ini dengan tujuan membantu mempermudah para pengguna jasa layanan Internet khususnya pada kalangan sekolah, perkantoran dan penyedia layanan warung Internet untuk dapat memaksimalkan *bandwidth* Internet yang mereka gunakan. Para pengguna jasa layanan Internet dapat memanfaatkan teknik penggabungan *bandwidth* ini agar mendapatkan akselerasi yang maksimal. Dengan demikian diharapkan masyarakat merasakan manfaat dari alat tersebut diatas.

## **B. Identifikasi Masalah**

Dalam pembuatan proyek akhir dengan judul Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini, maka masalah-masalah yang terkait dengan proyek akhir ini dapat diidentifikasi sebagai berikut :

1. Metode penggabungan *bandwidth downstream (Load Balancing)* untuk kepentingan Perkantoran, Sekolah, Warung Internet yang digunakan saat ini belum banyak digunakan.
2. Banyaknya para pengguna jasa layanan Internet yang belum memahami metode penggabungan *bandwidth downstream (Load Balancing)*.
3. Metode penggabungan *bandwidth downstream (Load Balancing)* yang dilakukan pada umumnya belum optimal.
4. Banyaknya para pengguna jasa layanan Internet yang menggunakan lebih dari 1 jaringan Internet dan masih menggunakan metode *grouping*.

5. Metode penggabungan *bandwidth downstream* (*Load Balancing*) yang ada saat ini relatif mahal.

### **C. Batasan Masalah**

Mengingat keterbatasan ilmu pengetahuan, waktu, dana, dan tenaga dalam mengerjakan Proyek Akhir ini, maka penulis membatasi permasalahannya pada :

- Penggabungan 2 jalur Internet.
- Penggunaan software MikroTik sebagai penggabung *bandwidth downstream* (*Load Balancing*).
- Konfigurasi software MikroTik sebagai penggabung *bandwidth downstream* (*Load Balancing*).

### **D. Rumusan Masalah**

Berdasarkan uraian latar belakang masalah diatas dapat dirumuskan permasalahannya sebagai berikut :

1. Bagaimana konfigurasi perangkat keras Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix?
2. Bagaimana konfigurasi pemrograman MikroTik pada Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix?
3. Bagaimana unjuk kerja dari Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix?

## **E. Tujuan**

Pembuatan Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix bertujuan untuk :

1. Mengetahui dan mampu mengkonfigurasi *hardware* (perangkat keras) dan *software* (perangkat lunak) Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix.
2. Merealisasikan konfigurasi program MikroTik pada Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix tersebut.
3. Mengetahui unjuk kerja Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix tersebut.

## **F. Manfaat**

Dalam pembuatan Proyek Akhir ini diharapkan terpenuhi beberapa manfaat sebagai berikut, antara lain:

1. Bagi Mahasiswa, dapat menambah bekal dan pengalaman untuk mengaplikasikan ilmu pengetahuan yang didapat selama duduk di bangku kuliah ke dalam dunia kerja.
2. Bagi Jurusan Pendidikan Teknik Elektronika, sebagai tolok ukur daya serap mahasiswa yang bersangkutan selama menempuh pendidikan dan kemampuan menerapkan ilmunya secara praktis.
3. Bagi Dunia usaha khususnya yang memanfaatkan jasa layanan Internet lebih dari satu jaringan diharapkan metode ini dapat membantu memaksimalkan akselerasi akses Internet yang digunakan.

## G. Keaslian Gagasan

Gagasan pembuatan Proyek Akhir dengan judul Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini muncul berdasarkan dari Skripsi Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, 2009 dengan judul Analisis dan Implementasi Load Balance Dua ISP Menggunakan MikroTik Dengan Metode Round Robin yang ditulis oleh Agus Heriyanto yang masih menggunakan metode *Fail Over* dinilai masih kurang efektif, sebab dengan menggunakan metode tersebut *bandwidth downstream* belum dapat digabungkan dalam waktu bersamaan. Untuk itu penulis membuat Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix dengan metode PCC yang sudah dapat menggabungkan 2 (dua) buah *bandwidth downstream* dalam waktu yang bersamaan dan dengan hasil yang maksimal.

## BAB II

### KAJIAN TEORI

#### A. LOAD BALANCING

Penggabungan *bandwidth downstream* atau yang lebih familiar dengan *Load Balancing (LB)* adalah sebuah konsep yang gunanya untuk menyeimbangkan beban *bandwidth* dari ke dua atau lebih jaringan yang ada atau muatan pada infrastruktur TI dalam hal ini jaringan Internet agar seluruh kemampuan yang dimiliki dapat dimanfaatkan secara maksimal dan optimal. Jadi tiap jaringan dapat dimanfaatkan secara efisien. Alur trafik akan diatur sedemikian rupa berdasarkan beban masing-masing koneksi.

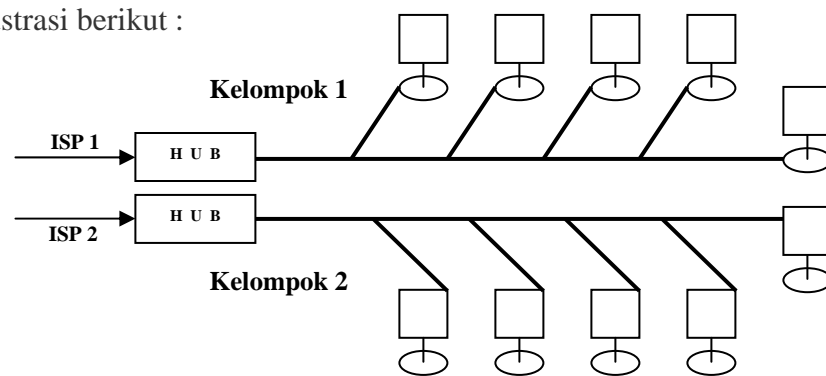
*Load Balance* pada MikroTik adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi.

Syarat mutlak yang harus dipenuhi guna terciptanya sebuah metode Load Balance ada 2 (dua), yang pertama yaitu harus memiliki jaringan internet lebih dari 1 (satu) dan syarat yang kedua adalah kedua jaringan tersebut harus memiliki gateway yang berbeda.

*Load Balancing* sendiri memiliki beberapa jenis dan metode yang umum digunakan, yaitu :

## 1. Grouping

Adalah teknik pembagian *bandwidth downstream* berdasarkan jumlah komputer klien atau bias disebut juga dengan metode pengelompokan klien. Algoritma dari metode *grouping* ini sendiri adalah 2 (dua) jaringan dengan masing-masing *bandwidth* dari ISP digunakan untuk 10 (sepuluh) komputer klien dimana dari 10 (sepuluh) komputer klien tersebut dibagi menjadi 2 (dua) kelompok, untuk kelompok pertama berisi 5 (lima) unit komputer klien dan diberikan jaringan dari ISP pertama dan untuk komputer klien yang satu juga berisi 5 (lima) komputer klien dan diberikan jaringan dari ISP kedua. Untuk lebih jelasnya dapat dilihat pada ilustrasi berikut :



Gambar 1. Ilustrasi Metode *Grouping*

Dari ilustrasi gambar di atas dapat di lihat bahwa pada masing-masing kelompok terdapat 5 (lima) unit komputer klien yang mana tiap kelompok dialokasikan 1 (satu) jaringan internet yaitu dari ISP 1 dan ISP 2. Keuntungan dari metode *grouping* ini adalah jika salah satu jaringan internet dari ISP 1 atau dari ISP 2 mati, maka jaringan yang lain masih bisa

digunakan. Sedangkan untuk kelemahan dari metode tersebut yaitu pembagian bandwidth tidak efektif, sebagai contoh misalkan dari kelompok komputer pertama digunakan oleh 2 pengguna dan untuk kelompok komputer kedua digunakan oleh 5 pengguna, misal *bandwidth* internet dari ISP 1 dan ISP 2 masing-masing adalah 1Mb/s maka tiap pengguna komputer kelompok pertama mendapatkan *bandwidth* kisaran 512 Kb/s sedangkan pada kelompok kedua hanya mendapatkan *bandwidth* kisaran 204,8 Kb/s per orang.

## 2. *Fail over*

*Fail Over* adalah merupakan sebuah teknik pengalihan jaringan dimana jika ada salah satu atau beberapa line yang mati, maka trafik jaringan akan dialihkan ke line yang masih hidup (*online*). Keuntungan menggunakan metode ini yaitu tersedianya backup jaringan yaitu ketika koneksi utama terputus maka koneksi internet masih bisa digunakan karena secara otomatis koneksi cadangan akan mengambil alih kinerja dari koneksi utama dan metode seperti ini biasanya digunakan oleh para penyedia layanan Game Center untuk mengantisipasi putusnya koneksi internet ketika sedang digunakan. Sedangkan untuk kekurangan dari metode ini yaitu jika koneksi utama tidak mengalami gangguan atau terputus maka bandwidth cadangan akan sia-sia karena tidak digunakan.

### 3. *PCC*

Adalah metode penggabungan jaringan yang disertai dengan kemampuan menandai paket data untuk dilewatkan ke beberapa alur koneksi, namun di akhir penerimaan data akan dikenali sebagai 1 (satu) paket data. Beberapa keuntungan menggunakan metode ini yaitu berapapun koneksi yang kita gunakan, maka dapat digabungkan sehingga pada output didapatkan satu jaringan utuh dari hasil penggabungan jaringan yang telah dilakukan. Selain itu dengan menggunakan metode ini ketika kita mengakses ke suatu website yang mana kita harus login ke website tersebut, dimana website yang kita tuju memiliki sistem ketika IP yang digunakan berubah, maka secara otomatis kita diminta untuk melakukan login ulang sebab sistem mendeteksi IP yang digunakan telah berubah, maka hal itu tidak akan terjadi sebab kita koneksi tidak akan dianggap putus walau menggunakan beberapa koneksi. Sebagai contoh website yang dimaksudkan adalah : <https://www.dmehosting.com/clients/clientarea.php>. Untuk kekurangannya sendiri sejauh ini menggunakan metode tersebut penulis belum menemukan kelemahan dari metode PCC ini.



## B. Perangkat Keras (*Hardware*)

### 1. KOMPUTER



Gambar 2. Unit Komputer

[http://queenqolbu.files.wordpress.com/2009/03/icom\\_-\\_komputer\\_-\\_1.jpg](http://queenqolbu.files.wordpress.com/2009/03/icom_-_komputer_-_1.jpg)

Komputer adalah alat yang dipakai untuk mengolah data menurut perintah yang telah dirumuskan. Kata komputer semula dipergunakan untuk menggambarkan orang yang pekerjaannya melakukan perhitungan aritmatika, dengan atau tanpa alat bantu, tetapi arti kata ini kemudian dipindahkan kepada mesin itu sendiri. Asal mulanya, pengolahan informasi hampir eksklusif berhubungan dengan masalah aritmatika, tetapi komputer modern dipakai untuk banyak tugas yang tidak berhubungan dengan matematika.

Secara luas, Komputer dapat didefinisikan sebagai suatu peralatan elektronik yang terdiri dari beberapa komponen, yang dapat bekerja sama antara komponen satu dengan yang lain untuk menghasilkan suatu informasi berdasarkan program dan data yang ada. Adapun komponen komputer adalah meliputi : Layar

Monitor, CPU, *Keyboard*, *Mouse* dan *Printer* (sebagai pelengkap). Tanpa *printer* komputer tetap dapat melakukan tugasnya sebagai pengolah data, namun sebatas terlihat dilayar monitor belum dalam bentuk *print out* (kertas).

Dalam definisi seperti itu terdapat alat seperti *slide rule*, jenis kalkulator mekanik mulai dari *abakus* dan seterusnya, sampai semua komputer elektronik yang kontemporer. Istilah lebih baik yang cocok untuk arti luas seperti "komputer" adalah "yang memproses informasi" atau "sistem pengolah informasi." Selama bertahun-tahun sudah ada beberapa arti yang berbeda dalam kata "komputer", dan beberapa kata yang berbeda tersebut sekarang disebut sebagai komputer.

Kata komputer secara umum pernah dipergunakan untuk mendefinisikan orang yang melakukan perhitungan aritmatika, dengan atau tanpa mesin pembantu. Menurut *Barnhart Concise Dictionary of Etymology*, kata tersebut digunakan dalam bahasa Inggris pada tahun 1646 sebagai kata untuk "orang yang menghitung" kemudian menjelang 1897 juga digunakan sebagai "alat hitung mekanis". Selama Perang Dunia II kata tersebut menunjuk kepada para pekerja wanita Amerika Serikat dan Inggris yang pekerjaannya menghitung jalan artileri perang dengan mesin hitung.

([http://id.wikipedia.org/wiki/Sejarah\\_komputer](http://id.wikipedia.org/wiki/Sejarah_komputer),

<http://id.wikipedia.org/wiki/Komputer>)

Dengan menggunakan unit komputer inilah nantinya penggabungan *bandwidth* diproses, komputer ini dilengkapi dengan 3 buah LAN CARD yang berfungsi untuk meng-*interface*-kan MODEM. Agar dapat digunakan, komputer ini harus di-*install* program MikroTik terlebih dahulu.

## 2. MODEM ADSL



Gambar 3. MODEM ADSL Aztech model DSL605E

ADSL atau *Asymmetric Digital Subscriber Line* adalah salah satu bentuk dari teknologi DSL. Ciri khas ADSL adalah sifatnya yang asimetrik, yaitu bahwa data ditransferkan dalam kecepatan yang berbeda dari satu sisi ke sisi yang lain.

Sebelum ADSL, kita sudah terlebih dulu mengenal sistem yang disebut *dial-up*. Sistem ini menggunakan sambungan kabel telepon sebagai jaringan penghubung dengan *Internet Service Provider* (ISP). Namun dalam penggunaannya, *dial-up* memiliki beberapa kekurangan. Seperti rendahnya kecepatan dalam mengakses Internet, terlebih di jam-jam tertentu yang merupakan waktu sibuk atau *office hour*. Selain itu, karena menggunakan sambungan telepon, kita tidak bisa menggunakan telepon bila sedang melakukan koneksi Internet. Penggunaan sambungan telepon juga memungkinkan tingginya tingkat gangguan atau *noise* bila sedang menggunakan Internet. Kekurangan lainnya adalah sistem penghitungan *dial-up* yang masih berdasarkan waktu dan masih dirasakan sangat mahal.

ADSL sendiri merupakan salah satu dari beberapa jenis DSL, disamping SDSL, GHDSL, IDSL, VDSL, dan HDSL. DSL merupakan teknologi akses Internet menggunakan kabel tembaga, sering disebut juga sebagai teknologi suntikan atau *injection technology* yang membantu kabel telepon biasa dalam menghantarkan data dalam jumlah besar. DSL sendiri dapat tersedia berkat adanya sebuah perangkat yang disebut DSLAM (*DSL Acces Multiplexer*). Untuk mencapai tingkat kecepatan yang tinggi, DSL menggunakan sinyal frekuensi hingga 1 MHz. Lain halnya untuk ADSL, sinyal frekuensi yang dipakai hanya berkisar antara 20 KHz sampai 1 MHz. Sementara ini untuk penggunaan ADSL di Indonesia baru diterapkan oleh PT. TELKOM dengan program Telkom *Speedy*, kecepatan yang ditawarkan berkisar antara 1024 Kb/s untuk *downstream* dan 128 Kb/s untuk *upstream*.

Perkenalan masyarakat Indonesia sendiri akan ADSL mulai berkembang saat PT. TELKOM, yang merupakan perusahaan pengatur jaringan telepon nasional memperkenalkan program yang disebut sebagai *Telkom Speedy*, yaitu jaringan khusus dari PT. TELKOM untuk penggunaan Internet.

Adapun cara-cara penggunaan ADSL di Indonesia, pertama-tama kita terlebih dahulu harus memiliki perangkat ADSL. Setelah memiliki perangkat ADSL, kita harus memeriksa keberadaan nomor telepon rumah kita di layanan *Telkom Speedy*, apakah sudah terdaftar atau belum. Selanjutnya yang harus diperhatikan adalah, seberapa jauh jarak antara gardu Telkom dengan rumah kita. Karena dalam ADSL, jarak sangat berpengaruh pada kecepatan koneksi Internet. Setelah memastikan bahwa nomor telepon sudah terdaftar dan jarak sudah diperhitungkan,

yang harus kita lakukan selanjutnya adalah pemasangan ADSL pada sambungan telepon.

Untuk menyambungkan antara ADSL dengan *line* telepon, kita menggunakan sebuah alat yang disebut sebagai *Splitter* atau pembagi *line*. *Splitter* ini berguna untuk menghilangkan gangguan ketika kita menggunakan modem ADSL. Sehingga nantinya kita tetap dapat menggunakan Internet dan menjawab telepon secara bersamaan.

ADSL sendiri memiliki bermacam-macam jenis dengan kecepatan, jenis *router*, USB dan perangkat lain yang ada di dalamnya. Misalnya ada yang dapat dipakai untuk dua komputer dengan menggunakan sambungan USB, tapi ada juga yang dapat digunakan untuk empat komputer dengan koneksi *LAN Ethernet*. Namun ada baiknya dalam memilih modem ADSL, kita memilih menggunakan modem yang memiliki tombol on dan off. Hal ini dimaksudkan supaya kita dapat mengatur penggunaan koneksi sebanyak yang kita butuhkan dan menghemat biaya koneksi yang digunakan. Terlebih di Indonesia masih menggunakan penghitungan waktu (*Time Base*) atau banyaknya *bandwidth* yang digunakan (*Quota / Volume Base*).

Hal penting lain yang dimiliki oleh modem ADSL adalah adanya lampu indikator yang berguna mengetahui jalannya proses koneksi yang terjadi. Umumnya lampu yang ada pada modem ADSL adalah lampu PPP, *Power*, DSL. Ada juga lampu tambahan bila kita menggunakan koneksi *Ethernet* dan USB.

Dari tiga lampu indikator yang ada pada modem, yang terpenting adalah lampu PPP dan DSL. Di mana lampu DSL menunjukkan koneksi sudah terhubung dengan baik pada *line*. Sementara lampu PPP menunjukkan adanya arus data ketika

seseorang melakukan *browsing*. Setelah perangkat lengkap, hal yang penting dalam penggunaan ADSL di Indonesia adalah penggunaan IP modem dan *password*. Hal ini digunakan untuk melindungi penggunaan layanan bagi konsumen yang diberikan oleh *provider*. IP yang kita miliki akan menjadi gerbang untuk memasuki jaringan. Jika kita mengubah *password* untuk *login*, maka kita perlu memasukkan kembali sesuai perubahan yang dilakukan. Bila seluruh proses ini berhasil dilalui, maka selanjutnya kita sudah dapat berkoneksi Internet dengan ADSL.

Penggunaan ADSL di Indonesia saat ini tidak hanya berkisar hanya di pulau Jawa saja, tapi juga sudah meluas sampai ke luar Jawa. Seperti Bali dan Sumatera. Walaupun kualitas yang ditawarkan memang masih banyak mengalami masalah, namun adanya ADSL dalam berkoneksi Internet sangatlah membantu dibandingkan dengan cara lama yang menggunakan sistem *dial-up*.

### **Kelebihan ADSL**

- Pembagian *frekuensi* menjadi dua, yaitu *frekuensi* tinggi untuk menghantarkan data, sementara *frekuensi* rendah untuk menghantarkan suara dan fax.
- Bagi pengguna di Indonesia yang memakai program *Speedy*, penggunaan ADSL membuat kegiatan Internet menjadi jauh lebih murah. Sehingga kita dapat ber-Internet tanpa khawatir dengan tagihan yang membengkak.

### **Kekurangan ADSL**

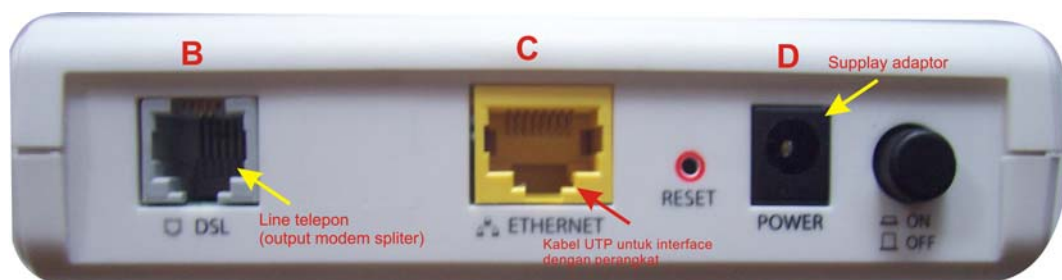
Adapun kualitas dari ADSL saat ini masih memiliki kekurangan, diantaranya adalah sebagai berikut :

- Seperti sangat berpengaruhnya jarak pada kecepatan pengiriman data. Semakin jauh jarak antara modem dengan PC, atau saluran telepon kita dengan gardu telepon, maka semakin lambat pula kecepatan mengakses Internetnya.
- Tidak semua *software* dapat menggunakan modem ADSL semisal Mac. Cara yang dipakai pun akan lebih rumit dan ada kemungkinan memakan waktu lama, tapi pada MODEM ADSL jenis terbaru management MODEM dapat dilakukan via *web interface* sehingga tingkat kompatibilitasnya meningkat dan menjadikan modem ADSL dapat digunakan pada setiap jenis PC selama PC bersangkutan memiliki *ethernet card*.
- Adanya *load coils* yang dipakai untuk memberikan layanan telepon ke daerah-daerah, sementara *load coils* sendiri adalah peralatan induksi yang menggeser *frekuensi* pembawa ke atas. Sayangnya *load coils* menggeser *frekuensi* suara ke *frekuensi* yang biasa digunakan DSL. Sehingga mengakibatkan terjadinya interferensi dan ketidakcocokkan jalur untuk ADSL.
- Adanya *Bridged tap*, yaitu bagian kabel yang tidak berada pada jalur yang langsung antara pelanggan dan CO. *Bridged tap* ini dapat menimbulkan *noise* yang mengganggu kinerja DSL.

- Penggunaan *fiber optic* pada saluran telepon digital yang dipakai saat ini. Di mana penggunaan *fiber optic* ini tidak sesuai dengan sistem ADSL yang masih menggunakan saluran analog yaitu kabel tembaga, sehingga akan sulit dalam pengiriman sinyal melalui *fiber optic*.
- Kecepatan koneksi MODEM ADSL masih tergantung dengan jarak tiang Telkom atau DSLAM terdekat, artinya jika jarak MODEM ADSL dengan DSLAM jauh maka kecepatan koneksi akan menurun karena banyaknya hambatan medium yang dilaluinya dan sebaliknya jika jaraknya dekat, koneksinya akan mencapai kecepatan yang diharapkan.

(<http://id.wikipedia.org/wiki/ADSL>)

Seperti yang sudah dijelaskan di atas, MODEM ADSL ini termasuk dalam golongan MODEM digital, berbeda dengan MODEM yang biasa terpasang pada PC (*Personal Komputer*) yakni MODEM *analog* yang berbentuk *card*. MODEM ADSL berfungsi untuk merubah sinyal *inputan* yang masih berupa sinyal *analog* sehingga *output* yang dihasilkan MODEM ADSL tersebut sebelum masuk ke PC sudah dalam bentuk *digital*.



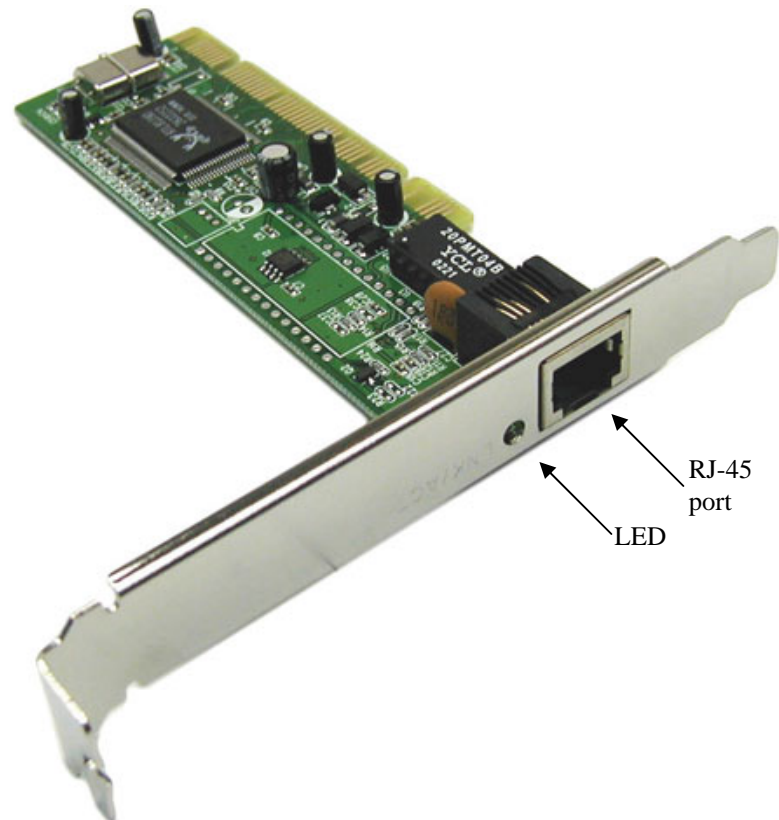
Gambar 4. *Back panel* MODEM ADSL Aztech model DSL605E



Keterangan :

- Soket B yang terletak pada sebelah kiri (DSL) adalah merupakan *input* dari MODEM ADSL dari *line* telepon menggunakan konektor RJ11.
- Soket C yang berwarna kuning (*ETHERNET*) adalah merupakan *output* dari MODEM ADSL. Soket inilah yang digunakan untuk meng-*interface*-kan jaringan Internet ke PC dengan memanfaatkan kabel UTP yang dilengkapi dengan konektor RJ45.
- Soket D (*POWER*) terhubung dengan adaptor sebagai *supply* tegangan MODEM.
- Terdapat pula saklar ON / OFF yang digunakan untuk memutus atau mengalirkan aliran listrik dari adaptor ke MODEM.
- MODEM ADSL juga dilengkapi dengan tombol *reset*. Tombol ini digunakan ketika terjadi suatu masalah pada settingan MODEM tersebut, sehingga setelah direset settingan MODEM akan kembali seperti semula (*default*).

### 3. LAN CARD



Gambar 5. LAN Card

(<http://www.pooyeshnetwork.com/images/lan%20card.jpg>)

*LAN CARD* merupakan penghubung antara *system* komputer dengan kinerja komputer lain atau dapat dikatakan *LAN CARD* merupakan sinyal untuk saling berbagi antara komputer satu dengan komputer yang lain.

Prinsip kerja *LAN CARD* adalah menerima sinyal dari komputer lain kemudian mentransmisikan kedalam masukan kemudian diolah menjadi data begitu sebaliknya saling berbagi. *LAN CARD* dapat digunakan untuk menghubungkan *system* komputer satu dengan komputer lain melalui perantara HUB sehingga dalam area tersebut membentuk suatu jaringan komputer.

(<http://blog.uad.ac.id/kartikaf/files/2009/11/A09-8-LAN-Card.pdf>)

## C. Perangkat Lunak (*Software*)

### 1. MIKROTIK

#### a. Sejarah MikroTik

MikroTik adalah sebuah perusahaan kecil berkantor pusat di Latvia, bersebelahan dengan Rusia. Pembentukannya diprakarsai oleh John Trully dan Arnis Riekstins. John Trully adalah seorang berkewarganegaraan Amerika yang bermigrasi ke Latvia. Di Latvia ia bejumpa dengan Arnis, Seorang darjana Fisika dan Mekanik sekitar tahun 1995.

John dan Arnis mulai me-routing dunia pada tahun 1996 (misi MikroTik adalah merouting seluruh dunia). Mulai dengan sistem Linux dan MS-DOS yang dikombinasikan dengan teknologi *Wireless-LAN* (WLAN) Aeronet berkecepatan 2 Mbps di Moldova, negara tetangga Latvia, baru kemudian melayani lima pelanggannya di Latvia. Prinsip dasar mereka bukan membuat *Wireless ISP* (W-ISP), tetapi membuat program router yang handal dan dapat dijalankan diseluruh dunia. Latvia hanya merupakan tempat eksperimen John dan Arnis, karena saat ini mereka sudah membantu negara-negara lain termasuk Srilanka yang melayani sekitar 400 pengguna.

Linux yang pertama kali digunakan adalah Kernel 2.2 yang dikembangkan secara bersama-sama dengan bantuan 5-15 orang staff Research and Development (R&D) MikroTik yang sekarang menguasai dunia routing di negara-negara berkembang. Menurut

Arnis, selain staf di lingkungan MikroTik, mereka juga merekrut tenaga-tenaga lepas dan pihak ketiga yang dengan intensif mengembangkan MikroTik secara marathon.

#### **b. Pengertian Dasar**

Saat ini sudah banyak system operasi yang dapat digunakan sebagai router, mulai yang paling sederhana hingga yang sangat canggih. Dari sekian banyak system operasi tersebut yang paling populer dan mulai banyak digunakan adalah MikroTik. MikroTik mudah digunakan, dan sangat canggih sehingga tidak memerlukan kemampuan teknis yang tinggi, sehingga para pemula pun akan mudah untuk menggunakannya.

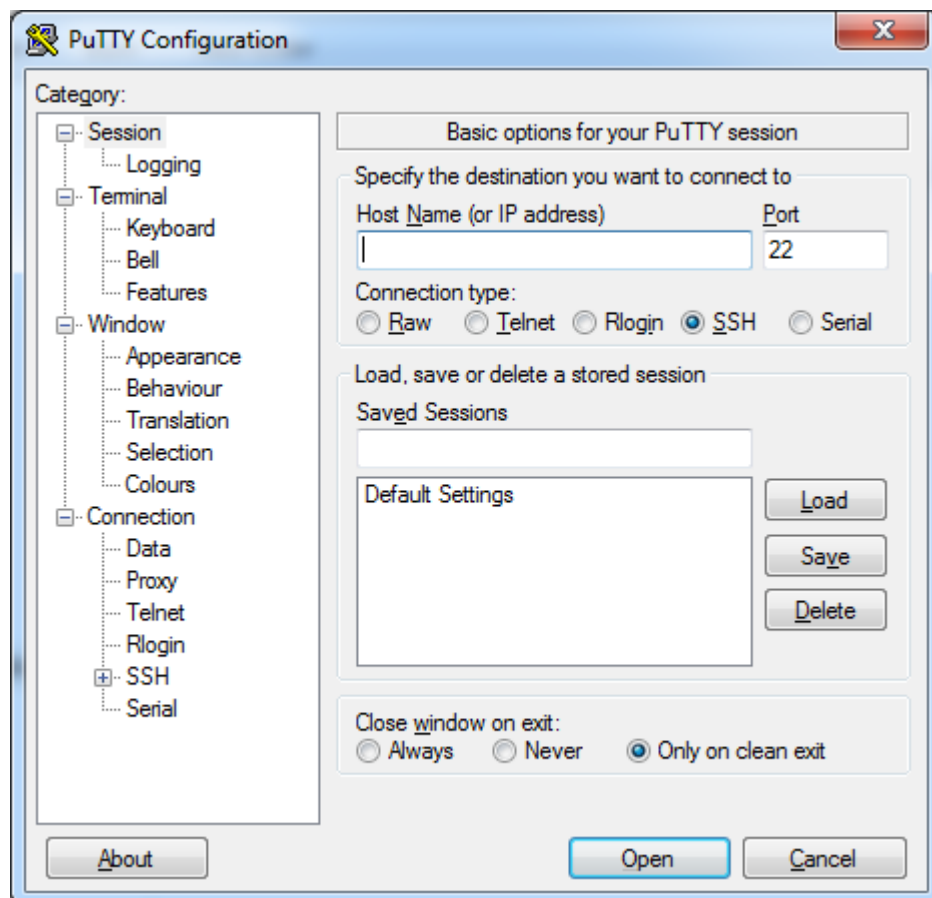
MikroTik dapat digunakan dalam 2 tipe, yaitu dalam bentuk perangkat keras dan perangkat lunak, dimana keduanya terpasang secara sinkron agar dapat bekerja dengan baik. Dalam bentuk perangkat keras, MikroTik biasanya sudah diinstalasi pada suatu board tertentu, sedangkan dalam bentuk perangkat lunak, MikroTik merupakan satu distro Linux yang memang dikhususkan untuk fungsi router. MikroTik routerOS adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer biasa menjadi router network yang handal, mencakup berbagai fitur yang dibuat untuk *network* dan jaringan *wireless*.

### c. Metode Konfigurasi

Ada 4 cara pengaksesan MikroTik Router, antara lain :

#### 1. Via Console/Command MikroTik

Jenis Router Board maupun PC bisa kita akses langsung via *console* / *shell* maupun remote akses menggunakan PuTTY ([www.putty.nl](http://www.putty.nl))



Gambar 6. Tampilan PuTTY

Tips Command : "Manfaatkan *auto complete*" (mirip bash *auto complete* di linux) tekan tombol TAB di keyboard untuk mengetahui/melengkapi daftar perintah selanjutnya. Jadi perintah yang panjang tidak perlu kita ketik

lagi, cukup ketikkan awal perintah itu, lalu tekan TAB-TAB maka otomatis Shell akan menampilkan / melengkapi daftar perintah yang kita maksud.

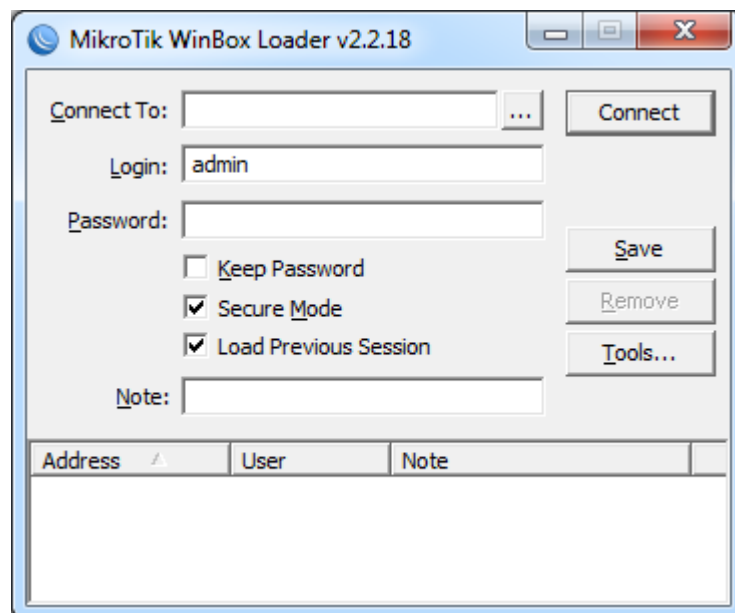
Contoh: Cukup ketikkan Ip Fir >>> lalu tekan TAB >>> maka otomatis shell akan melengkapi menjadi Ip Firewall. Lalu ketik “.” (titik dua) untuk kembali ke sub menu di atasnya, dan ketik “/” untuk kembali ke root menu.

## 2. Via Web Browser

Mikrotik bisa juga diakses via web/port 80 pada browser. Contoh : ketik di browser IP mikrotik kita: 192.168.0.18.

## 3. Via WinBox

Mikrotik bisa juga diakses/remote menggunakan *tool winbox* (utility kecil di windows yang sangat praktis dan cukup mudah digunakan). Tampilan awal mengaktifkan WinBox seperti ini :



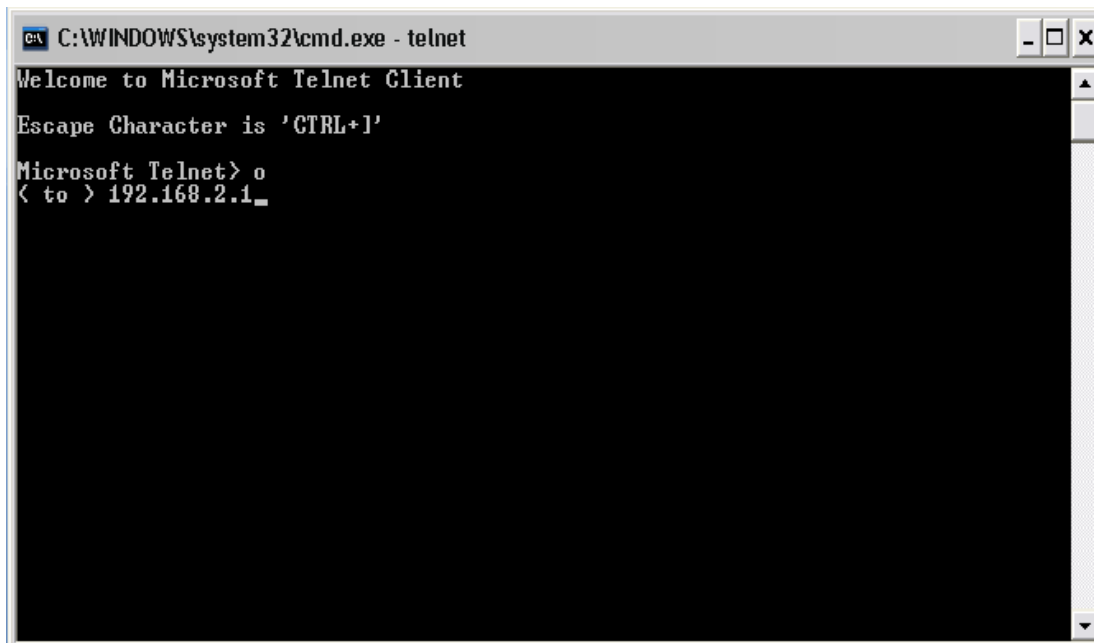
Gambar 7. Tampilan WinBox

Winbox bisa mendeteksi mikrotik yang sudah di install asal masih dalam satu network, yaitu dengan mendeteksi MAC address dari ethernet yang terpasang di MikroTik.

#### 4. Via Telnet

Kita dapat me-remote MikroTik menggunakan telnet melalui program aplikasi "command prompt" (cmd) yang ada pada windows. Namun, penggunaan telnet tidak dianjurkan dalam jaringan karena masalah keamanannya.

Contoh : `C:\>telnet 192.168.2.1`



```
C:\WINDOWS\system32\cmd.exe - telnet
Welcome to Microsoft Telnet Client
Escape Character is 'CTRL+I'
Microsoft Telnet> o
< to > 192.168.2.1_
```

Gambar 8. Contoh tampilan *Telnet via Command Prompt (CMD)*



### **BAB III**

#### **PERANCANGAN ALAT**

##### **A. Identifikasi Kebutuhan *Hardware* dan *Software***

Dalam pengembangan Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix harus mempertimbangkan beberapa kebutuhan dalam hal ini adalah perangkat-perangkat penyusun yang sangat berpengaruh terhadap tercapainya satu tujuan, antara lain:

1. Perlunya fasilitas perangkat yang dapat digunakan untuk menjaga kestabilan tegangan pada komputer *server*.
2. Perlunya perangkat pengendali untuk mengendalikan sistem secara keseluruhan.
3. Perlunya perangkat untuk interface dari MODEM ke komputer.
4. Perlunya perangkat MODEM ADSL dengan kualitas baik untuk menjaga kestabilan koneksi Internet.
5. Perlunya perangkat untuk mengantisipasi padamnya aliran listrik.
6. Perlunya perangkat untuk menguji output server.
7. Perlunya perangkat untuk menjaga kestabilan kinerja system.
8. Perlunya perangkat untuk memperlancar proses konfigurasi.
9. Perlunya kabel UTP beserta konektor RJ-45.

## **B. Analisis Kebutuhan *Hardware* dan *Software***

Berdasarkan identifikasi kebutuhan yang ada, maka diperoleh beberapa analisis kebutuhan terhadap alat yang dibuat dengan spesifikasi sebagai berikut:

1. Menggunakan *stabilizer* / stavolt 600 VA sebagai penstabil tegangan PLN.
2. Menggunakan satu unit komputer / laptop untuk memonitor dan mengendalikan komputer *server*.
3. Menggunakan LAN CARD dengan kemampuan transfer data 10 Mb/s.
4. Menggunakan MODEM ADSL Aztech model DSL605E atau sejenis.
5. Menggunakan *UPS PROLINK* 600 VA.
6. Menggunakan satu unit komputer yang berlaku sebagai klien.
7. Menggunakan RAM KINGSTONE dengan kapasitas minimum 1 GB.
8. Menggunakan kabel UTP dengan merk BELDEN.
9. Menggunakan *cutter*, gunting dan juga *Crimping Tools* untuk memasang konektor RJ-45

### C. Alasan Pemilihan Komponen

Dalam proses pembuatan Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini mempunyai beberapa alasan pada pemilihan komponen yang digunakan, antara lain:

1. *Stabilizer* / stavolt 600 VA
  - ✓ Mudah untuk didapatkan.
  - ✓ Mudah untuk digunakan dan dibawa kemanapun.
  - ✓ Di dalamnya terdapat rangkaian yang berfungsi untuk menstabilkan tegangan walaupun aliran listrik dari PLN tidak stabil.
2. LAN CARD 10 Mb/s
  - ✓ Merupakan perangkat interface yang menghubungkan MODEM ADSL dengan komputer dengan perantara kabel UTP.
  - ✓ Mempunyai kemampuan transfer data 10 Mb/s.
  - ✓ Mudah untuk didapatkan.
3. MODEM ADSL Aztech model DSL605E
  - ✓ Merupakan unit yang mudah untuk didapatkan.
  - ✓ Unit tersebut mudah dalam penggunaan dan pengkonfigurasian.
  - ✓ Kestabilan unit tersebut cukup untuk memenuhi kebutuhan.
4. UPS PROLINK 600 VA
  - ✓ Merupakan unit yang mampu menyimpan cadangan energi untuk mempertahankan komputer *server* tetap dalam kondisi menyala selama kurang lebih 15 menit meskipun aliran listrik dari PLN terputus.
  - ✓ Mudah untuk didapatkan.

- ✓ Mempunyai rangkaian penstabil tegangan didalamnya.

#### 5. RAM KINGSTONE 1 GB

- ✓ Mudah untuk didapatkan.
- ✓ Merupakan memory yang cukup handal untuk proses kinerja yang berat.
- ✓ Memiliki kemampuan pemrosesan data yang cukup cepat.

#### 6. Kabel UTP BELDEN

- ✓ Merupakan kabel UTP dengan kualitas yang unggul dari segi kelenturan dan transfer data.
- ✓ Loss data relatif lebih sedikit.
- ✓ Mudah untuk didapatkan.

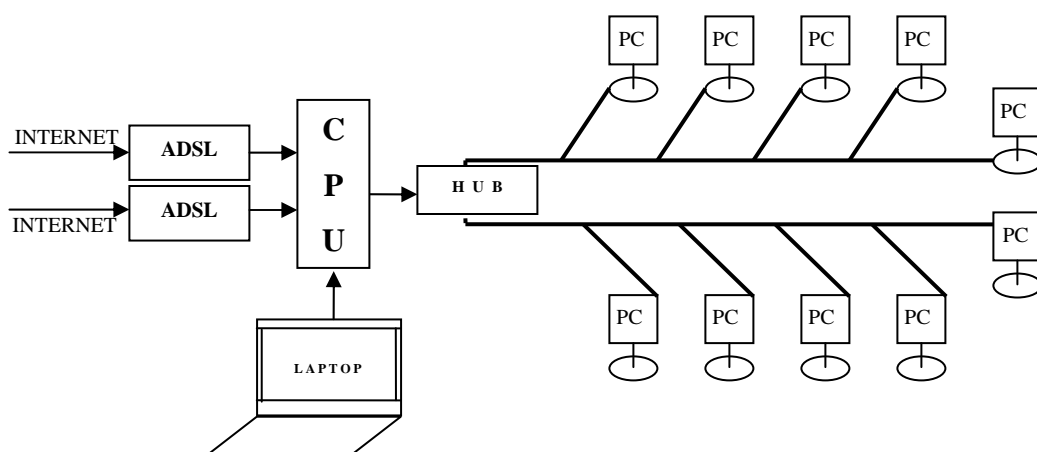
#### 7. LAPTOP

- ✓ Mudah dibawa kemanapun guna melakukan pengaturan / pemeliharaan server.
- ✓ Memiliki interface yang cukup banyak.

## D. Konsep Rancangan

### 1. Perancangan Perangkat Keras (*Hardware*)

Dalam pembuatan sebuah sistem yang lengkap harus mengetahui terlebih dahulu gambaran umum tentang rancangan apa yang akan dibuat. Secara garis besar model rancangan Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini dapat digambarkan sebagai berikut:



Gambar 9. Blok Perancangan *Hardware*

Dalam gambar blok diagram Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix tersebut di atas, menggunakan metode PCC dengan struktur perangkat keras (*hardware*) terbagi dalam beberapa kelompok skema rangkaian yang masing-masing mempunyai fungsi yang berbeda-beda tetapi saling berhubungan antara satu dengan yang lain, yaitu:

### **a. Koneksi Internet**

Internet disini yang dimaksudkan adalah 2 (dua) buah koneksi Internet dengan *provider* berbeda / *subnet* IP berbeda dan memiliki *bandwidth* yang cukup besar yaitu kisaran 512 Kb/s atau juga bias lebih menyesuaikan kebutuhan. Kedua koneksi Internet inilah yang nantinya diproses untuk digabungkan menjadi 1 (satu) *bandwidth* utuh / satu jaringan.

### **b. MODEM ADSL**

MODEM ADSL ini yang berfungsi untuk merubah sinyal analog yang dilewatkan melalui jaringan telepon yang selanjutnya diproses dan dikeluarkan dari MODEM sudah dalam bentuk *digital*. Jika tidak menggunakan koneksi Internet yang memanfaatkan jaringan telepon, maka biasanya sudah dalam bentuk sinyal *digital*. Hal ini dapat kita ketahui dari *output* jaringan Internet tersebut apakah sudah didistribusikan menggunakan kabel UTP atau masih menggunakan kabel telepon, jika sudah menggunakan kabel UTP maka dapat langsung digunakan.

### **c. CPU / Komputer Server**

CPU disini berfungsi sebagai *router* atau bias kita sebut juga sebagai komputer *server*. Pada unit inilah proses penggabungan *bandwidth* berlangsung, dengan memanfaatkan *Operating System* berbasis *unix* yaitu MikroTik yang sudah dikonfigurasi sedemikian rupa sehingga dapat difungsikan sebagai perangkat *Load Balance*.

**d. Laptop / Komputer**

Laptop disini berfungsi sebagai perangkat yang digunakan untuk melakukan pengaturan konfigurasi pada komputer *server*. Dalam hal ini tidak harus menggunakan laptop, menggunakan komputer biasa juga bisa dan fungsinya juga sama.

**e. Switch HUB**

Switch HUB disini berfungsi untuk memparalel *output* hasil penggabungan *bandwidth* yang telah diproses pada *server* untuk didistribusikan ke beberapa komputer (lebih dari satu).

**f. PC (*Personal Komputer*)**

PC (*Personal Komputer*) disini yaitu seperangkat / unit komputer yang berfungsi sebagai komputer klien. Dari komputer klien inilah nantinya kita melakukan kegiatan akses Internet.

**2. Perancangan Perangkat Lunak (*Software*)**

Dalam membuat Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix diperlukan *software* / program untuk menjalankannya. Untuk dapat membuat program dengan lebih mudah dan cepat pada umumnya diperlukan suatu alur berpikir (*algoritma*) dari sistem yang akan dibuat. Kemudian alur berpikir tersebut dikembangkan menjadi sebuah diagram alir (*flow chart*) yang jelas dan sistematis. Dengan panduan ini kemudian dapat dibuat susunan program *assembly* yang tepat.

Secara garis besar program aplikasi untuk Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini dapat dikelompokkan ke dalam beberapa program, yaitu:

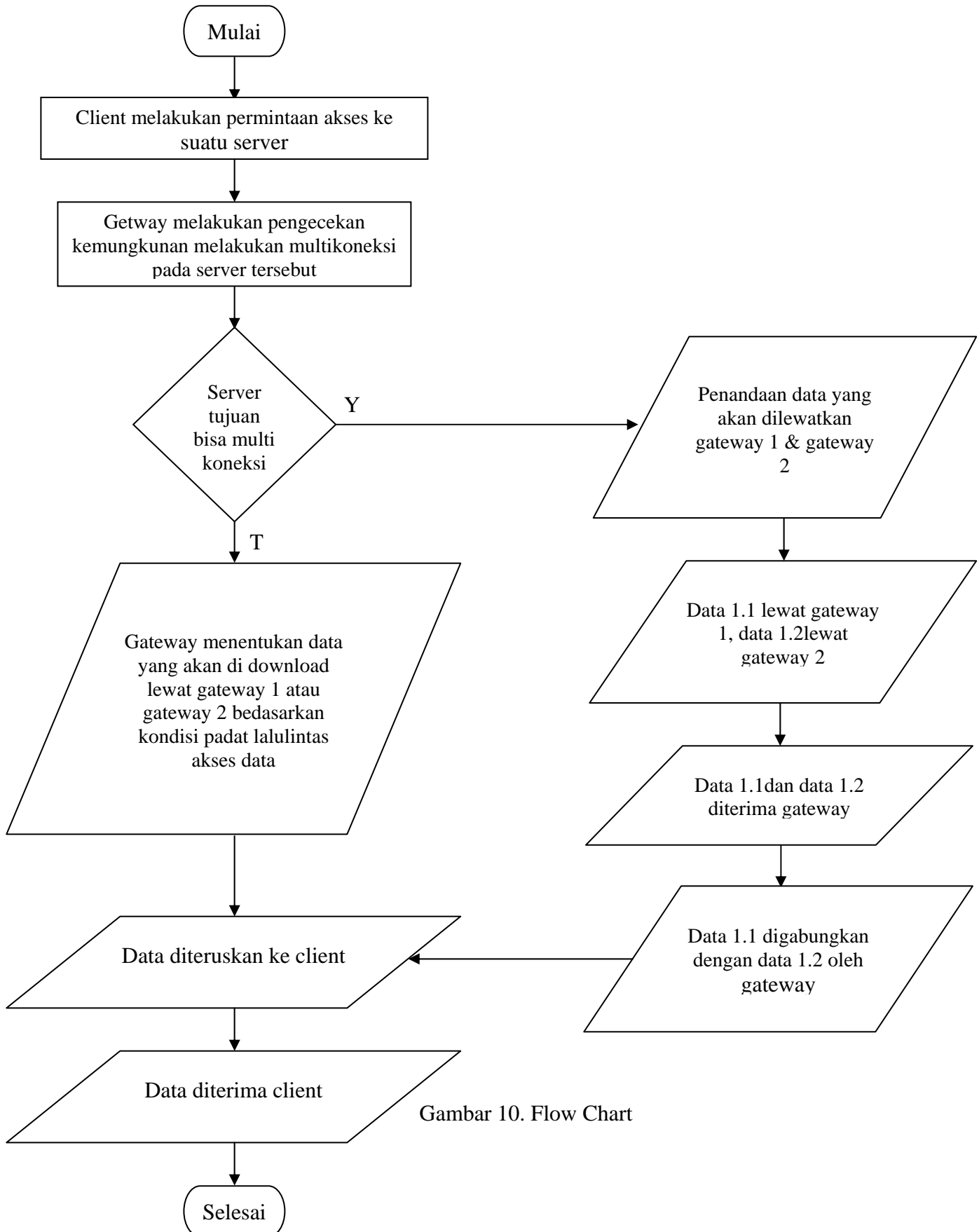
**a. Program Utama**

**1) Algoritma**

- a) Step 1 : Client melakukan permintaan akses ke suatu server yang direquest oleh client
- b) Step 2 : Getway melakukan pengecekan kemungkinan melakukan multikoneksi pada server yang dituju
- c) Step 3 : Apakah server tujuan bisa multi koneksi? Jika YA maka lanjut ke step 4, jika TIDAK maka melalui step 5
- d) Step 4 : Gateway menentukan data yang akan di download lewat gateway 1 atau gateway 2 berdasarkan kondisi padat lalulintas akses data
- e) Step 5 : Penandaan data yang akan dilewatkan melalui gateway 1 & gateway 2
- f) Step 6 : Data 1.1 dilewatkan ke gateway 1, data 1.2 dilewatkan ke gateway 2
- g) Step 7 : Data 1.1 dan data 1.2 diterima gateway
- h) Step 8 : Data 1.1 digabungkan dengan data 1.2 oleh gateway
- i) Step 9 : Data diteruskan dari server LB ke client
- j) Step 10 : Data diterima client dalam bentuk satu paket



## 2) Flow Chart



Gambar 10. Flow Chart

## **E. Pembuatan Alat**

Dalam proses pembuatan Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini terlebih dahulu harus mempersiapkan alat dan bahan yang diperlukan. Berikut adalah alat dan bahan yang digunakan untuk membuat alat tersebut :

1. Satu unit komputer yang difungsikan sebagai komputer server LB.
2. Satu unit komputer yang difungsikan sebagai komputer klien.
3. Satu unit laptop yang difungsikan untuk mengatur / mengkonfigurasi komputer server.
4. Optical drive.
5. Tang krimping.
6. Gunting.
7. Cutter.
8. Bahan dan peralatan pendukung lainnya.

Adapun proses pembuatan alat adalah sebagai berikut:

1. Merakit unit CPU.
2. Installasi jaringan Internet.
3. Merakit unit komputer dengan perangkat pendukung lainnya.
4. Menginstall MikroTik pada komputer server.
5. Melakukan aktivasi MikroTik.
6. Mengatur konfigurasi awal MikroTik.
7. Meremot server menggunakan laptop / komputer lain.

8. Mengkonfigurasi MikroTik dengan pengaturan sebagai Load Balancing.
9. Mendistribusikan jaringan dari server ke komputer klien.

#### a. Langkah Instalasi MikroTik

```

Welcome to MikroTik Router Software installation

Move around menu using 'p' and 'n' or arrow keys, select with 'spacebar'.
Select all with 'a', minimum with 'm'. Press 'i' to install locally or 'r' to
install remote router or 'q' to cancel and reboot.

[X] system          [ ] lcd             [ ] telephony
[ ] ppp             [ ] ntp             [ ] ups
[ ] dhcp           [ ] radiolan        [ ] user-manager
[ ] advanced-tools [ ] routerboard     [ ] web-proxy
[ ] arlan          [ ] routing         [ ] webproxy-test
[ ] gps            [ ] routing-test    [ ] wireless
[ ] hotspot        [ ] rstp-bridge-test [ ] wireless-legacy
[ ] hotspot-fix    [X] security
[ ] isdn           [ ] synchronous

system (depends on nothing):
Main package with basic services and drivers

```

Gambar 11. Menu Utama Instalasi MikroTik

[x] Keterangan beberapa yang penting diantaranya:

- System : Packet wajib install (inti system mikrotik/paket dasar), berisi Kernel Mikrotik.
- PPP : Untuk membuat Point to Point Protocol Server, Point-to-Point tunneling protocols-PPTP, PPPoE and L2TP Access Concentrators and clients; PAP, CHAP, MSCHAPv1 and MSCHAPv2 authentication protocols; RADIUS authentication and accounting; MPPE encryption; compression for PPPoE; data rate limitation; differentiated firewall; PPPoE dial on demand.
- DHCP : Packet yang dibutuhkan apabila ingin membuat DHCP-server (agar client bisa

mendapatkan ip address otomatis -dynamic IP) \* DHCP-DHCP server per interface; DHCP relay; DHCP client; multiple DHCP networks; static and dynamic DHCP leases; RADIUS support.

- Advanced tool : Tools tambahan untuk administrasi jaringan seperti ipscan, bandwidth test, Scanning, Nslookup dan lain lain.
  - Arlan : Packet untuk konfigurasi chipset wireless aironet arlan .
  - Gps : Packet untuk support GPS Device.
  - Hotspot : Packet untuk membuat hotspot gateway, seperti authentication , traffic quota dan SSL HotSpot Gateway with RADIUS authentication and accounting; true Plug-and-Play access for network users; data rate limitation; differentiated firewall; traffic quota; real-time status information; walled-garden; customized HTML login pages; iPass support; SSL secure authentication; advertisement support.
  - Hotspot -fix: Tambahan packet hotspot.
  - Security : Berisi fasilitas yang mengutamakan Keamanan jaringan, seperti Remote Mesin dengan SSH, Remote via MAC Address.
  - Web-proxy : Untuk menjalankan service Web proxy yang akan menyimpan cache agar traffic ke Internet bisa di reduksi sehingga sensasi browsing lebih cepat FTP and HTTP caching proxy server; HTTPS proxy; transparent DNS and HTTP proxying; SOCKS protocol support; DNS static entries; support for caching on a separate drive; access control lists; caching lists; parent proxy support.
  - ISDN : Packet untuk isdn server dan isdn client membutuhkan packet PPP.
  - Lcd : Packet untuk customize port lcd dan lain lain.
- [x] Kita pilih service apa saja yang ingin kita install.

[x] Tekan:

'a' = semua service akan terpilih.

'n' = bila kita menginstall baru.

'y' = bila kita hanya ingin menambah service baru (konfigurasi sebelumnya tidak akan hilang)

[x] Lalu ketik " i " untuk memulai instalasi, maka proses berlanjut... "proses format dan pengkopian file-file yang dibutuhkan akan berjalan otomatis"

(<http://www.unsri.ac.id/upload/arsip/KAJIAN%20PENGGUNAAN%20MIKROTIK%20OS%20SEBAGAI%20ROUTER.pdf>)

Berikut adalah contoh tampilan dari Console MikroTik :

```

MikroTik Login: sohib378
Password:

MMM      MMM      KKK                      TTTTTTTTTT      KKK
MMMM     MMMM     KKK                      TTTTTTTTTT      KKK
MMM MMMM MMM III  KKK KKK RRRRRR   000000   TTT      III  KKK KKK
MMM MM  MMM III  KKKKK  RRR RRR  000 000   TTT      III  KKKKK
MMM     MMM III  KKK KKK RRRRRR   000 000   TTT      III  KKK KKK
MMM     MMM III  KKK KKK RRR RRR   000000   TTT      III  KKK KKK

MikroTik RouterOS 2.9.6 (c) 1999-2005      http://www.mikrotik.com/

Terminal linux detected, using multiline input mode
[sohib378@MikroTik] > _

```

Gambar 12. Tampilan Awal Console MikroTik

```

quit -- Quit console
certificate/ -- Certificate management
redo -- Redo previously undone action
special-login/ -- Special login users
interface/ -- Interface configuration
driver/ -- Driver management
ping -- Send ICMP Echo packets
setup -- Do basic setup of system
password -- Change password
undo -- Undo previous action
port/ -- Serial ports
import --
snmp/ -- SNMP settings
user/ -- User management
file/ -- Local router file storage.
queue/ -- Bandwidth management
system/ -- System information and utilities
ip/ -- IP options
tool/ -- Diagnostics tools
isdn-channels/ -- ISDN channel status info
ppp/ -- Point to Point Protocol
routing/ -- Various routing protocol settings
export --
[sohib378@MikroTik] >

```

Gambar 13. Perintah-perintah konfigurasi pada MikroTik

### ***b. Script Load Balancing***

Berikut adalah beberapa Langkah-langkah yang digunakan penulis untuk konfigurasi MikroTik sebagai server Load Balancing :

```

/ip address

add address=192.168.1.1/24 broadcast=192.168.1.255 comment=""
disabled=no interface=ether1 network=192.168.1.0

add address=192.168.2.1/24 broadcast=192.168.2.255 comment=""
disabled=no interface=ether2 network=192.168.2.0

add address=192.168.3.254/24 broadcast=192.168.3.255 comment=""
disabled=no interface=ether3 network=192.168.3.0

/ip dns

set allow-remote-requests=yes cache-max-ttl=1w cache-size=2048KiB
max-udp-packet-size=512 primary-dns=192.168.1.254 secondary-
dns=192.168.2.254

```

```
/ip firewall connection tracking
set enabled=yes generic-timeout=10m icmp-timeout=10s tcp-close-
timeout=10s tcp-close-wait-timeout=10s tcp-established-timeout=1d
tcp-fin-wait-timeout=10s \
    tcp-last-ack-timeout=10s tcp-syn-received-timeout=5s tcp-syn-
sent-timeout=5s tcp-syncookie=no tcp-time-wait-timeout=10s udp-
stream-timeout=3m udp-timeout=10s
/ip firewall mangle
add action=mark-connection chain=prerouting comment="" disabled=no
dst-address-type=!local in-interface=ether3 new-connection-
mark=local-pcc-01 passthrough=yes \
    per-connection-classifier=src-address-and-port:2/0
add action=mark-connection chain=prerouting comment="" disabled=no
dst-address-type=!local in-interface=ether3 new-connection-
mark=local-pcc-02 passthrough=yes \
    per-connection-classifier=src-address-and-port:2/1
add action=mark-routing chain=prerouting comment="" connection-
mark=local-pcc-01 disabled=no in-interface=ether3 new-routing-
mark=pppoe-01 passthrough=no
add action=mark-routing chain=prerouting comment="" connection-
mark=local-pcc-02 disabled=no in-interface=ether3 new-routing-
mark=pppoe-02 passthrough=no
```

```

/ip firewall nat
add action=masquerade chain=srcnat comment="" disabled=no out-
interface=\
    ether1
add action=masquerade chain=srcnat comment="" disabled=no out-
interface=\
    ether2

/ip firewall service-port
set ftp disabled=no ports=21
set tftp disabled=no ports=69
set irc disabled=no ports=6667
set h323 disabled=no
set sip disabled=no ports=5060,5061
set pptp disabled=no

/ip route
add comment="" disabled=no distance=1 dst-address=0.0.0.0/0
gateway=192.168.1.254 routing-mark=pppoe-01 scope=30 target-scope=10
add comment="" disabled=no distance=1 dst-address=0.0.0.0/0
gateway=192.168.2.254 routing-mark=pppoe-02 scope=30 target-scope=10
add comment="" disabled=no distance=1 dst-address=0.0.0.0/0
gateway=192.168.1.254,192.168.2.254 scope=30 target-scope=10

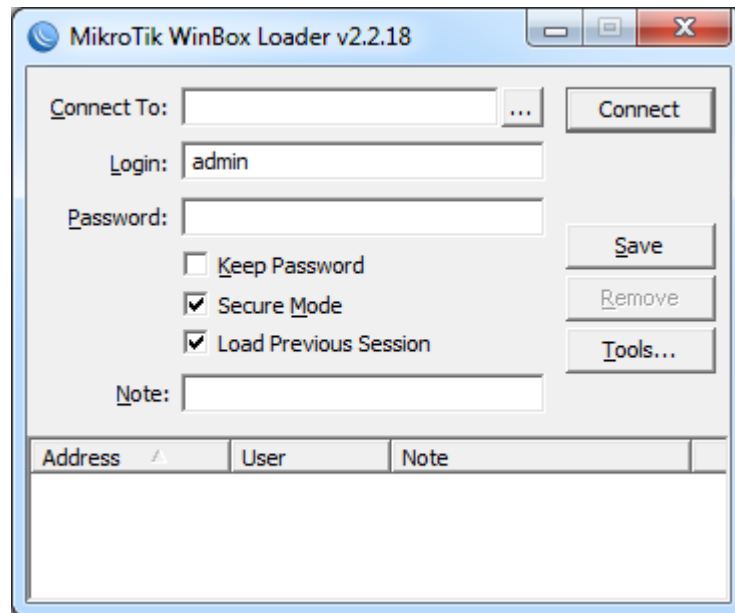
```

### c. Konfigurasi *Load Balancing* dengan WinBox.

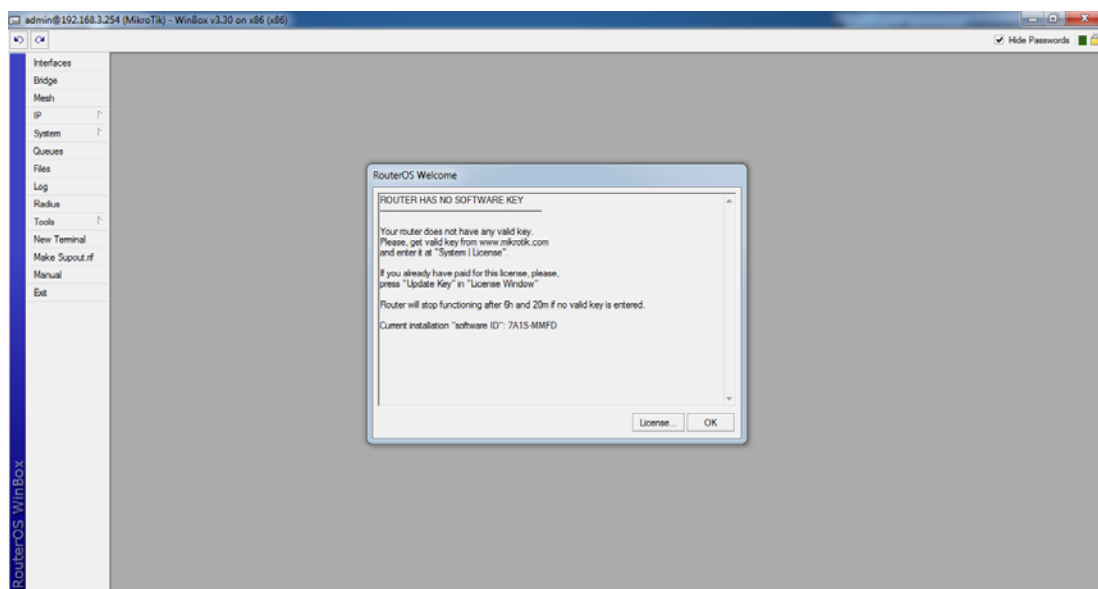
Selain menggunakan perintah di atas yang diketikkan menggunakan console, MikroTik juga menyediakan fasilitas Grapich interface untuk melakukan konfigurasi. Berikut adalah gambaran grapich interface MikroTik atau yang lebih dikenal dengan Winbox :



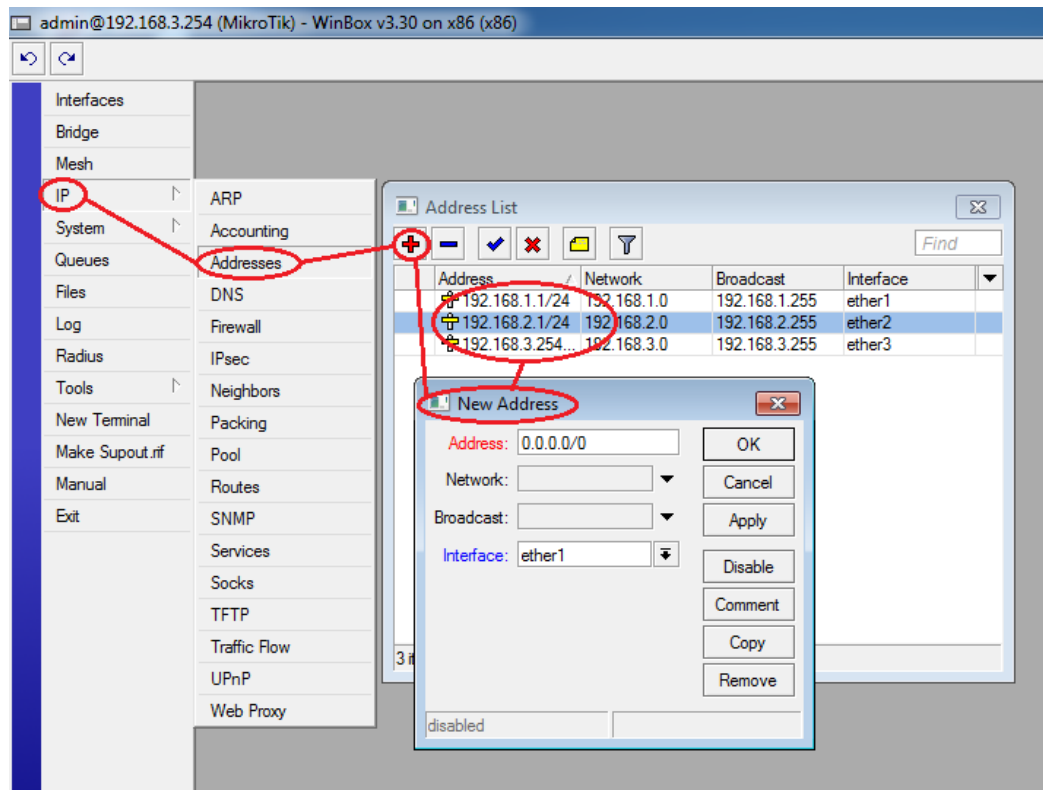
Berikut adalah contoh tampilan grafis interface Winbox :



Gambar 14. Tampilan Awal WinBox



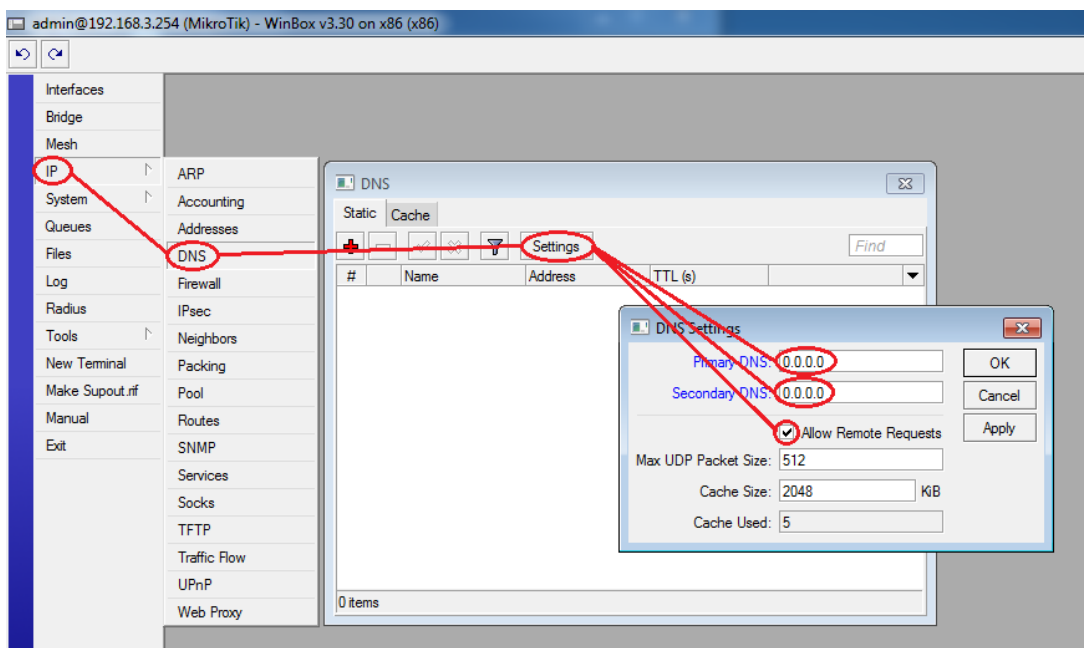
Gambar 15. Tampilan Winbox Setelah Login



Gambar 16. Cara Mendeklarasikan IP pada MikroTik

Gambar diatas adalah langkah pertama yang kita lakukan setelah kita masuk ke MikroTik dengan menggunakan WinBox. Kita deklarasikan terlebih dahulu *IP Address* dari masing-masing *Ethernet* yang kita gunakan dengan cara klik pada *IP* lalu klik *Address*, akan muncul halaman baru dengan nama *Address List*, klik tanda + yang berada pada pojok kiri atas (berwarna merah), maka muncul halaman baru lagi dengan nama *New Address*, pada kolom *Address* isikan *IP Address* dari ISP 1 kemudian pada kolom *Interface* kita pilih *ether1* lalu klik *OK*. Jika sudah maka pada halaman *Address List* akan muncul satu baris keterangan berupa *IP Address* yang telah kita masukkan tadi. Ulangi langkah tersebut untuk memasukkan *IP Address* dari ISP 2 kemudian pada kolom *Interface* kita pilih *ether2* lalu klik *OK*.

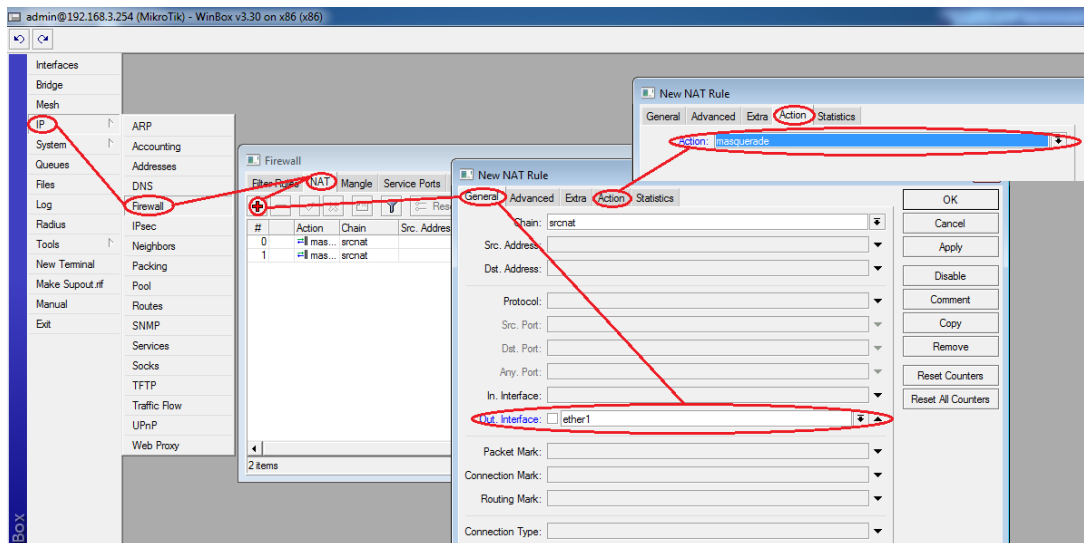
Lakukan sekali lagi langkah seperti sebelumnya, akan tetapi pada langkah yang ke tiga ini sedikit berbeda, klik tanda + yang berada pada pojok kiri atas (berwarna merah), maka muncul halaman baru lagi dengan nama *New Adress*, pada kolom Address isikan *IP Address* yang berfungsi sebagai Gateway komputer *server* kemudian pada kolom Interface kita pilih *ether3* lalu klik *OK*. Melalui Ethernet inilah nantinya *bandwidth* yang telah digabungkan didalam komputer *server* akan didistribusikan ke komputer klien.



Gambar 17. Cara mengisi DNS pada MikroTik

Gambar diatas adalah langkah selanjutnya setelah kita selesai mendeklarasikan *IP Address* yaitu mengatur DNS, Seperti yang terlihat pada gambar di atas, pertama-tama kita klik IP kemudian klik pada DNS maka akan muncul halaman baru dengan nama DNS, klik pada menu Settings dan muncul satu halaman

baru lagi dengan nama DNS Settings. Kemudian isikan pada kolom *Primary DNS* dengan *Gateway* dari ISP 1 sedangkan untuk *Secondary DNS* diisikan dengan *Gateway* dari ISP 2 dan jangan lupa untuk mencentang pada kolom dibawahnya yang bertuliskan *Allow Remote Requests* lalu klik *OK* untuk menyelesaikan.



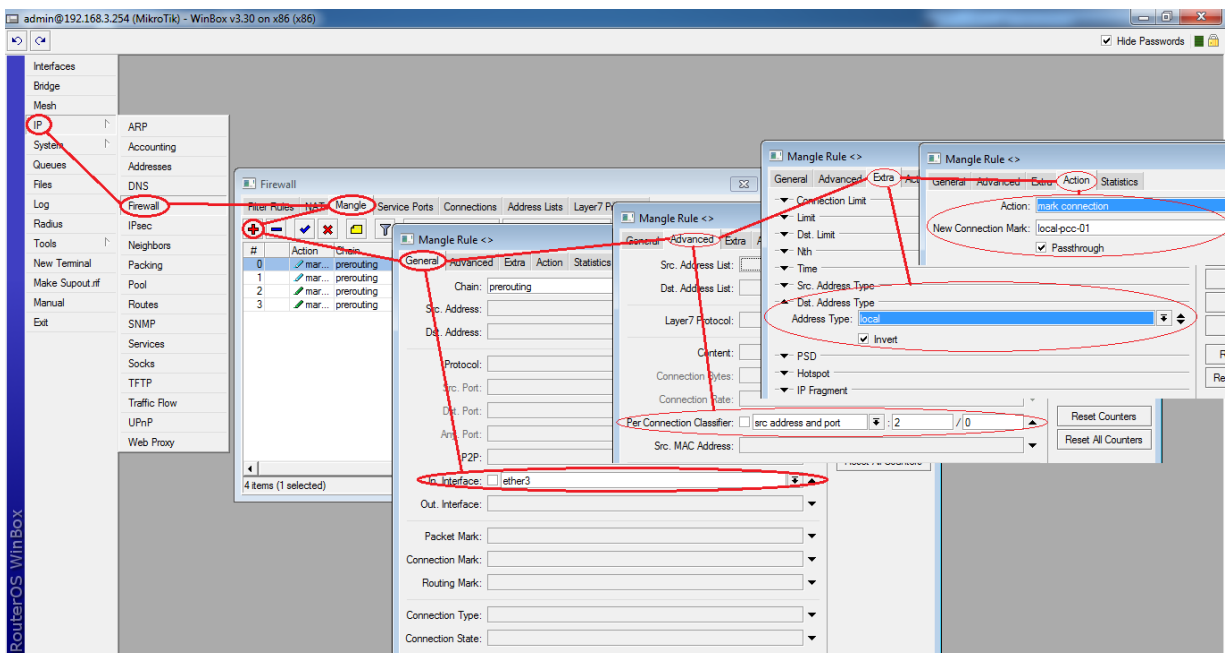
Gambar 18. Cara NAT dari *Firewall* pada MikroTik

Gambar diatas adalah langkah untuk mengisikan NAT dengan menggunakan WinBox. Seperti yang terlihat pada gambar di atas, pertama-tama kita klik terlebih dahulu menu *IP* kemudian *Firewall* lalu klik pada tab *NAT*, klik + untuk menambahkan, maka akan muncul halaman baru dengan nama *New NAT Rule*, pada tab *Genral* yang perlu diganti hanyalah pada kolom *Out. Interface* dan diisi dengan *ether1*, untuk menambahkan *ether1* pada kolom *Out. Interface* caranya adalah dengan mengklik tombol panah bawah yang berada di sisi kanan kolom, lalu pilih *ether1*. Setelah selesai, masih di halaman yang sama, klik pada tab *Action* dan

isi kolom *Action* dengan *masquerade*. Caranya adalah dengan mengklik *drop box* yang terletak pada sisi kanan kolom, kemudian pilih *masquerade*, klik OK untuk menyelesaikan.

Lakukan langkah tersebut sekali lagi, kali ini pada tab *General* yang perlu diganti hanyalah pada kolom *Out. Interface* dan diisi dengan *ether2*, untuk menambahkan *ether2* pada kolom *Out. Interface* caranya adalah dengan mengklik tombol panah bawah yang berada di sisi kanan kolom, lalu pilih *ether2* lalu klik OK untuk menyelesaikan. Jika sudah, maka pada halaman *Firewall* akan tampil dua baris yaitu hasil dari konfigurasi yang telah kita lakukan tadi.

Langkah ini adalah berfungsi untuk membuka / melalukan *traffic* / paket data.



Gambar 19. Cara mengisi NAT dari *Firewall* pada MikroTik

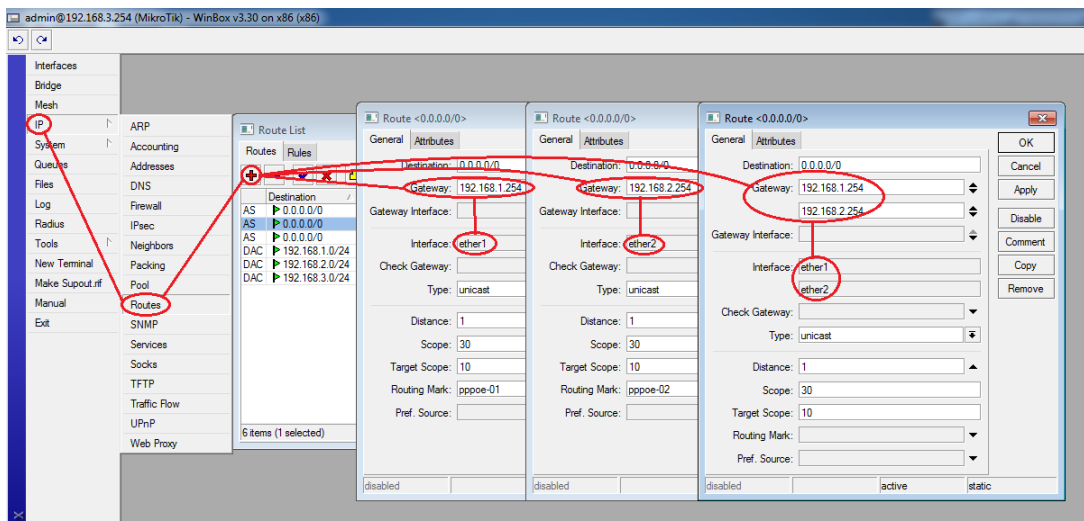
Gambar diatas adalah langkah untuk mengisikan *Mangle* dengan menggunakan WinBox. Seperti yang terlihat pada gambar di atas, pertama-tama kita klik *IP* kemudian *Firewall*, klik pada tab *Mangle* lalu klik **+** maka akan muncul halaman baru dengan nama *Mangle Rule < >*. Pada tab *General* yang perlu diganti hanya pada kolom *In. Interface* dan diisi dengan *ether3*, untuk mengisinya klik tombol panah bawah yang berada di sisi kanan kolom lalu pilih *ether3*. Selanjutnya kita berpindah ke tab *Advanced*, pada tab *Advanced* ini yang perlu diisi hanya pada kolom *Per Connection Classifier* atau yang lebih dikenal dengan *PCC*. Cara untuk mengisinya adalah dengan mengklik tombol arah bawah yang terletak di sisi kanan kolom, kemudian pada kolom yang paling kiri pilih yang bertuliskan *src address and port* lalu kolom selanjutnya ganti isinya dengan **2** sedangkan untuk kolom yang paling kanan biarkan seperti aslinya. Jika sudah kita berpindah ke tab *Extra*, pada tab *Extra* ini yang perlu diisi hanya pada bagian *Dst. Address Type*, cara mengisinya adalah klik tombol arah bawah yang terletak di sebelah kiri dari tulisan *Dst. Address Type* maka dibawahnya akan muncul kolom *Address Type*. Klik pada tombol arah bawah yang terletak di sebelah kanan dari kolom *Address Type* kemudian pilih *local* dan jangan lupa untuk memberi tanda centang pada kolom *Invert* di bawahnya. Jika sudah, maka tab yang selanjutnya yaitu *Action* pilih menu *mark connection* pada *drop box*, maka secara otomatis akan muncul satu kolom baru dibawahnya yaitu *New Connection Mark*, pada kolom ini kita pilih *local-pcc-01* yang terletak di dalam *drop box*, jika sudah maka klik OK untuk menyelesaikan.

Klik **+** sekali lagi, lakukan langkah seperti diatas. Pada langkah ke dua ini ada sedikit perbedaan yaitu pada tab *Advanced* pada kolom *Per Connection Classifier* diisi

dengan *src address and port* : 2 / 1 kemudian tab *Action*, pada kolom *New Connection Mark* diisi dengan *local-pcc-02* lalu klik *OK* untuk menyelesaikan.

Klik + sekali lagi, lakukan langkah seperti diatas. Pada langkah ke tiga ini untuk tab *General* tambahkan *local-pcc-01* pada kolom *Connection Mark*, untuk tab *Advanced* dan tab *Extra* tidak perlu diisi, selebihnya diisi seperti langkah sebelumnya yaitu langkah pertama. Klik *OK* untuk menyelesaikan.

Klik + sekali lagi, lakukan langkah seperti diatas. Pada langkah ke tiga ini untuk tab *General* tambahkan *local-pcc-02* pada kolom *Connection Mark*, untuk tab *Advanced* dan tab *Extra* tidak perlu diisi, selebihnya diisi seperti langkah sebelumnya yaitu langkah ke dua. Klik *OK* untuk menyelesaikan.



Gambar 20. Cara mendefinisikan *Routes* pada MikroTik

Gambar diatas adalah langkah untuk mendefinisikan *Routes* dengan menggunakan WinBox. Seperti yang terlihat pada gambar di atas, pertama-tama kita

klik *IP* kemudian *Routes* maka akan muncul halaman baru yaitu *Route List*, kemudian klik **+** maka muncul halaman baru lagi yaitu *Route <0.0.0.0/0>* yang perlu diisi hanyalah pada kolom *Gateway* dan diisi dengan *Gateway* dari *ISP 1*, setelah diisi maka secara otomatis pada kolom *Interface* muncul tulisan *ether1*. Klik *OK* untuk menyelesaikan. Ulangi kembali langkah tersebut, dan untuk langkah ke dua ini pada *Gateway* diisi dengan *Gateway* dari *ISP 2*, setelah diisi maka secara otomatis pada kolom *Interface* muncul tulisan *ether2*. dan selesaikan dengan mengklik *OK*. Lakukan sekali lagi langkah di atas, kali ini pada kolom *Gateway* yang pertama diisi dengan *Gateway* dari *ISP 1* dan pada kolom *Gateway* yang berada di bawahnya diisi dengan *Gateway* dari *ISP 2*, setelah diisi maka secara otomatis pada kolom *Interface* muncul tulisan *ether1* dan di bawahnya muncul tulisan *ether2*. Jika sudah, klik *OK* untuk menyelesaikan.

Setelah langkah-langkah di atas selesai dilakukan, maka komputer *server* telah siap untuk digunakan.



## F. Spesifikasi Alat

Spesifikasi alat dari Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini adalah sebagai berikut:

1. Tegangan *Supply* CPU AC :  $220 V_{AC}$  50 Hz/60 Hz
2. Tegangan *Supply* adaptor MODEM ADSL AC :  $220 V_{AC}$  50 Hz/60 Hz
3. Tegangan *Supply* MODEM ADSL DC :  $9 V_{DC}$
4. Unit komputer (minimal Pentium 3)
5. MODEM ADSL
6. Fungsi : Menggabungkan 2 (dua) *bandwidth downstream Internet*

## G. Pengoperasian Alat

Pengoperasian alat ini dapat dilakukan dengan cara sebagai berikut :

1. Memasang kabel LAN dari ISP (*Internet Service Provider*) ke komputer server.
2. Memasang kabel LAN dari komputer server ke komputer klien.
3. Menghubungkan kabel konektor AC ke sumber tegangan  $220 V_{AC}$  50 Hz/ 60 Hz.
4. Menghidupkan MODEM ADSL dengan cara menekan tombol *power* pada posisi *ON*, pada beberapa jenis MODEM ADSL tidak disediakan tombol *power*.
5. Menghidupkan CPU / komputer server dengan cara menekan tombol *power*.

6. Untuk menghentikan / mematikan MODEM ADSL, tekan saklar *power* sehingga berubah ke posisi *off*, sedangkan jika tidak tersedia saklar *power*, maka dapat langsung mencabut adaptor.
7. Untuk menghentikan / mematikan komputer server, kita ketikkan perintah berikut :

```
[admin@MikroTik] > system shutdown
```

```
Shutdown, yes? [y/N]: y. (lalu enter)
```

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **A. Hasil Pengujian**

Pengujian alat dilakukan untuk mengetahui kinerja dari Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix, baik kinerja tiap bagian maupun keseluruhan, dalam pengujian ini meliputi :

##### **1. Pengujian Fungsi Per bagian**

###### **a. Pengujian Unit MODEM ADSL**

Pengujian ini berfungsi untuk mengetahui apakah MODEM ADSL ini sudah bekerja dengan baik atau belum. Pengujian ini dilakukan dengan cara melihat indikator yang terdapat pada MODEM ADSL, jika menyala semua dengan warna hijau serta berkedip pada *Ethernet* maka dapat dipastikan bahwa modem dalam keadaan normal.

Selain dari indikator tersebut, dapat pula dilihat melalui *traffic* data pada modem dengan cara mengakses IP modem (*default* : 192.168.1.1) lalu masukkan *username password* dan pada halaman awal tersebut biasanya tertera *traffic* data yang sedang berjalan.

###### **b. Pengujian Unit komputer**

Pengujian ini berfungsi untuk mengetahui apakah unit komputer sudah bekerja dengan baik atau belum. Pengujian ini dilakukan dengan cara melihat layar monitor yang terhubung dengan komputer *server*. Muncul tulisan MikroTik Login : cobalah *login* dengan

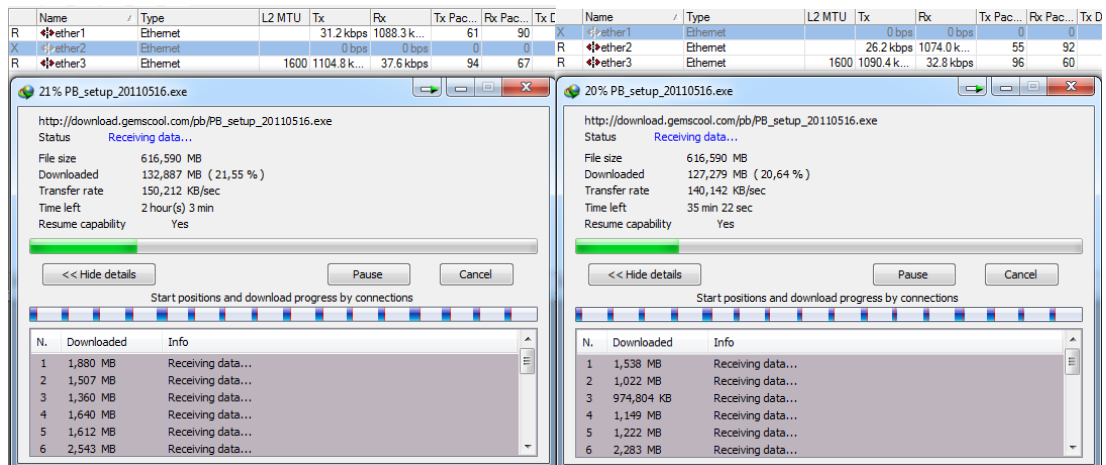
menggunakan *username* dan *password* yang telah ditentukan. Jika berhasil, maka akan muncul tulisan MikroTik di tengah layar monitor. Hal ini menandakan bahwa unit komputer server dalam keadaan normal.

### c. Pengujian Jaringan Internet

Pengujian ini berfungsi untuk mengetahui apakah ada salah satu jaringan Internet yang tidak bekerja. Pengujian ini dilakukan dengan cara mencoba satu per satu jaringan Internet tersebut pada sebuah komputer / laptop, jika masing-masing jaringan dapat digunakan untuk mengakses Internet maka dapat dipastikan bahwa jaringan tersebut dalam kondisi bagus.

## 2. Pengujian Kinerja Alat Keseluruhan

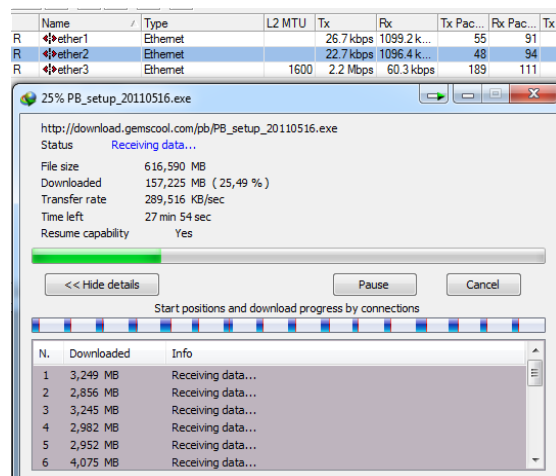
Pengujian ini bertujuan untuk mengetahui apakah Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix telah bekerja sesuai dengan yang diharapkan atau belum. Pengujian dilakukan dengan cara menghubungkan semua kabel LAN yaitu dari jaringan Internet menuju ke komputer *server*, output komputer *server* didistribusikan ke masing-masing komputer klien. Dengan asumsi masing-masing jaringan memiliki kapasitas *bandwidth* 512 Kb/s, maka kita coba *download* suatu file dan kita melakukan pengujian koneksi Internet kita. Setelah kita lakukan pengujian dan hasil menunjukkan angka kisaran 1 Mb/s atau mendekati 2 (dua) kali *bandwidth* semula, maka proses penggabungan *bandwidth* pada komputer *server* telah selesai dan berhasil. Berikut adalah bukti hasil penggabungan *bandwidth downstream* menggunakan metode PCC :



Gambar 21. Pengujian Jaringan Speedy pada Ether1 dan Ether2

Pada gambar diatas menunjukkan hasil pengujian menggunakan jaringan Speedy dengan *bandwidth downstream* 1 Mb/s, pada pengujian ini untuk ether1 dan ether2 dimatikan atau dalam keadaan *disable* secara bergantian dengan tujuan untuk mengetahui kecepatan *download* ketika hanya menggunakan 1 (satu) jaringan Speedy dengan *bandwidth* 1 Mb/s.

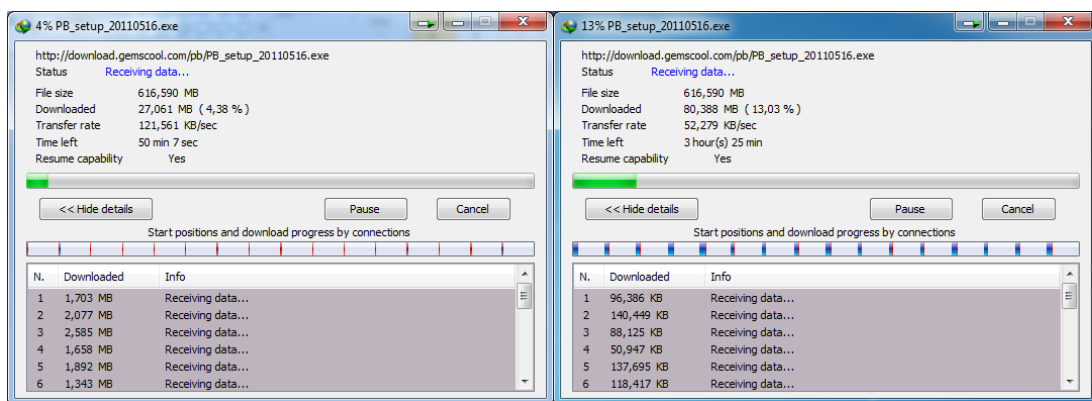
Berikut adalah hasil dari penggabungan *bandwidth downstream* dari 2 (dua) jaringan Speedy dengan *bandwidth* masing-masing 1 Mb/s :



Gambar 22. Hasil Penggabungan 2 Jaringan Speedy

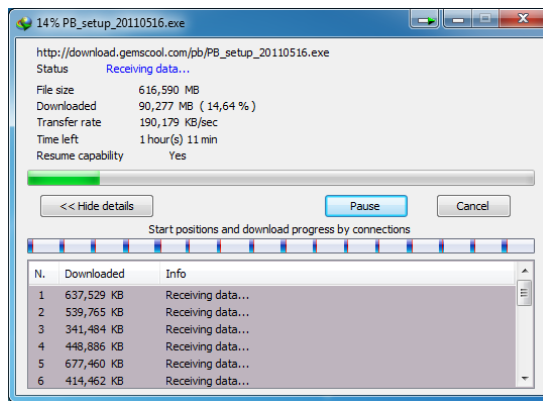
Gambar diatas menunjukkan hasil pengujian menggunakan jaringan *Speedy* dengan *bandwidth downstream* masing-masing 1 Mb/s, pada pengujian tersebut keduanya (*ether1* dan *ether2*) dalam keadaan aktif dan didapatkan hasil bahwa kecepatan *download* bertambah setelah kedua jaringan *Speedy* tersebut digabungkan dengan metode PCC.

Pengujian juga dilakukan dengan memadukan antara jaringan *Speedy bandwidth* 1 Mb/s dan GSM yaitu TELKOMSEL FLASH *bandwidth* 384 Kb/s pada jaringan HSDPA, berikut adalah hasil pengujiannya :



Gambar 23. Pengujian Jaringan *Speedy* 1 Mb/s dan TELKOMSEL FLASH 384 Kb/s


Gambar diatas menunjukkan hasil pengujian menggunakan jaringan *Speedy* dengan *bandwidth downstream* 1 Mb/s dan TELKOMSEL FLASH dengan *bandwidth downstream* 384 Kb/s, pada pengujian ini untuk *ether1* dan *ether2* dimatikan atau dalam keadaan *disable* secara bergantian dengan tujuan untuk mengetahui kecepatan *download* ketika hanya menggunakan 1 (satu) jaringan *Speedy* dengan *bandwidth* 1 Mb/s dan ketika menggunakan jaringan TELKOMSEL FLASH dengan *bandwidth* 384 Kb/s.

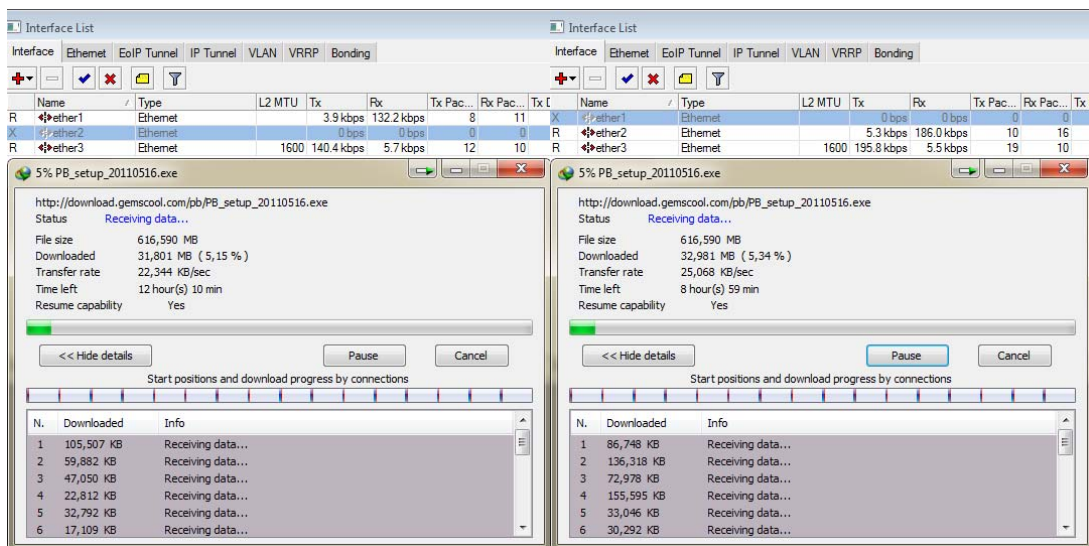


Gambar 24. Pengujian Hasil Penggabungan *Speedy* dengan TELKOMSEL FLASH

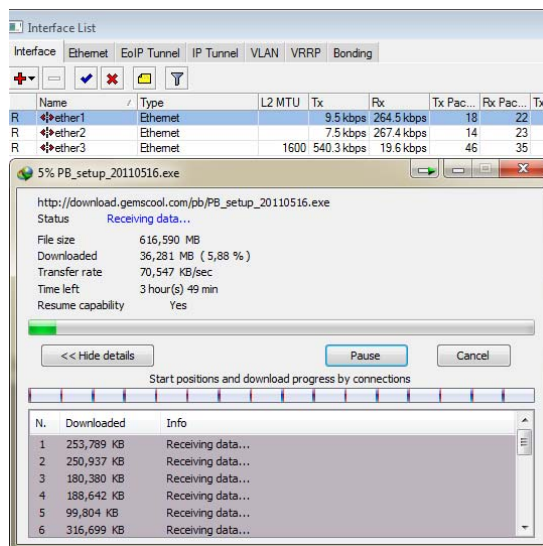
Pada gambar diatas adalah hasil pengujian menggunakan *Speedy* dengan *bandwidth downstream* 1 Mb/s yang digabungkan dengan TELKOMSEL FLASH dengan *bandwidth downstream* 384 Kb/s jaringan HSDPA, pada pengujian tersebut keduanya (*ether1* dan *ether2*) dalam keadaan aktif dan didapatkan hasil bahwa kecepatan *download* bertambah setelah kedua jaringan tersebut digabungkan dengan metode PCC.

Selain dipergunakan untuk jaringan dari *provider* seperti *Speedy*, Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini juga dapat digunakan untuk menggabungkan bandwidth dari GSM-GSM, CDMA-CDMA, dan juga GSM-CDMA dan berikut adalah bukti untuk hasil dari penggabungan bandwidth baik GSM, CDMA dan gabungan dari keduanya :

Hasil pengujian untuk penggabungan 2 jaringan GSM dari *provider* 





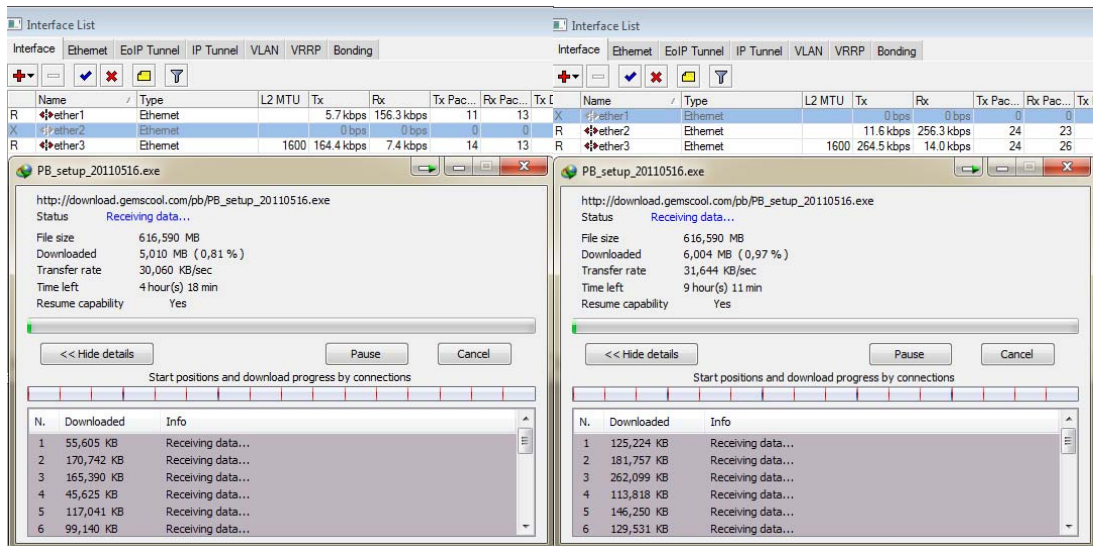
Gambar 25. Pengujian Jaringan GSM 3 pada *ether1* dan *ether2*



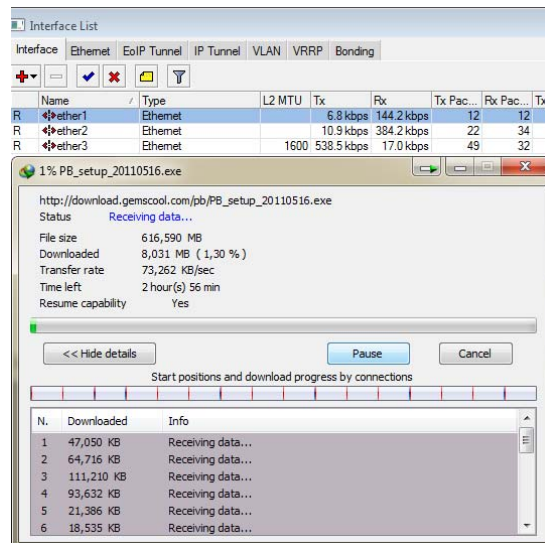
Gambar 26. Hasil Penggabungan Jaringan GSM 3 pada *ether1* dan *ether2*



Hasil pengujian untuk penggabungan 2 jaringan GSM dari provider  + 



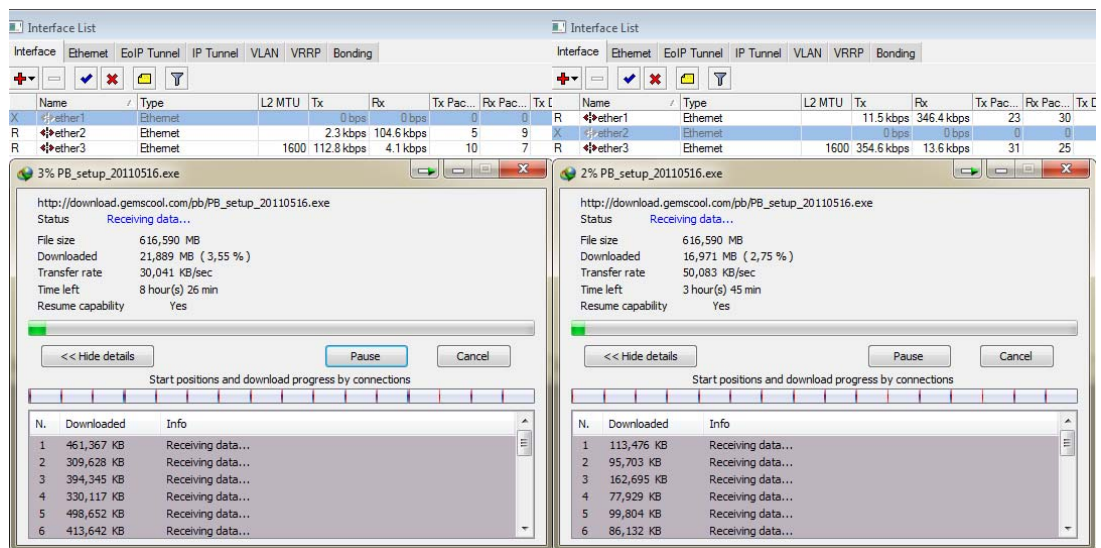
Gambar 27. Pengujian Jaringan GSM 3 dan FLASH pada *ether1* dan *ether2*



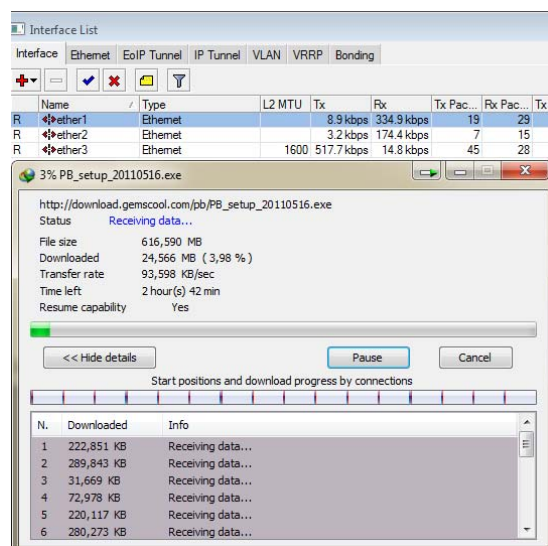
Gambar 28. Hasil Penggabungan Jaringan GSM 3 dan FLASH pada *ether1* dan *ether2*

Hasil pengujian untuk penggabungan 2 jaringan GSM dari *provider 3* dengan

jaringan CDMA dari *provider smartfren*




Gambar 29. Pengujian Jaringan GSM 3 dan Jaringan CDMA *Smartfren* pada *ether1* dan *ether2*

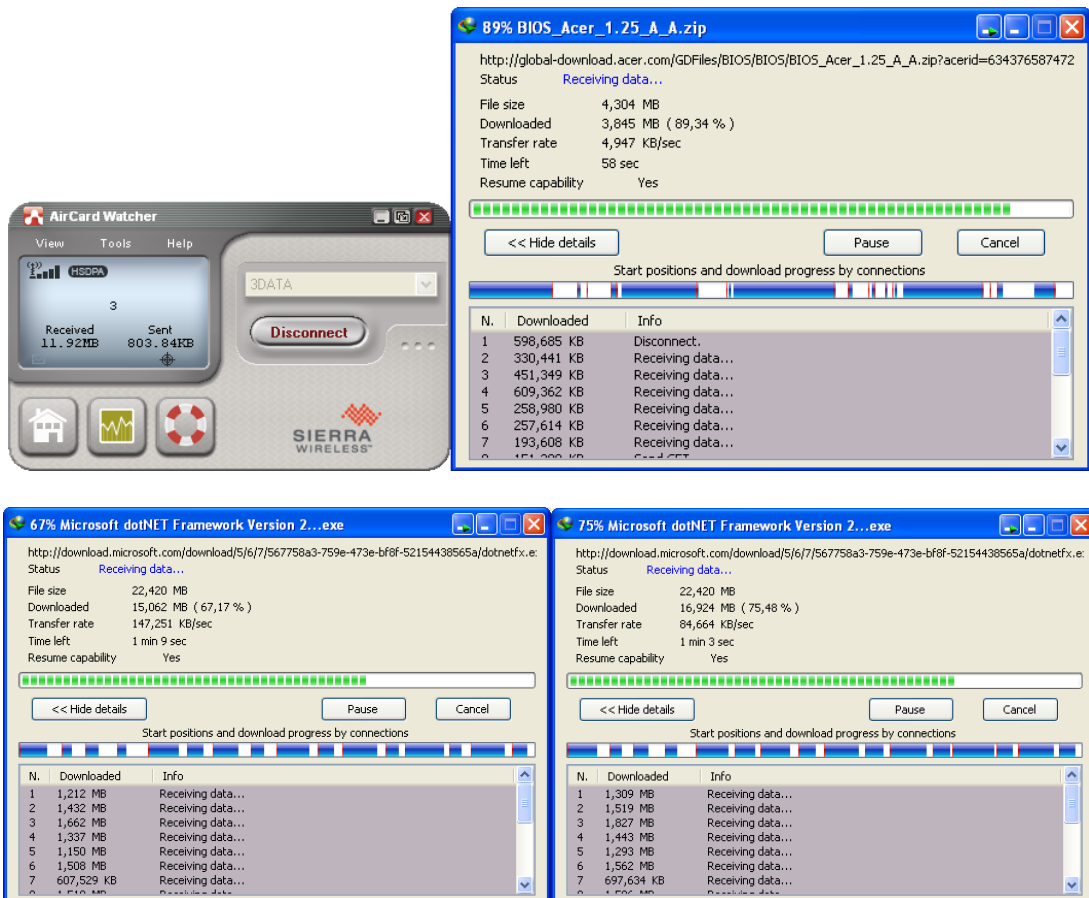


Gambar 30. Hasil Penggabungan Jaringan GSM 3 dan Jaringan CDMA *Smartfren* pada *ether1* dan *ether2*

Untuk GSM sendiri dikarenakan memanfaatkan jaringan yang didistribusikan menggunakan *wireless*, maka kecepatannya pun berubah ubah secara drastis, hal ini dipengaruhi oleh banyak faktor antara lain yaitu cuaca, pepohonan, gedung, lokasi dan beberapa faktor lainnya.

Berikut adalah bukti dari hasil pengujian GSM yang penulis lakukan di beberapa tempat dan dengan waktu yang berbeda.

Hasil pengujian menggunakan GSM dengan provider  pada jaringan HSDPA

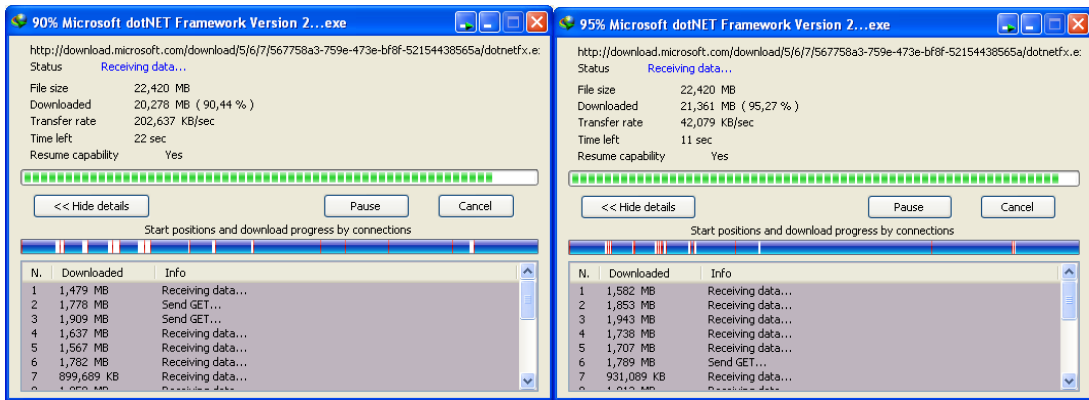


The image displays three screenshots related to mobile data usage and downloads. On the left is the 'Air Card Watcher' interface, which shows a signal strength indicator, 'HSDPA' network type, and a '3' icon. It also displays 'Received 11.92MB' and 'Sent 803.94KB' data usage, along with a 'Disconnect' button and the 'SIERRA WIRELESS' logo. On the right are two download progress windows. The top window is titled '89% BIOS\_Acer\_1.25\_A.A.zip' and shows a download progress of 3,845 MB (89.34%) at a rate of 4,947 KB/sec. The bottom window shows two instances of '67% Microsoft dotNET Framework Version 2...exe' downloads, both at 147,251 KB/sec. Each window includes a progress bar, 'Pause' and 'Cancel' buttons, and a table of download connections.


N.	Downloaded	Info
1	598,685 KB	Disconnect...
2	330,441 KB	Receiving data...
3	451,349 KB	Receiving data...
4	609,362 KB	Receiving data...
5	258,980 KB	Receiving data...
6	257,614 KB	Receiving data...
7	193,608 KB	Receiving data...

N.	Downloaded	Info
1	1,212 MB	Receiving data...
2	1,432 MB	Receiving data...
3	1,662 MB	Receiving data...
4	1,337 MB	Receiving data...
5	1,150 MB	Receiving data...
6	1,508 MB	Receiving data...
7	607,529 KB	Receiving data...

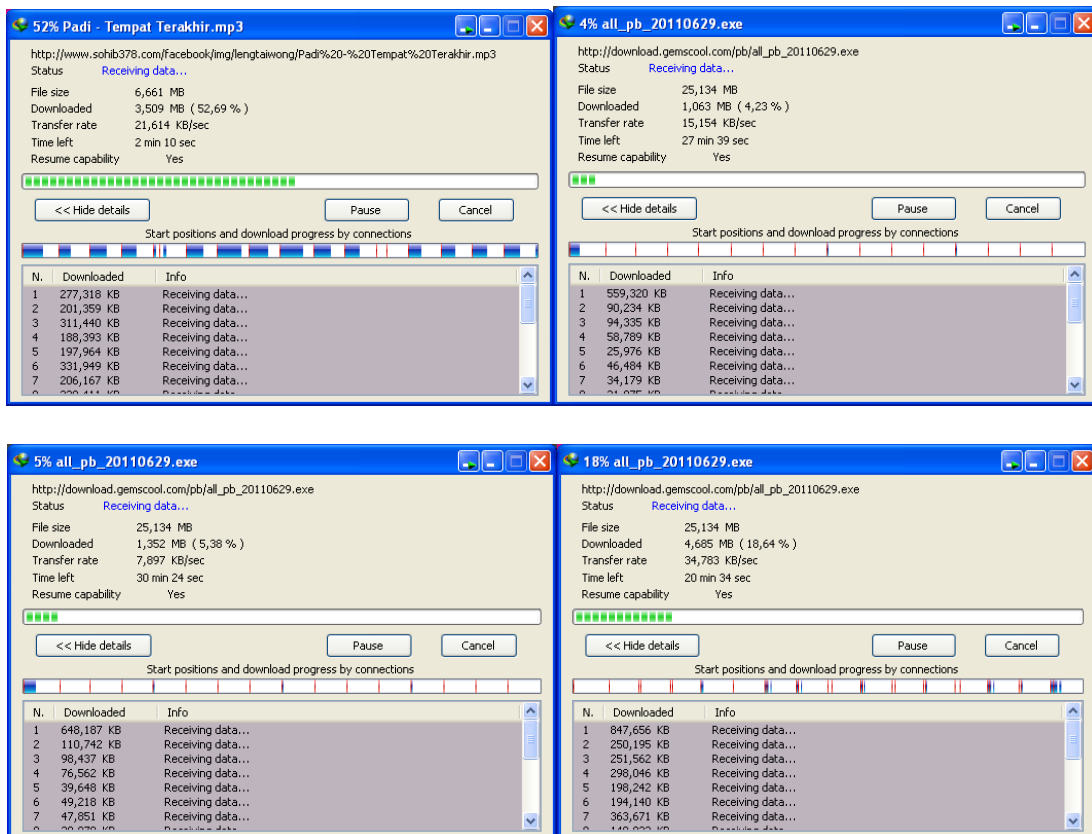
N.	Downloaded	Info
1	1,309 MB	Receiving data...
2	1,519 MB	Receiving data...
3	1,827 MB	Receiving data...
4	1,443 MB	Receiving data...
5	1,293 MB	Receiving data...
6	1,562 MB	Receiving data...
7	697,634 KB	Receiving data...




Gambar 31. Pengujian Jaringan GSM 3

Hasil pengujian menggunakan CDMA dengan provider  pada jaringan EVDO

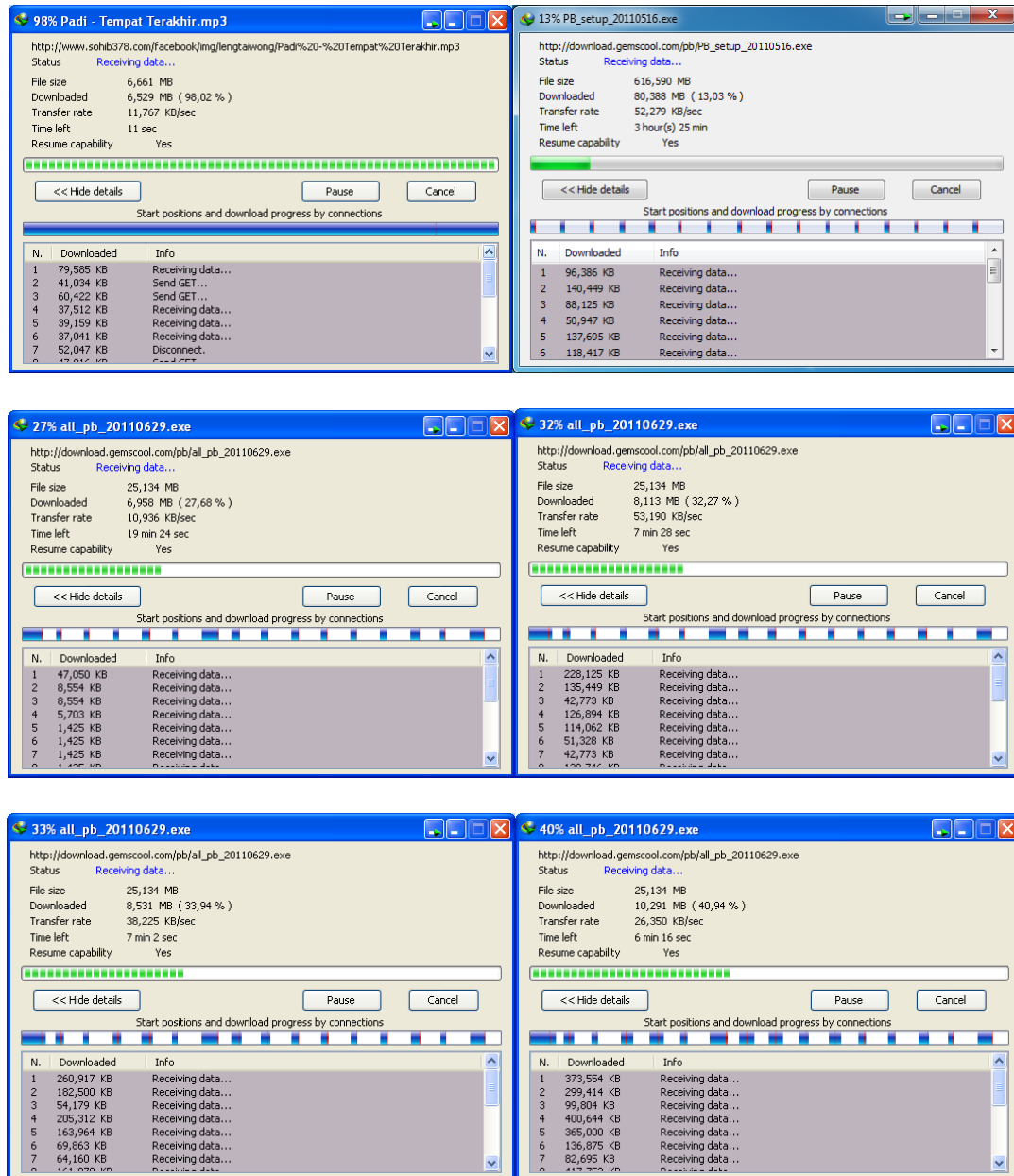
Rev. A



Gambar 32. Pengujian Jaringan CDMA Smartfren

Hasil pengujian menggunakan GSM dengan provider  pada jaringan

HSDPA



Gambar 33. Pengujian Jaringan GSM TELKOMSEL FLASH

## **B. Pembahasan**

### **1. MODEM ADSL**

Berdasarkan hasil pengujian unit MODEM ADSL ini telah bekerja dengan baik, setelah *supply* tegangan dipasangkan dan saklar *power* dinyalakan maka dalam beberapa menit modem telah siap digunakan. Hal ini ditunjukkan dengan berkedipnya lampu *ethernet* pada modem dengan nyala lampu warna hijau serta adanya *traffic* data pada masing-masing MODEM yang dapat dilihat melalui *firmware* yang sudah terpasang dari pabrikan.

### **2. Komputer Server**

Berdasarkan hasil pengujian pada komputer *server* menunjukkan bahwa unit ini telah bekerja dengan baik dimana setelah *supply* tegangan dipasangkan dan saklar *power* dinyalakan, maka pada layar monitor akan muncul tulisan MikroTik Login : cobalah *login* dengan menggunakan *username* dan *password* yang telah ditentukan. Jika berhasil, maka akan muncul tulisan MikroTik di tengah layar monitor.

### **3. Kinerja Alat**

Kinerja dari Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix dapat kita ketahui setelah kita menginstallasi keseluruhan jaringan / kabel LAN mulai dari ISP kemudian didistribusikan ke komputer *Server*, dimana didalam komputer *Server* tersebut terjadi proses penggabungan kedua *bandwidth* Internet dari kedua jaringan yang didistribusikan oleh ISP. Untuk mengetahui hasilnya, maka *output* dari komputer *Server* didistribusikan ke komputer klien

menggunakan perantara kabel LAN. Kabel LAN inilah yang menghantarkan *bandwidth* yang telah digabungkan didalam komputer *Server* sehingga menjadi 1 (satu) *bandwidth* yang utuh. Untuk membuktikannya dapat dilakukan dengan cara men-*download* suatu file dan kita melakukan pengujian koneksi Internet kita. Setelah kita lakukan pengujian dan hasil menunjukkan angka kisaran 2 (dua) kali lipat dari *bandwidth* semula seperti yang telah ditunjukkan pada gambar sebelumnya, maka proses penggabungan *bandwidth* pada komputer *server* telah selesai dan berhasil.

## BAB V

### KESIMPULAN DAN SARAN

#### A. Kesimpulan

Dari uraian perancangan, pembuatan dan pembahasan yang telah dijabarkan di atas, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini dibangun dari penggabungan perangkat keras (*hardware*) dan perangkat lunak (*software*). Untuk *hardware* sendiri terdiri dari MODEM ADSL, komputer *server*, komputer klien atau bisa juga menggunakan laptop.
2. Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini memanfaatkan operating *system* yang masih satu keluarga dengan Linux, hanya saja pada MikroTik itu sendiri dibuat untuk menyederhanakan perintah *routing* yang terdapat di Linux, dan program yang mengendalikan program tersebut adalah API (*Application Programming Interface*) untuk lebih jelasnya dapat dilihat di *website* resmi MikroTik : <http://wiki.mikrotik.com/wiki/Manual:API>
3. Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini secara keseluruhan sudah sesuai dengan fungsi yang ditetapkan. Alat ini dapat menggabungkan *bandwidth* Internet menyesuaikan kemampuan *slot* PCI pada *motherboard*, dalam artian semakin banyak *slot* PCI maka semakin banyak pula jaringan Internet yang dapat digabungkan oleh alat ini.



## B. Keterbatasan Alat

Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini masih mempunyai beberapa keterbatasan, yaitu:

1. Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini tidak dapat menggabungkan *bandwidth* Internet dengan banyak jaringan karena keterbatasan *slot* PCI pada *motherboard*.
2. Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini rentan terhadap kerusakan yang biasanya diakibatkan karena temperatur komputer *server* cukup tinggi.
3. Identifikasi lisensi MikroTik adalah melalui *harddisk*, dimana jika hardisk kita rusak, maka kita tidak dapat menggunakan lisensi yang sama pada *harddisk* pengganti / unit lain.

## C. Saran

Beberapa hal yang dapat dikaji dan dikembangkan dari Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix ini, yaitu :

Penggunaan komputer yang dimanfaatkan untuk Teknologi Penggabungan Bandwidth Downstream Menggunakan Operating System Berbasis Unix kurang fleksibel sebab selain harus merakit terlebih dahulu unit komputer tersebut, kita juga masih harus melakukan penginstallan MikroTik sendiri, maka dari itu agar lebih efektif dan efisien disarankan untuk mengganti

unit komputer dengan *Router Borrad (RB)* sebab RB memiliki *slot* LAN yang cukup banyak sehingga dapat dimanfaatkan untuk menggabungkan banyak jaringan dan juga RB lebih tahan panas bila dibandingkan komputer. Untuk segi harga pun RB sudah cukup murah untuk saat ini.

## DAFTAR PUSTAKA

- Anonim. (2010). Wikipedia Indonesia. Didownload dari:  
([http://id.wikipedia.org/wiki/Sejarah\\_komputer](http://id.wikipedia.org/wiki/Sejarah_komputer)), 28 Juli 2010
- Anonim. (2010). Wikipedia Indonesia. Didownload dari:  
(<http://id.wikipedia.org/wiki/Komputer>), 28 Juli 2010
- Anonim. (2010). Wikipedia Indonesia. Didownload dari:  
(<http://id.wikipedia.org/wiki/ADSL>), 30 Juli 2010
- Anonim. (2010). Manual : PCC Didownload dari:  
(<http://wiki.mikrotik.com/wiki/Manual:PCC>), 17 September 2010
- Anonim. (2010). Manual : PCC Didownload dari:  
(<http://wiki.mikrotik.com/wiki/Manual:PCC>), 17 September 2010
- Dwi Febrian Handriyanto. (2009). KAJIAN PENGGUNAAN MIKROTIK ROUTER OS™ SEBAGAI ROUTER PADA JARINGAN KOMPUTER. Didownload dari: (<http://www.unsri.ac.id/upload/arsip/KAJIAN%20PENGGUNAAN%20MIKROTIK%20OS%20SEBAGAI%20ROUTER.pdf>), 13 Mei 2010
- Eka Bangun Setiawan, Syaifudin Ahmad. (tth). LAN CARD. Didownload dari: (<http://blog.uad.ac.id/kartikaf/files/2009/11/A09-8-LAN-Card.pdf>), 2 Agustus 2010
- Pujo Dewobroto. (tth). Load Balance menggunakan Metode PCC. Didownload dari: ([http://www.mikrotik.co.id/artikel\\_lihat.php?id=34](http://www.mikrotik.co.id/artikel_lihat.php?id=34)), 22 September 2010

# LAMPIRAN



Six-in-one Wireless and Wired VPN Solution

**VPN, Wireless Access Point, Firewall, Router, Switch and ADSL2+ modem**

This comprehensive and affordable solution offers wired, wireless connectivity with business-class protection for small office and remote/branch office users. It combines six functions in a single, compact package – Stateful Packet Inspection (SPI) firewall, 802.11g wireless access point, IPSec Virtual Private Network (VPN), NAT router, an eight port 10/100 fast ethernet switch and built-in ADSL2+ modem support.

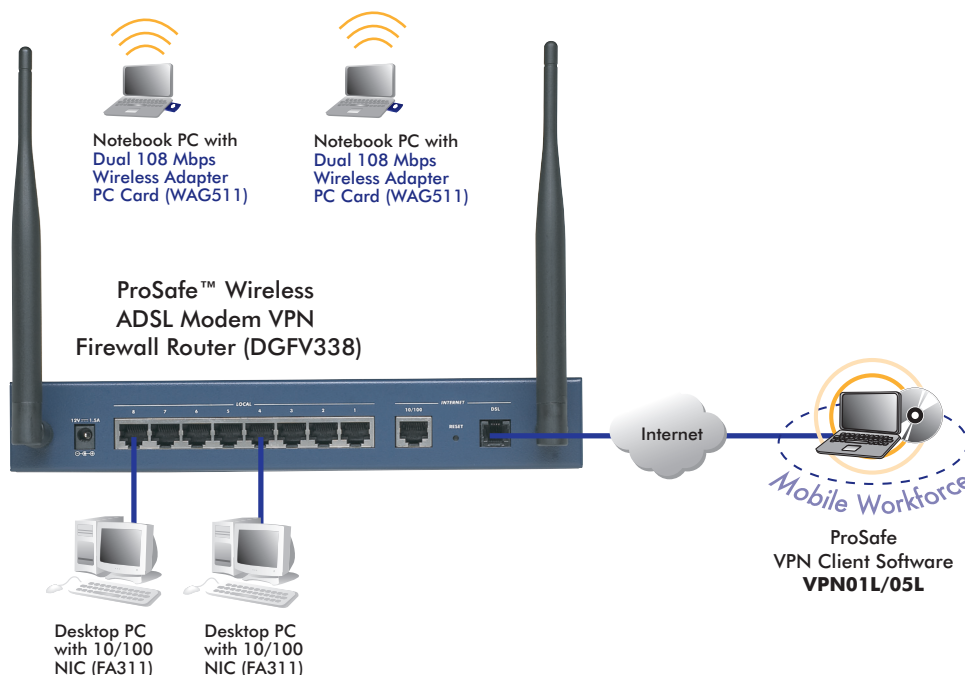
Easy and reliable, the DGFV338 delivers both 10 and 100 Mbps connections for wired devices, plus wireless support at 2.4 Ghz frequency at 54 Mbps radio transmission with up to 108 Mbps in 802.11g (region specific).

More than just a simple NAT router, the ProSafe DGFV338 SPI firewall provides business-class protection, blocking unwanted users from accessing the network. Support for Wi-Fi Protected Access 2 Enterprise (WPA2) shields wireless communication with the highest available level of industry-standard encryption and authentication. Comprehensive controls block or filter unwanted addresses, services, protocols, and URLs, fortifying your network. Up to fifty simultaneous IPSec VPN connections protect links between business locations, encrypting all traffic as it traverses the Internet.

**Ideal Solution for Telecommuters and Remote Offices**

The DGFV338 is easy to set up. It delivers all the advantages of centralized administration for remote offices, teleworkers and small businesses, including dynamic DNS capabilities. Providing support for ADSL2+, the DGFV338 also supports an Ethernet WAN port for failover on non-ADSL connections.

ProSafe DGFV338 is the all-in-one solution for all your small to mid-sized business needs.



## Technical Specifications

### • Wireless

- Standards: IEEE, 802.11g, 54 Mbps, Super G technology up to 108 Mbps
- Antenna: (2) 5dBi, detachable

### • Physical Interfaces

- LAN ports: Eight (8) 10/100Mbps auto-sensing, Auto Uplink™ RJ-45 ports
- WAN port: 10/100BASE-T Ethernet RJ-45
- ADSL WAN port: One (1) RJ-11 supporting annex A or Annex B (Region Specific)

### • Security

- SPI firewall: Stateful packet inspection (SPI) Intrusion Detection System (IDS) including logging, reporting and email alerts, address, service and protocol, Web URL keyword filtering, port/service blocking. Advanced features include block Java/URL/ActiveX based on extension, FTP/SMTP/RPC program filtering
- VPN Functionality: Fifty (50) dedicated VPN tunnels, Manual Key and Internet Key Exchange Security Association (IKE SA) assignment with pre-shared key and RSA/DSA signatures, key life and IKE lifetime settings, perfect forward secrecy (Diffie-Hellman groups 1 and 2 and Oakley support), operating modes (Main, Aggressive, Quick), fully qualified domain name (FQDN) support for dynamic IP address VPN connections
- IPSec Support: IPSec-based 56-bit (DES), 168-bit (3DES), or 256-bit (AES) encryption algorithm, MD5 or SHA-1 hashing algorithm, AH/AH-ESP support, PKI features with X.509 v.3 certificate support, remote access VPN (client-to-site), site-to-site VPN, IPSec NAT traversal (VPN pass-through)
- Mode of operation: One-to-one/multi-to-one multi-network address translation (NAT), classical routing, unrestricted users per port
- IP address assignment: Static IP address assignment, DHCP replay, DHCP address reservation
- Wireless Security: Wi-Fi Protected Access 2 Enterprise (WPA2), 802.1x

### • Performance Features:

- Throughput: Up to 50 Mbps WAN-to-LAN, up to 20 Mbps for 3DES throughput

### • Management Features

- Administration interface: SNMP (v2c) support, Web graphic user interface, Secure Sockets Layer (SSL) remote management, user name and password protected; Secure Remote Management support authenticated through IP address or IP address range and password; configuration changes/upgrades through web GUI
- Configuration and upgrades: Upload and download configuration settings, firmware upgradeable flash memory
- Secure HTTPS remote management support: Authenticated through IP address or IP range and password

### • Functions

- VPN Wizard to simplify configuration of the VPN, port range forwarding, port triggering, exposed host (DMZ), enable/disable WAN ping, DNS proxy, MAC address cloning/spoofing, Network Time Protocol support, keyword content filtering, email alerts, DHCP server (info and display table), WAN DHCP client, diagnostic tools (ping, trace route, other), port/service/MAC address blocking

### • Protocol Support

- Network: IP routing, TCP/IP, UDP, ICMP, PPPoE
- IP Addressing: DHCP (client and server)
- Routing: Static, RIPv1, RIPv2
- VPN/Security: IPSec (ESP, AH), MD5, SHA-1, DES, 3DES, IKE, PKI, AES

### • User Support

- LAN: Up to 253 users
- WLAN: Up to 20 users

### • Hardware

- Processor: 266 Mhz 32-bit RISC
- ADSL Processor: T. 1. AR7
- Memory: 8 MB flash, 32 MB SDRAM
- Power requirements: 100-240V AC, 50-60Hz

- Power adapter: 12V DC, 1.2A Plug is localized to country of sale
- Dimensions: 280 x 245 x 106 mm (11.02 x 9.65 x 4.17 in)
- Weight: 2.19 kg/4.83 lb
- MTBF: 35141.413 hours = 4.012 years
- Acoustic noise: 0 dB

### • Environmental Specifications

- Operating temperature: 0° to 40°C (32° to 104°F)
- Operating humidity: 90% maximum relative humidity, non-condensing

### • System Requirements

- Cable, DSL, and Internet service
- Ethernet connectivity from broadband modem
- Wired or wireless capability for each PC
- Network software (e.g. Windows®)
- Internet Explorer 6.0 or higher, Netscape Navigator 7.2 or higher, Mozilla Firefox® 1.5 or higher

### Warranty

- NETGEAR Lifetime Warranty†

### ProSupport™ Service Packs Available

#### • On Call 24 x 7

- PMB0331-100 (US)
- PMB0331 (non-US)

#### • Xpress HW

- PRR0331

### Package Contents

- ProSafe Wireless ADSL Modem VPN Firewall Router (DGFV338)
- Ethernet cable
- Two (2) detachable 5 dBi antenna
- Installation guide
- Resource CD with single user ProSafe VPN Client Software license
- Warranty/support information card

### Ordering Information

- Europe - General: DGFV338 - 100ISS
- United Kingdom: DGFV338 - 100UKS
- Germany: DGFV338 - 100GRS
- Australia: DGFV338 - 100AUS

# NETGEAR®

4500 Great America Parkway  
Santa Clara, CA 95054 USA  
1-888-NETGEAR (638-4327)  
E-mail: info@NETGEAR.com  
www.NETGEAR.com

©2007 NETGEAR, Inc., the NETGEAR logo, Connect with Innovation, Everybody's connecting, FrontView, the Gear Guy logo, IntelliFi, ProSafe, RangeMax, ReadyNAS, RAIDar, RAIDiator, X-RAID, and Smart Wizard are trademarks of NETGEAR, Inc. and/or its subsidiaries in the United States and/or other countries. Microsoft and Windows are trademarks of Microsoft Corporation in the United States and/or other countries. Other brand and product names are trademarks or registered trademarks of their respective holders. Information is subject to change without notice. All rights reserved.

\*Free basic installation support provided for 90 days from date of purchase. Advanced product features and configurations are not included in free basic installation support; optional premium support available.

†Lifetime warranty for product purchased after 05/01/2007. For product purchased before 05/01/2007, warranty is 3 years.



**MikroTik RouterOS™ v2.9**  
**Reference Manual**

# Installing RouterOS with CD-Install

Document revision 1.2 (Tue Jul 13 13:06:16 GMT 2004)

This document applies to MikroTik RouterOS V2.9

## Table of Contents

[Table of Contents](#)

[CD-Install](#)

[Description](#)

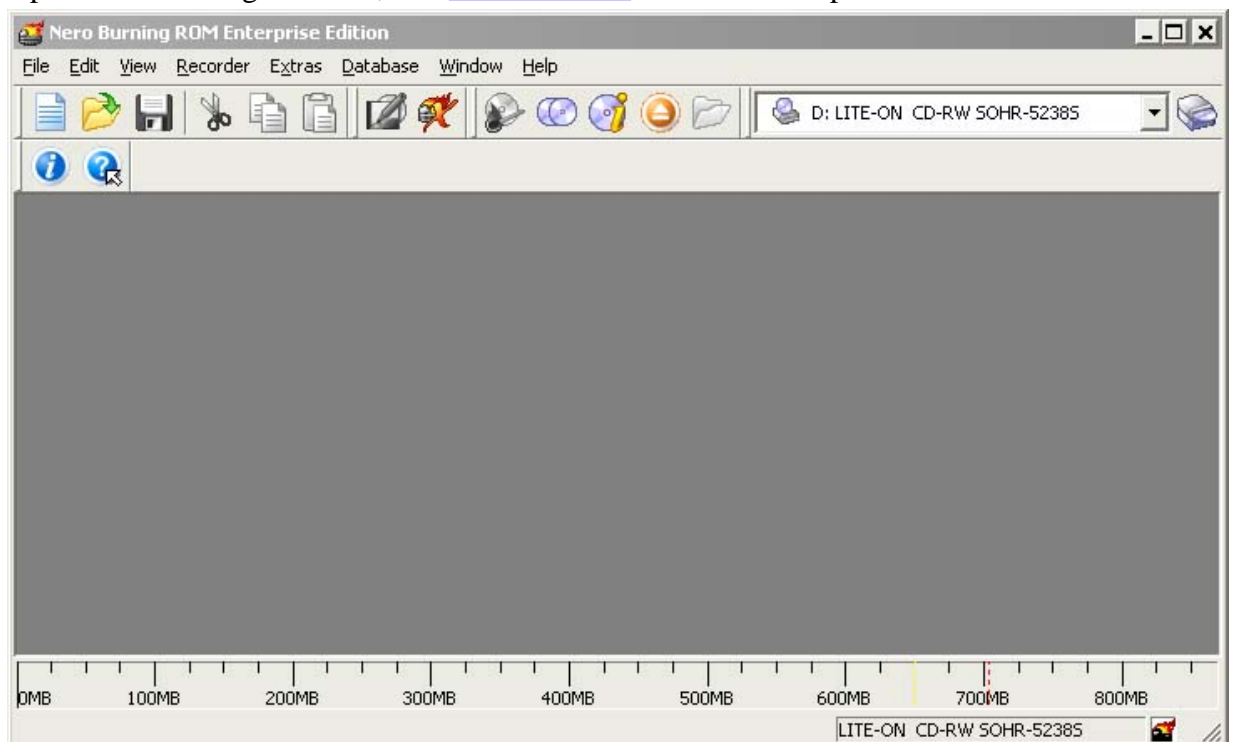
## CD-Install

### Description

To install the RouterOS using a CD you will need a CD-writer and a blank CD. Burn the CD-image (an **.iso** file) to a CD. The archive with image can be downloaded [here](#) .

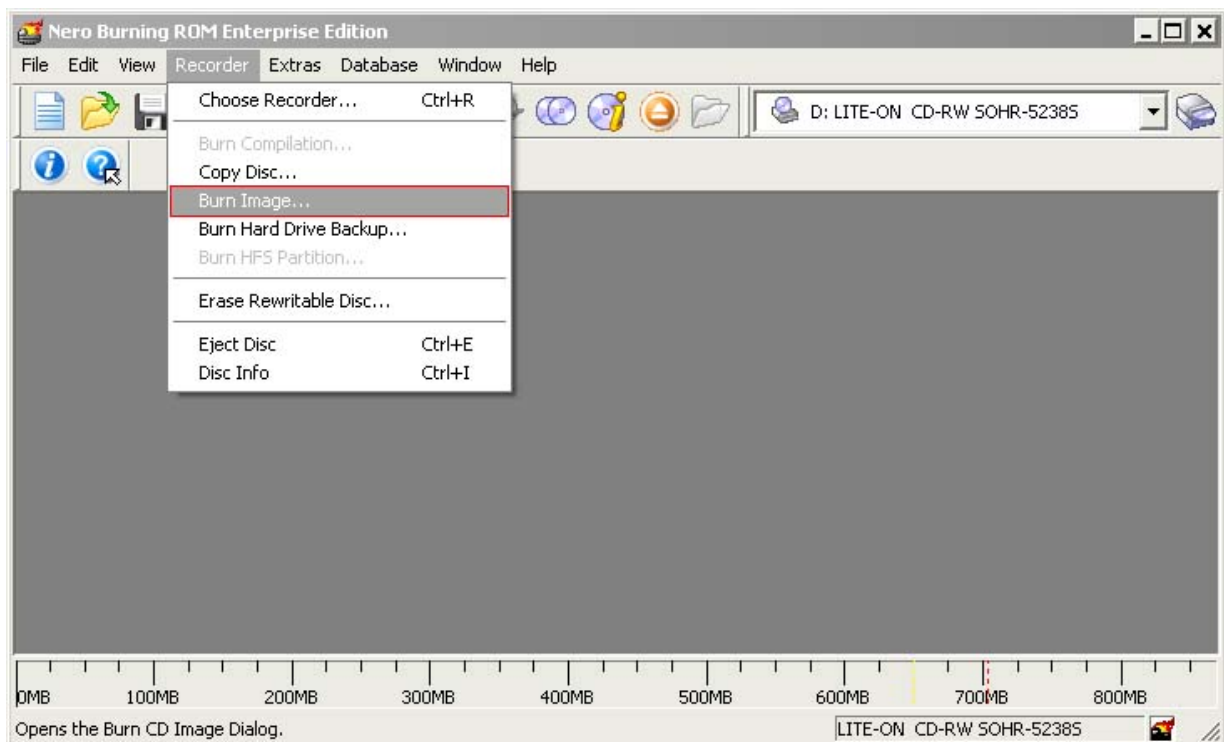
Follow the instructions to install RouterOS using CD-Install:

1. After downloading the CD image from [www.mikrotik.com](http://www.mikrotik.com) you will have an ISO file on your computer:
2. Open a CD Writing software, like [Ahead NERO](#) as in this example:

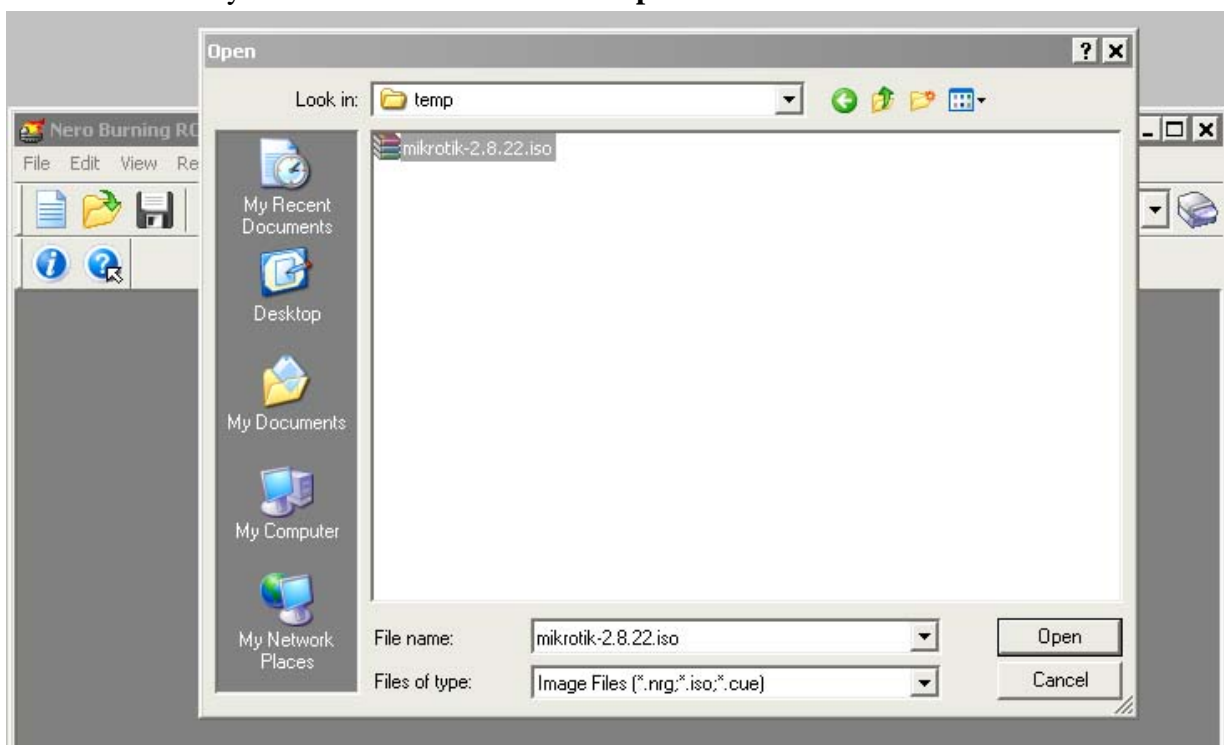


3. In the program, choose **Burn Image** entry from the **Recorder** menu (there should be similiary named option in all major CD burning programs):

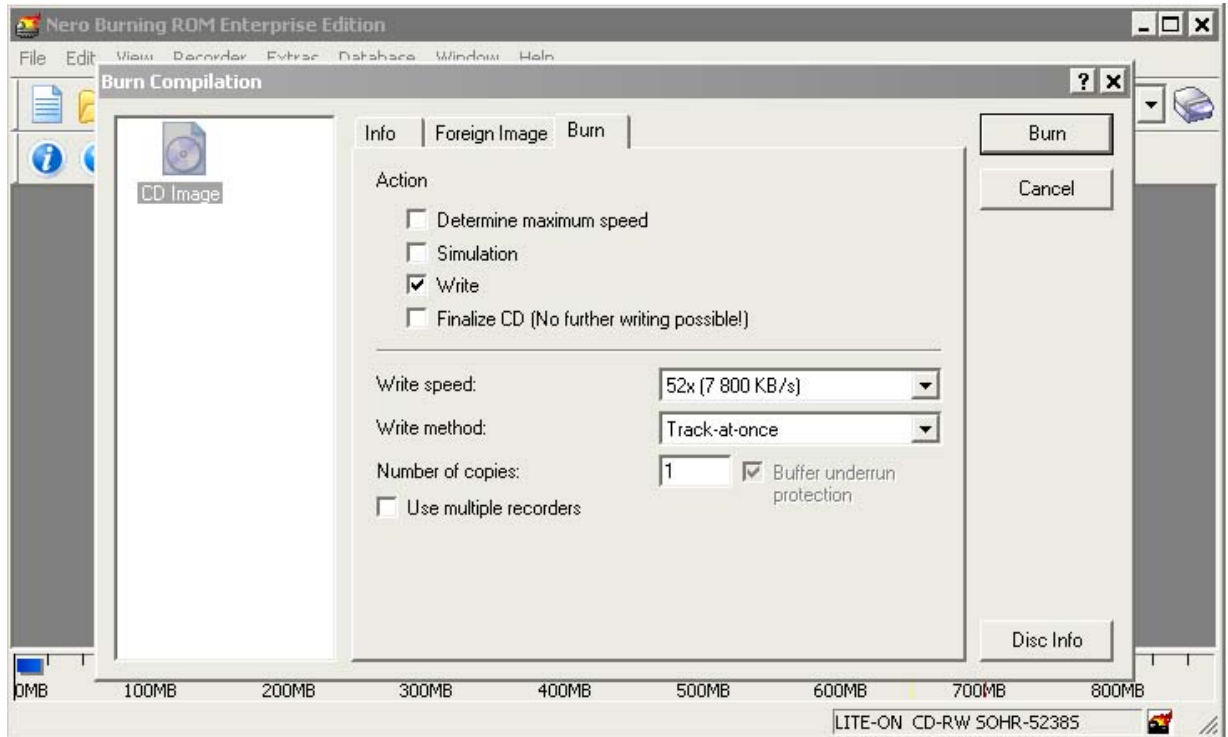




4. Select the recently extracted ISO file and click **Open**:



5. Finally, click **Burn** button:



6. Set the first boot device to CDROM in router's BIOS.
7. After booting from CD you will see a menu where to choose packages to install:

Welcome to MikroTik Router Software installation

Move around menu using 'p' and 'n' or arrow keys, select with 'spacebar'.  
 Select all with 'a', minimum with 'm'. Press 'i' to install locally or 'r' to  
 install remote router or 'q' to cancel and reboot.

```
[X] system           [ ] isdn             [ ] synchronous
[X] ppp              [ ] lcd              [ ] telephony
[X] dhcp             [ ] ntp              [ ] ups
[X] advanced-tools  [ ] radiolan         [ ] web-proxy
[ ] arlan            [ ] routerboard     [ ] wireless
[ ] gps              [X] routing
[ ] hotspot         [X] security
```

Follow the instructions, select needed packages, and press 'i' to install the software.

8. You will be asked for 2 questions:

Warning: all data on the disk will be erased!

Continue? [y/n]

Press [Y] to continue or [N] to abort the installation.

Do you want to keep old configuration? [y/n]:

You should choose whether you want to keep old configuration (press [Y]) or to erase the configuration permanently (press [N]) and continue without saving it. For a fresh installation, press [N].

Creating partition...  
 Formatting disk...

The system will install selected packages. After that you will be prompted to press 'Enter'.

Before doing that, remove the CD from your CD-Drive:

Software installed.  
Press ENTER to reboot

**Note:** after the installation you will have to enter the Software key. See [this](#) manual how to do it.

# Terminal Console

Document revision 1.0 (Mon Nov 8 13:15:54 GMT 2004)

This document applies to MikroTik RouterOS V2.9

## Table of Contents

### [Table of Contents](#)

[Summary](#)

[Specifications](#)

[Related Documents](#)

### [Common Console Functions](#)

[Description](#)

[Example](#)

### [Lists and Item Names](#)

[Description](#)

[Notes](#)

[Example](#)

### [Quick Typing](#)

[Description](#)

[Notes](#)

### [Additional Information](#)

[Description](#)

### [General Commands](#)

[Description](#)

[Command Description](#)

### [Safe Mode](#)

[Description](#)

## General Information

### Summary

The Terminal Console is used for accessing the MikroTik Router's configuration and management features using text terminals, *id est* remote terminal clients or locally attached monitor and keyboard. The Terminal Console is also used for writing scripts. This manual describes the general console operation principles. Please consult the Scripting Manual on some advanced console commands and on how to write scripts.

### Specifications

Packages required: *system*

License required: *level1*

Hardware usage: *Not significant*

### Related Documents

- [\*Scripting Host and Complementary Tools\*](#)

# Common Console Functions

## Description

The console allows configuration of the router's settings using text commands. Although the command structure is similar to the Unix shell, you can get additional information about the command structure in the **Scripting Host and Complementary Tools** manual. Since there is a lot of available commands, they are split into groups organized in a way of hierarchical menu levels. The name of a menu level reflects the configuration information accessible in the relevant section, *exempli gratia* **/ip hotspot**.

In general, all menu levels hold the same commands. The difference is expressed mainly in command parameters.

## Example

For example, you can issue the **/ip route print** command:

```
[admin@MikroTik] > /ip route print
Flags: A - active, X - disabled, I - invalid, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, d - dynamic
#   DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0 ADC 1.1.1.0/24
1 A S 2.2.2.0/24      r 1.1.1.2      0         isp2
2 ADC 3.3.3.0/24
3 ADC 10.1.0.0/24
4 A S 0.0.0.0/0       r 10.1.0.1     0         isp1

[admin@MikroTik] >
```

Instead of typing **ip route** path before each command, the path can be typed only once to move into this particular branch of menu hierarchy. Thus, the example above could also be executed like this:

```
[admin@MikroTik] > ip route
[admin@MikroTik] ip route> print
Flags: A - active, X - disabled, I - invalid, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, d - dynamic
#   DST-ADDRESS      G GATEWAY      DISTANCE  INTERFACE
0 ADC 1.1.1.0/24
1 A S 2.2.2.0/24      r 1.1.1.2      0         isp2
2 ADC 3.3.3.0/24
3 ADC 10.1.0.0/24
4 A S 0.0.0.0/0       r 10.1.0.1     0         isp1

[admin@MikroTik] ip route>
```

Notice that the prompt changes in order to reflect where you are located in the menu hierarchy at the moment. To move to the top level again, type **/**:

```
[admin@MikroTik] > /ip route
[admin@MikroTik] ip route> /
[admin@MikroTik] >
```

To move up one command level, type **..**:

```
[admin@MikroTik] ip route> ..
[admin@MikroTik] ip>
```

You can also use **/** and **..** to execute commands from other menu levels without changing the current level:

```

[admin@MikroTik] ip route> /ping 10.0.0.1
10.0.0.1 ping timeout
2 packets transmitted, 0 packets received, 100% packet loss
[admin@MikroTik] ip firewall nat> .. service-port print
Flags: X - disabled, I - invalid
#   NAME                                PORTS
0   ftp                                  21
1   tftp                                  69
2   irc                                  6667
3 X h323
4   quake3
5   mms
6   gre
7   pptp
[admin@MikroTik] ip firewall nat>

```

## Lists and Item Names

### Description

#### Lists

Many of the command levels operate with arrays of items: interfaces, routes, users etc. Such arrays are displayed in similarly looking lists. All items in the list have an item number followed by its parameter values.

To change parameters of an item, you have to specify it's number to the set command.

#### Item Names

Some lists have items that have specific names assigned to each. Examples are **interface** or **user** levels. There you can use item names instead of item numbers.

You do not have to use the **print** command before accessing items by name. As opposed to numbers, names are not assigned by the console internally, but are one of the items' properties. Thus, they would not change on their own. However, there are all kinds of obscure situations possible when several users are changing router's configuration at the same time. Generally, item names are more "stable" than the numbers, and also more informative, so you should prefer them to numbers when writing console scripts.

#### Notes

Item numbers are assigned by **print** command and are not constant - it is possible that two successive **print** commands will order items differently. But the results of last **print** commands are memorized and thus, once assigned, item numbers can be used even after **add**, **remove** and **move** operations (after **move** operation item numbers are moved with the items). Item numbers are assigned on per session basis, they will remain the same until you quit the console or until the next **print** command is executed. Also, numbers are assigned separately for every item list, so **ip address print** would not change numbers for **interface** list.

#### Example

```

[admin@MikroTik] interface> set 0 mtu=1200
ERROR: item number must be assigned by a print command
use print command before using an item number in a command
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
#   NAME           TYPE           RX-RATE   TX-RATE   MTU
0   R Public       ether          0         0         1500
1   R Local        ether          0         0         1500
2   R wlan1       wlan           0         0         1500
[admin@MikroTik] interface> set 0
disabled mtu name rx-rate tx-rate
[admin@MikroTik] interface> set 0 mtu=1200
[admin@MikroTik] interface> set wlan1 mtu=1300
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
#   NAME           TYPE           RX-RATE   TX-RATE   MTU
0   R Public       ether          0         0         1200
1   R Local        ether          0         0         1500
2   R wlan1       wlan           0         0         1300
[admin@MikroTik] interface>

```

## Quick Typing

### Description

There are two features in the console that help entering commands much quicker and easier - the [Tab] key completions, and abbreviations of command names. Completions work similarly to the **bash** shell in UNIX. If you press the [Tab] key after a part of a word, console tries to find the command within the current context that begins with this word. If there is only one match, it is automatically appended, followed by a space:

```
/inte[Tab]_ becomes /interface _
```

If there is more than one match, but they all have a common beginning, which is longer than that what you have typed, then the word is completed to this common part, and no space is appended:

```
/interface set e[Tab]_ becomes /interface set ether_
```

If you've typed just the common part, pressing the tab key once has no effect. However, pressing it for the second time shows all possible completions in compact form:

```

[admin@MikroTik] > interface set e[Tab]_
[admin@MikroTik] > interface set ether[Tab]_
[admin@MikroTik] > interface set ether[Tab]_
ether1 ether5
[admin@MikroTik] > interface set ether_

```

The [Tab] key can be used almost in any context where the console might have a clue about possible values - command names, argument names, arguments that have only several possible values (like names of items in some lists or name of protocol in firewall and NAT rules). You cannot complete numbers, IP addresses and similar values.

Another way to press fewer keys while typing is to abbreviate command and argument names. You can type only beginning of command name, and, if it is not ambiguous, console will accept it as a full name. So typing:

```

[admin@MikroTik] > pi 10.1 c 3 si 100

equals to:

[admin@MikroTik] > ping 10.0.0.1 count 3 size 100

```

## Notes

Pressing [Tab] key while entering IP address will do a DNS lookup, instead of completion. If what is typed before cursor is a valid IP address, it will be resolved to a DNS name (reverse resolve), otherwise it will be resolved directly (i.e. to an IP address). To use this feature, DNS server must be configured and working. To avoid input lockups any such lookup will timeout after half a second, so you might have to press [Tab] several times, before the name is actually resolved.

It is possible to complete not only beginning, but also any distinctive substring of a name: if there is no exact match, console starts looking for words that have string being completed as first letters of a multiple word name, or that simply contain letters of this string in the same order. If single such word is found, it is completed at cursor position. For example:

```
[admin@MikroTik] > interface x[TAB]_  
[admin@MikroTik] > interface export _  
  
[admin@MikroTik] > interface mt[TAB]_  
[admin@MikroTik] > interface monitor-traffic _
```

## Additional Information

### Description

### Built-in Help

The console has a built-in help, which can be accessed by typing ?. General rule is that help shows what you can type in position where the ? was pressed (similarly to pressing [Tab] key twice, but in verbose form and with explanations).

### Internal Item Numbers

You can specify multiple items as targets to some commands. Almost everywhere, where you can write the number of item, you can also write a list of numbers:

```
[admin@MikroTik] > interface print  
Flags: X - disabled, D - dynamic, R - running  
#   NAME           TYPE           MTU  
0 R ether1         ether          1500  
1 R ether2         ether          1500  
2 R ether3         ether          1500  
3 R ether4         ether          1500  
[admin@MikroTik] > interface set 0,1,2 mtu=1460  
[admin@MikroTik] > interface print  
Flags: X - disabled, D - dynamic, R - running  
#   NAME           TYPE           MTU  
0 R ether1         ether          1460  
1 R ether2         ether          1460  
2 R ether3         ether          1460  
3 R ether4         ether          1500  
[admin@MikroTik] >
```

## General Commands



## Description

There are some commands that are common to nearly all menu levels, namely: **print**, **set**, **remove**, **add**, **find**, **get**, **export**, **enable**, **disable**, **comment**, **move**. These commands have similar behavior throughout different menu levels.

## Command Description

**print** - shows all information that's accessible from particular command level. Thus, /system clock print shows system date and time, /ip route print shows all routes etc. If there's a list of items in current level and they are not read-only, i.e. you can change/remove them (example of read-only item list is /system history, which shows history of executed actions), then print command also assigns numbers that are used by all commands that operate with items in this list. - applicable only to lists of items. The action is performed with all items in this list in the same order in which they are given. - forces the print command to use tabular output form - forces the print command to use property=value output form - shows the number of items - prints the contents of the specific submenu into a file. This file will be available in the router's ftp - shows the output from the print command for every interval seconds - prints the oid value, which is useful for SNMP - prints the output without paging, to see printed output which does not fit in the screen, use [Shift]+[PgUp] key combination

It is possible to sort print output. Like this:

```
[admin@MikroTik] interface> print type=ether
Flags: X - disabled, D - dynamic, R - running
#   NAME           TYPE           RX-RATE   TX-RATE   MTU
0   R isp1         ether         0         0         1500
1   R isp2         ether         0         0         1500
[admin@MikroTik] interface>
```

**set** - allows you to change values of general parameters or item parameters. The set command has arguments with names corresponding to values you can change. Use ? or double [Tab] to see list of all arguments. If there is a list of items in this command level, then set has one action argument that accepts the number of item (or list of numbers) you wish to set up. This command does not return anything.

**add** - this command usually has all the same arguments as set, except the action number argument. It adds a new item with values you have specified, usually to the end of list (in places where order is relevant). There are some values that you have to supply (like the interface for a new route), other values are set to defaults unless you explicitly specify them. - Copies an existing item. It takes default values of new item's properties from another item. If you do not want to make exact copy, you can specify new values for some properties. When copying items that have names, you will usually have to give a new name to a copy - add command returns internal number of item it has added - places a new item before an existing item with specified position. Thus, you do not need to use the move command after adding an item to the list - controls disabled/enabled state of the newly added item(-s) - holds the description of a newly created item

**remove** - removes item(-s) from a list - contains number(-s) or name(-s) of item(-s) to remove.

**move** - changes the order of items in list where one is relevant. Item numbers after move command are left in a consistent, but hardly intuitive order, so it's better to resync them by using print after each move command. - first argument. Specifies the item(-s) being moved. - second argument.

Specifies the item before which to place all items being moved (they are placed at the end of the list if the second argument is omitted).

**find** - The find command has the same arguments as set, and an additional from argument which works like the from argument with the print command. Plus, find command has flag arguments like disabled, invalid that take values yes or no depending on the value of respective flag. To see all flags and their names, look at the top of print command's output. The find command returns internal numbers of all items that have the same values of arguments as specified.

**edit** - this command is in every place that has set command, it can be used to edit values of properties, exempli gratia: [admin@MikroTik] ip route> print  
Flags: A - active, X - disabled, I - invalid, D - dynamic, C - connect, S - static, r - rip, b - bgp, o - ospf, d - dynamic  
# DST-ADDRESS G GATEWAY DISTANCE INTERFACE 0 ADC 1.1.1.0/24 isp2 1 A S 2.2.2.0/24  
r 1.1.1.2 0 isp2 2 ADC 3.3.3.0/24 bonding1 3 ADC 10.1.0.0/24 isp1 4 A S 0.0.0.0/0 r  
10.1.0.1 0 isp1 [admin@MikroTik] ip route> edit 1 gateway

## Safe Mode

### Description

It is possible to change router configuration in a way that will make it not accessible except from local console. Usually this is done by accident, but there is no way to undo last change when connection to router is already cut. Safe mode can be used to minimize such risk.

Safe mode is entered by pressing **[Ctrl]+[X]**. To quit safe mode, press **[Ctrl]+[X]** again.

```
[admin@MikroTik] ip route>[Ctrl]+[X]
[Safe Mode taken]
```

```
[admin@MikroTik] ip route<SAFE>
```

Message **Safe Mode taken** is displayed and prompt changes to reflect that session is now in safe mode. All configuration changes that are made (also from other login sessions), while router is in safe mode, are automatically undone if safe mode session terminates abnormally. You can see all such changes that will be automatically undone tagged with an **F** flag in system history:

```
[admin@MikroTik] ip route>
[Safe Mode taken]
```

```
[admin@MikroTik] ip route<SAFE> add
[admin@MikroTik] ip route<SAFE> /system history print
Flags: U - undoable, R - redoable, F - floating-undo
ACTION BY POLICY
F route added admin write
```

Now, if telnet connection is cut, then after a while (TCP timeout is **9** minutes) all changes that were made while in safe mode will be undone. Exiting session by **[Ctrl]+[D]** also undoes all safe mode changes, while **/quit** does not.

If another user tries to enter safe mode, he's given following message:

```
[admin@MikroTik] >
Hijacking Safe Mode from someone - unroll/release/don't take it [u/r/d]:
```

- **[u]** - undoes all safe mode changes, and puts the current session in safe mode.
- **[d]** - leaves everything as-is.
- **[r]** - keeps all current safe mode changes, and puts current session in a safe mode. Previous

owner of safe mode is notified about this:

```
[admin@MikroTik] ip firewall rule input  
[Safe mode released by another user]
```

If too many changes are made while in safe mode, and there's no room in history to hold them all (currently history keeps up to **100** most recent actions), then session is automatically put out of the safe mode, no changes are automatically undone. Thus, it is best to change configuration in small steps, while in safe mode. Pressing **[Ctrl]+[X]** twice is an easy way to empty safe mode action list.

# Winbox

*Document revision 1.0 (Fri Mar 05 07:59:49 GMT 2004)*

This document applies to MikroTik RouterOS V2.9

## Table of Contents

[Table of Contents](#)

[General Information](#)

[Summary](#)

[Description](#)

[Troubleshooting](#)

[Description](#)

## General Information

### Summary

The MikroTik RouterOS can be configured remotely, using **Telnet**, **SSH**, **WinBox Console** or **Webbox**. In this manual we will discuss how to use the interactive **WinBox** console.

### Description

The Winbox console is used for accessing the MikroTik Router configuration and management features, using graphical user interface (GUI).

All Winbox interface functions are as close as possible to Console functions: all Winbox functions are exactly in the same hierarchy in Terminal Console and vice versa (except functions that are not implemented in Winbox). That is why there are no Winbox sections in the manual.

The Winbox Console plugin loader, the winbox.exe program, can be retrieved from the MikroTik router, the URL is **http://router\_address/winbox/winbox.exe** Use any web browser on Windows 95/98/ME/NT4.0/2000/XP or Linux to retrieve the **winbox.exe** executable file from Router. If your router is not specifically configured, you can also type in the web-browser just **http://router\_address**

The Winbox plugins are cached on the local disk for each MikroTik RouterOS version. The plugins are not downloaded, if they are in the cache, and the router has not been upgraded since the last time it has been accessed.

### Starting the Winbox Console

When connecting to the MikroTik router via http (TCP port 80 by default), the router's Welcome Page is displayed in the web browser:

Login

## Winbox



Winbox is the graphical configuration application for RouterOS. [Download it](#), run it and connect to your router - all RouterOS functionality can be controlled with this application.

## Webbox



This is a web based configuration interface for RouterOS. Log in above to connect to this router - some of the most important RouterOS features can be controlled within this interface.

## Telnet



[Connect with telnet](#) and you will have access to the command line interface of RouterOS, every function of RouterOS can be controlled with it.

## Graphs



These [graphs](#) show you statistical information about your router's interfaces and the traffic that goes through them. Before you use [Graphs](#), [you have to configure them](#).

## Documentation



We have written many tutorials, examples and manuals for RouterOS, all of which are available here [on our homepage](#). If you get into trouble, you can always ask for [technical support](#).

## License

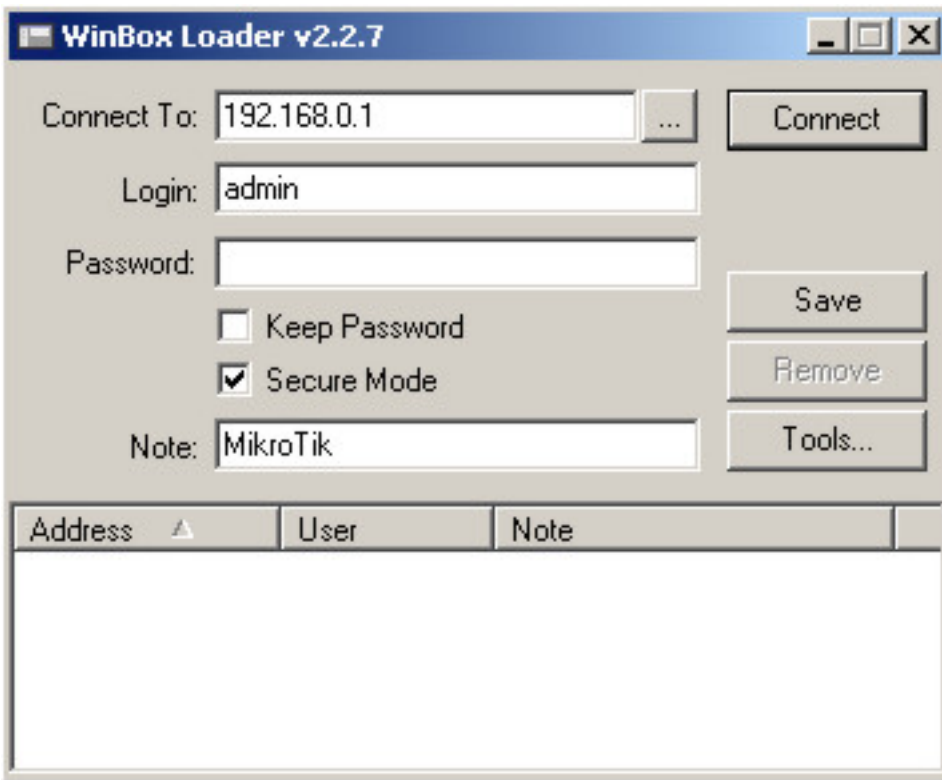


Mikrotik, RouterOS and the Mikrotik logo are registered trademarks of Mikrotik Ltd. Please read [the license](#)


## mikrotik routeros 2.9.5 configuration page

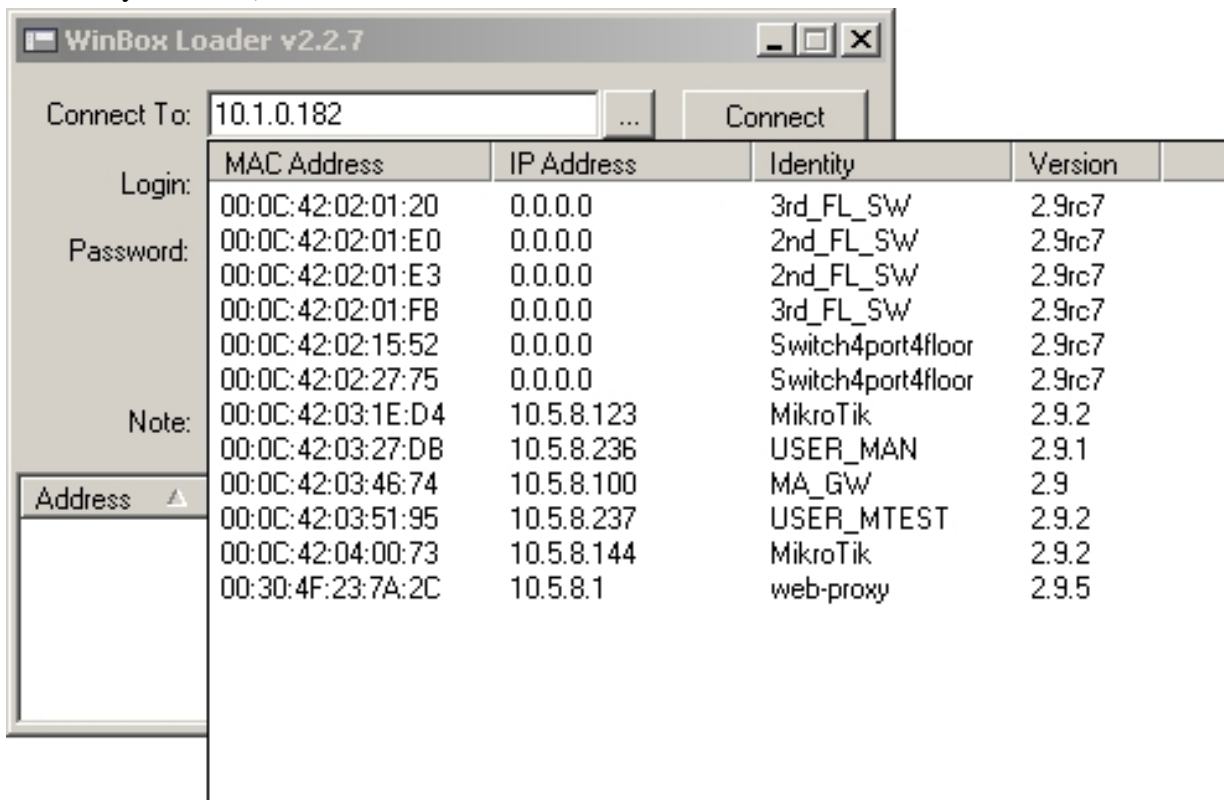
By clicking on the Winbox link you can start the winbox.exe download. Choose **Open** to start the Winbox loader program (you can also save this program to your local disk, and run it from there)

The **winbox.exe** program opens the Winbox login window.

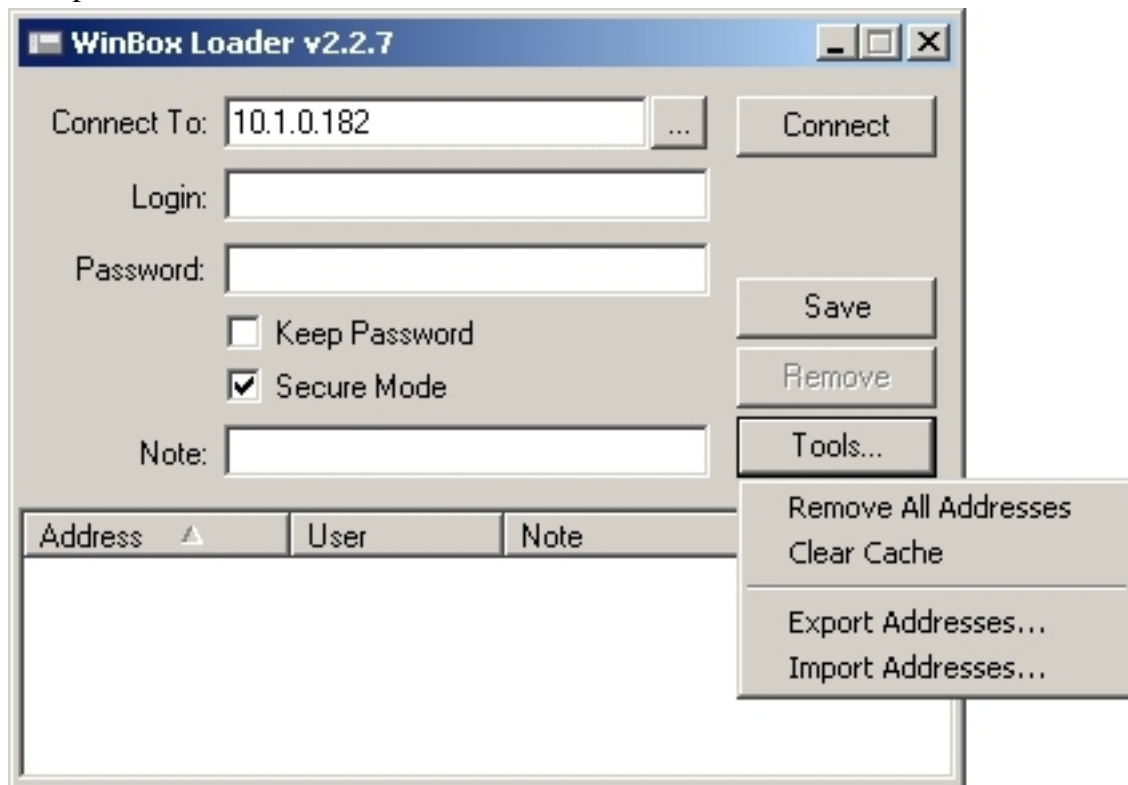


where:

-  discovers and shows MNDP (MikroTik Neighbor Discovery Protocol) or CDP (Cisco Discovery Protocol) devices.

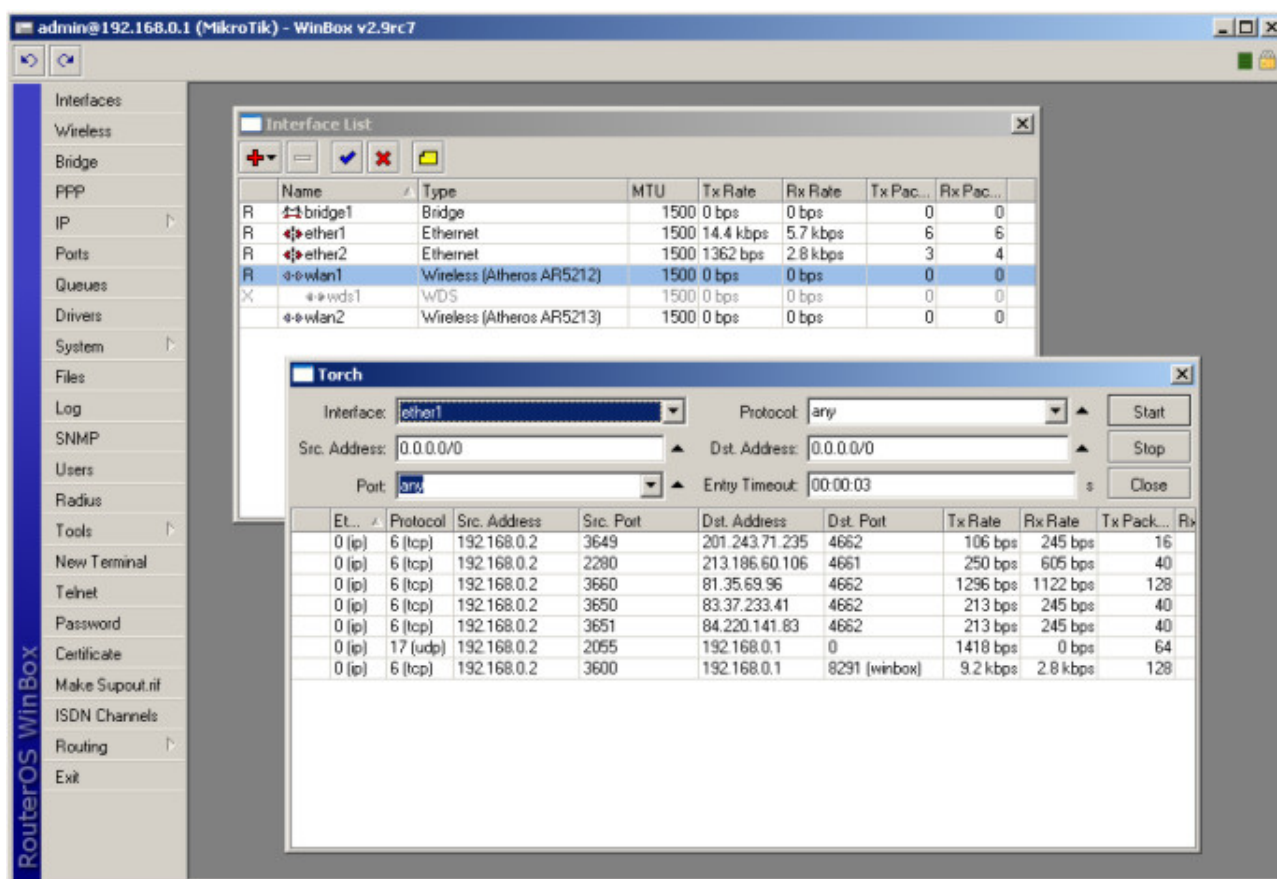


- **Connect**  
logs on to the router by specified IP address (and the port number if you have changed it from the default value of 80) or MAC Address (if the router is in the same subnet), user name, and password.
- **Save**  
saves the current sessions to the list (to run them, just double-click on an item).
- **Remove**  
removes selected item from the list.
- **Tools...**  
removes all items from the list, clears cache on the local disk, imports addresses from wbx file or exports them to wbx file.



- **Secure Mode**  
provides privacy and data integrity between WinBox and RouterOS by means of TLS (Transport Layer Security) protocol.
- **Keep Password**  
Saves password as a plain text on a local hard drive. **Warning:** storing passwords in plain text allows anybody with access to your files to read the password from there.

The Winbox Console of the router:







The Winbox Console uses TCP port 8291. After logging onto the router you can work with the MikroTik router's configuration through the Winbox console and perform the same tasks as using the regular console.






## Overview of Common Functions

You can use the menu bar to navigate through the router's configuration menus, open configuration windows. By double clicking on some list items in the windows you can open configuration windows for the specific items, and so on.

There are some hints for using the Winbox Console:

- To open the required window, simply click on the corresponding menu item
-  Add a new entry
-  Remove an existing entry
-  Enable an item
-  Disable an item



-  Make or edit a comment
-  Refresh a window
-  Undo an action
-  Redo an action
-  Logout from the Winbox Console

## Troubleshooting

### Description

- **Can I run WinBox on Linux?**
- Yes, you can run WinBox and connect to RouterOS, using Wine
- **I cannot open the Winbox Console**  
Check the port and address for **www** service in **/ip service print** list. Make sure the address you are connecting from matches the network you've specified in **address** field and that you've specified the correct port in the Winbox loader. The command **/ip service set www port=80 address=0.0.0.0/0** will change these values to the default ones so you will be able to connect specifying just the correct address of the router in the address field of Winbox loader
- The Winbox Console uses TCP port 8291. Make sure you have access to it through the firewall.

# NAT

Document revision 2.8 (Tue Feb 28 15:15:00 GMT 2006)

This document applies to MikroTik RouterOS V2.9

## Table of Contents

[Table of Contents](#)

[Summary](#)

[Specifications](#)

[Related Documents](#)

[NAT](#)

[Description](#)

[Property Description](#)

[NAT Applications](#)

[Description](#)

[Example of Source NAT \(Masquerading\)](#)

[Example of Destination NAT](#)

[Example of 1:1 mapping](#)

## General Information

### Summary

Network Address Translation (NAT) is a router facility that replaces source and (or) destination IP addresses of the IP packet as it pass through the router. It is most commonly used to enable multiple host on a private network to access the Internet using a single public IP address.

### Specifications

Packages required: *system*

License required: *level1 (number of rules limited to 1) , level3*

Home menu level: */ip firewall nat*

Standards and Technologies: [IP](#) , [RFC1631](#) , [RFC2663](#)

Hardware usage: *Increases with the count of rules*

### Related Documents

- [Software Package Management](#)
- [IP Addresses and ARP](#)
- [Routes, Equal Cost Multipath Routing, Policy Routing](#)
- [Filter](#)
- [Mangle](#)
- [Packet Flow](#)

# NAT

## Description

Network Address Translation is an Internet standard that allows hosts on local area networks to use one set of IP addresses for internal communications and another set of IP addresses for external communications. A LAN that uses NAT is referred as *natted* network. For NAT to function, there should be a NAT gateway in each natted network. The NAT gateway (NAT router) performs IP address rewriting on the way a packet travel from/to LAN.

There are two types of NAT:

- source NAT or *srcnat*. This type of NAT is performed on packets that are originated from a natted network. A NAT router replaces the private source address of an IP packet with a new public IP address as it travels through the router. A reverse operation is applied to the reply packets travelling in the other direction.
- destination NAT or *dstnat*. This type of NAT is performed on packets that are destined to the natted network. It is most comonly used to make hosts on a private network to be accesible from the Internet. A NAT router performing *dstnat* replaces the destination IP address of an IP packet as it travel through the router towards a private network.

## NAT Drawbacks

Hosts behind a NAT-enabled router do not have true end-to-end connectivity. Therefore some Internet protocols might not work in scenarios with NAT. Services that require the initiation of TCP connection from outside the private network or stateless protocols such as UDP, can be disrupted. Moreover, some protocols are inherently incompatible with NAT, a bold example is AH protocol from the IPsec suite.

RouterOS includes a number of so-called NAT helpers, that enable NAT traversal for various protocols.

## Redirect and Masquerade

Redirect and masquerade are special forms of destination NAT and source NAT, respectively. Redirect is similar to the regular destination NAT in the same way as masquerade is similar to the source NAT - masquerade is a special form of source NAT without need to specify **to-addresses** - outgoing interface address is used automatically. The same is for redirect - it is a form of destination NAT where **to-addresses** is not used - incoming interface address is used instead. Note that **to-ports** is meaningful for redirect rules - this is the port of the service on the router that will handle these requests (e.g. web proxy).

When packet is dst-natted (no matter - **action=nat** or **action=redirect**), dst address is changed. Information about translation of addresses (including original dst address) is kept in router's internal tables. Transparent web proxy working on router (when web requests get redirected to proxy port on router) can access this information from internal tables and get address of web server from them. If you are dst-natting to some different proxy server, it has no way to find web server's address from IP header (because dst address of IP packet that previously was address of web server has changed to address of proxy server). Starting from HTTP/1.1 there is special header in HTTP request which

tells web server address, so proxy server can use it, instead of dst address of IP packet. If there is no such header (older HTTP version on client), proxy server can not determine web server address and therefore can not work.

It means, that it is impossible to correctly transparently redirect HTTP traffic from router to some other transparent-proxy box. Only correct way is to add transparent proxy on the router itself, and configure it so that your "real" proxy is parent-proxy. In this situation your "real" proxy does not have to be transparent any more, as proxy on router will be transparent and will forward proxy-style requests (according to standard; these requests include all necessary information about web server) to "real" proxy.

## Property Description

**action** ( *accept* | *add-dst-to-address-list* | *add-src-to-address-list* | *dst-nat* | *jump* | *log* | *masquerade* | *netmap* | *passthrough* | *redirect* | *return* | *same* | *src-nat* ; default: **accept** ) - action to undertake if the packet matches the rule

- **accept** - accepts the packet. No action is taken, i.e. the packet is passed through and no more rules are applied to it
- **add-dst-to-address-list** - adds destination address of an IP packet to the address list specified by address-list parameter
- **add-src-to-address-list** - adds source address of an IP packet to the address list specified by address-list parameter
- **dst-nat** - replaces destination address of an IP packet to values specified by to-addresses and to-ports parameters
- **jump** - jump to the chain specified by the value of the jump-target parameter
- **log** - each match with this action will add a message to the system log
- **masquerade** - replaces source address of an IP packet to an automatically determined by the routing facility IP address
- **netmap** - creates a static 1:1 mapping of one set of IP addresses to another one. Often used to distribute public IP addresses to hosts on private networks
- **passthrough** - ignores this rule goes on to the next one
- **redirect** - replaces destination address of an IP packet to one of the router's local addresses
- **return** - passes control back to the chain from where the jump took place
- **same** - gives a particular client the same source/destination IP address from supplied range for each connection. This is most frequently used for services that expect the same client address for multiple connections from the same client
- **src-nat** - replaces source address of an IP packet to values specified by to-addresses and to-ports parameters

**address-list** ( *name* ) - specifies the name of the address list to collect IP addresses from rules having action=add-dst-to-address-list or action=add-src-to-address-list actions. These address lists could be later used for packet matching

**address-list-timeout** ( *time* ; default: **00:00:00** ) - time interval after which the address will be removed from the address list specified by address-list parameter. Used in conjunction with add-dst-to-address-list or add-src-to-address-list actions

- **00:00:00** - leave the address in the address list forever

**chain** ( *dstnat* | *srcnat* | *name* ) - specifies the chain to put a particular rule into. As the different traffic is passed through different chains, always be careful in choosing the right chain for a new rule. If the input does not match the name of an already defined chain, a new chain will be created

- **dstnat** - a rule placed in this chain is applied before routing. The rules that replace destination addresses of IP packets should be placed there
- **srcnat** - a rule placed in this chain is applied after routing. The rules that replace the source addresses of IP packets should be placed there

**comment** ( *text* ) - a descriptive comment for the rule. A comment can be used to identify rules from scripts

**connection-bytes** ( *integer* | *integer* ) - matches packets only if a given amount of bytes has been transferred through the particular connection

- **0** - means infinity, exempli gratia: connection-bytes=2000000-0 means that the rule matches if more than 2MB has been transferred through the relevant connection

**connection-limit** ( *integer* | *netmask* ) - restrict connection limit per address or address block

**connection-mark** ( *name* ) - matches packets marked via mangle facility with particular connection mark

**connection-type** ( *ftp* | *gre* | *h323* | *irc* | *mms* | *pptp* | *quake3* | *tftp* ) - matches packets from related connections based on information from their connection tracking helpers. A relevant connection helper must be enabled under /ip firewall service-port

**content** ( *text* ) - the text packets should contain in order to match the rule

**dst-address** ( *IP address* | *netmask* | *IP address* | *IP address* ) - specifies the address range an IP packet is destined to. Note that console converts entered address/netmask value to a valid network address, i.e.:1.1.1.1/24 is converted to 1.1.1.0/24

**dst-address-list** ( *name* ) - matches destination address of a packet against user-defined address list

**dst-address-type** ( *unicast* | *local* | *broadcast* | *multicast* ) - matches destination address type of the IP packet, one of the:

- **unicast** - IP addresses used for one point to another point transmission. There is only one sender and one receiver in this case
- **local** - matches addresses assigned to router's interfaces
- **broadcast** - the IP packet is sent from one point to all other points in the IP subnetwork
- **multicast** - this type of IP addressing is responsible for transmission from one or more points to a set of other points

**dst-limit** ( *integer* | *time* | *integer* | *dst-address* | *dst-port* | *src-address* | *time* ) - limits the packet per second (pps) rate on a per destination IP or per destination port base. As opposed to the limit match, every destination IP address / destination port has it's own limit. The options are as follows (in order of appearance):

- **Count** - maximum average packet rate, measured in packets per second (pps), unless followed by Time option
- **Time** - specifies the time interval over which the packet rate is measured
- **Burst** - number of packets to match in a burst
- **Mode** - the classifier(-s) for packet rate limiting
- **Expire** - specifies interval after which recorded IP addresses / ports will be deleted

**dst-port** ( *integer* : 0 ..65535 | *integer* : 0 ..65535 ) - destination port number or range

**hotspot** ( *multiple choice: from-client* | *auth* | *local-dst* ) - matches packets received from clients against various Hot-Spot. All values can be negated

- **from-client** - true, if a packet comes from HotSpot client
- **auth** - true, if a packet comes from authenticated client
- **local-dst** - true, if a packet has local destination IP address

**icmp-options** ( *integer* | *integer* ) - matches ICMP Type:Code fields

**in-interface** ( *name* ) - interface the packet has entered the router through

**ipv4-options** ( *any* | *loose-source-routing* | *no-record-route* | *no-router-alert* | *no-source-routing* | *no-timestamp* | *none* | *record-route* | *router-alert* | *strict-source-routing* | *timestamp* ) - match ipv4 header options

- **any** - match packet with at least one of the ipv4 options
- **loose-source-routing** - match packets with loose source routing option. This option is used to route the internet datagram based on information supplied by the source
- **no-record-route** - match packets with no record route option. This option is used to route the internet datagram based on information supplied by the source
- **no-router-alert** - match packets with no router alter option
- **no-source-routing** - match packets with no source routing option
- **no-timestamp** - match packets with no timestamp option
- **record-route** - match packets with record route option
- **router-alert** - match packets with router alter option
- **strict-source-routing** - match packets with strict source routing option
- **timestamp** - match packets with timestamp

**jump-target** ( *dstnat* | *srcnat* | *name* ) - name of the target chain to jump to, if the action=jump is used

**limit** ( *integer* | *time* | *integer* ) - restricts packet match rate to a given limit. Usefull to reduce the amount of log messages

- **Count** - maximum average packet rate, measured in packets per second (pps), unless followed by Time option
- **Time** - specifies the time interval over which the packet rate is measured
- **Burst** - number of packets to match in a burst

**log-prefix** ( *text* ) - all messages written to logs will contain the prefix specified herein. Used in conjunction with action=log

**nth** ( *integer* | *integer* : 0 ..15 | *integer* ) - match a particular Nth packet received by the rule. One of 16 available counters can be used to count packets

- **Every** - match every Every+1th packet. For example, if Every=1 then the rule matches every 2nd packet
- **Counter** - specifies which counter to use. A counter increments each time the rule containing nth match matches
- **Packet** - match on the given packet number. The value by obvious reasons must be between 0 and Every. If this option is used for a given counter, then there must be at least Every+1 rules with this option, covering all values between 0 and Every inclusively.

**out-interface** ( *name* ) - interface the packet is leaving the router through

**packet-mark** ( *text* ) - matches packets marked via mangle facility with particular packet mark

**packet-size** ( *integer* : 0 ..65535 | *integer* : 0 ..65535 ) - matches packet of the specified size or size range in bytes

- **Min** - specifies lower boundary of the size range or a standalone value
- **Max** - specifies upper boundary of the size range

**phys-in-interface** ( *name* ) - matches the bridge port physical input device added to a bridge device. It is only useful if the packet has arrived through the bridge

**phys-out-interface** ( *name* ) - matches the bridge port physical output device added to a bridge device. It is only useful if the packet will leave the router through the bridge

**protocol** ( *ddp* | *egp* | *encap* | *ggp* | *gre* | *hmp* | *icmp* | *idrp-cmtp* | *igmp* | *ipencap* | *ipip* | *ipsec-ah* | *ipsec-esp* | *iso-tp4* | *ospf* | *pup* | *rdp* | *rspf* | *st* | *tcp* | *udp* | *vmtp* | *xns-idp* | *xtp* | *integer* ) - matches particular IP protocol specified by protocol name or number. You should specify this setting if you want to specify ports

**psd** ( *integer* | *time* | *integer* | *integer* ) - attempts to detect TCP and UDP scans. It is advised to assign lower weight to ports with high numbers to reduce the frequency of false positives, such as from passive mode FTP transfers

- **WeightThreshold** - total weight of the latest TCP/UDP packets with different destination ports coming from the same host to be treated as port scan sequence
- **DelayThreshold** - delay for the packets with different destination ports coming from the same host to be treated as possible port scan subsequence
- **LowPortWeight** - weight of the packets with privileged (<=1024) destination port
- **HighPortWeight** - weight of the packet with non-privileged destination port

**random** ( *integer* ) - match packets randomly with given propability

**routing-mark** ( *name* ) - matches packets marked by mangle facility with particular routing mark

**same-not-by-dst** ( *yes* | *no* ) - specifies whether to account or not to account for destination IP address when selecting a new source IP address for packets matched by rules with action=same

**src-address** ( *IP address* | *netmask* | *IP address* | *IP address* ) - specifies the address range an IP packet is originated from. Note that console converts entered address/netmask value to a valid network address, i.e.:1.1.1.1/24 is converted to 1.1.1.0/24

**src-address-list** ( *name* ) - matches source address of a packet against user-defined address list

**src-address-type** ( *unicast* | *local* | *broadcast* | *multicast* ) - matches source address type of the IP packet, one of the:

- **unicast** - IP addresses used for one point to another point transmission. There is only one sender and one receiver in this case
- **local** - matches addresses assigned to router's interfaces
- **broadcast** - the IP packet is sent from one point to all other points in the IP subnetwork
- **multicast** - this type of IP addressing is responsible for transmission from one or more points to a set of other points

**src-mac-address** ( *MAC address* ) - source MAC address

**src-port** ( *integer* : 0 ..65535 | *integer* : 0 ..65535 ) - source port number or range

**tcp-mss** ( *integer* : 0 ..65535 ) - matches TCP MSS value of an IP packet

**time** ( *time* | *time* | *sat* | *fri* | *thu* | *wed* | *tue* | *mon* | *sun* ) - allows to create filter based on the packets' arrival time and date or, for locally generated packets, departure time and date

**to-addresses** ( *IP address* | *IP address* ; default: **0.0.0.0** ) - address or address range to replace original address of an IP packet with

**to-ports** ( *integer* : 0 ..65535 | *integer* : 0 ..65535 ) - port or port range to replace original port of an IP packet with

**tos** ( *max-reliability* | *max-throughput* | *min-cost* | *min-delay* | *normal* ) - specifies a match to the value of Type of Service (ToS) field of IP header

- **max-reliability** - maximize reliability (ToS=4)
- **max-throughput** - maximize throughput (ToS=8)
- **min-cost** - minimize monetary cost (ToS=2)
- **min-delay** - minimize delay (ToS=16)
- **normal** - normal service (ToS=0)

## NAT Applications

### Description

In this section some NAT applications and examples of them are discussed.

### Basic NAT configuration

Assume we want to create router that:

- "hides" the private LAN "behind" one address
- provides Public IP to the Local server
- creates 1:1 mapping of network addresses

### Example of Source NAT (Masquerading)

If you want to "hide" the private LAN 192.168.0.0/24 "behind" one address 10.5.8.109 given to you by the ISP, you should use the source network address translation (masquerading) feature of the MikroTik router. The masquerading will change the source IP address and port of the packets originated from the network 192.168.0.0/24 to the address 10.5.8.109 of the router when the packet is routed through it.

To use masquerading, a source NAT rule with action 'masquerade' should be added to the firewall configuration:

```
/ip firewall nat add chain=srcnat action=masquerade out-interface=Public
```

All outgoing connections from the network 192.168.0.0/24 will have source address 10.5.8.109 of the router and source port above 1024. No access from the Internet will be possible to the Local addresses. If you want to allow connections to the server on the local network, you should use destination Network Address Translation (NAT).



## Example of Destination NAT

If you want to link Public IP 10.5.8.200 address to Local one 192.168.0.109, you should use destination address translation feature of the MikroTik router. Also if you want allow Local server to talk with outside with given Public IP you should use source address translation, too

Add Public IP to Public interface:

```
/ip address add address=10.5.8.200/32 interface=Public
```

Add rule allowing access to the internal server from external networks:

```
/ip firewall nat add chain=dstnat dst-address=10.5.8.200 action=dst-nat \  
to-addresses=192.168.0.109
```

Add rule allowing the internal server to talk to the outer networks having its source address translated to 10.5.8.200:

```
/ip firewall nat add chain=srcnat src-address=192.168.0.109 action=src-nat \  
to-addresses=10.5.8.200
```

## Example of 1:1 mapping

If you want to link Public IP subnet 11.11.11.0/24 to local one 2.2.2.0/24, you should use destination address translation and source address translation features with **action=netmap**.

```
/ip firewall nat add chain=dstnat dst-address=11.11.11.1-11.11.11.254 \  
action=netmap to-addresses=2.2.2.1-2.2.2.254
```

```
/ip firewall nat add chain=srcnat src-address=2.2.2.1-2.2.2.254 \  
action=netmap to-addresses=11.11.11.1-11.11.11.254
```