MIKROTIK FIREWALL

Ebook ini berupa laporan praktikum Mikrotik, yang saya yakin akan sangat mudah difahami, karna di analisa dengan begitu detail.
Laporan ini di susun oleh rekan-rekan elektro-informatika angkatan 2008.

Semoga ini membantu anda yang ingin mengenal lebih dalam tentang mikrotik. Dokument ini bisa anda sebarluaskan, dengan tidak emngurangi dan menambahkan isi dokumen ini.

TUTORIAL BELAJAR JARINGAN LENGKAP DI
http://emensite.blogspot.com

MODUL VI
MIKROTIK FIREWALL

A. TUJUAN

- Mengetahui *management* di mikrotik*routerboard*.

B. DASAR TEORI

1. *Router*

Router adalah perangkat jaringan yang digunakan untuk membagi *protocol* kepada anggota jaringan yang lainnya, dengan adanya *router*maka sebuah *protocol* dapat di-*sharing* kepada perangkat jaringan lain. Contoh aplikasinya adalah jika kita ingin membagi IP *address* kepada anggota jaringan maka kita dapat menggunakan *router* ini, ciri-ciri *router* adalah adanya fasilitas DHCP (*Dynamic Host Configuration Procotol*), dengan meng*konfigurasi* DHCP, maka kita dapat membagi IP *Address*, fasilitas lain dari *router* adalah adanya NAT (*NetworkAddress Translator*) yang dapat memungkinkan suatu IP *Address* atau koneksi *internet* dis*haring* ke IP *Address* lain.

Router adalah peralatan yang bekerja pada layer 3 OSI(*Open System Interconnection*)dan sering digunakan untuk menyambungkan jaringan luas *Wide Area Network* (WAN) atau untuk melakukan *segmentation* layer 3 di LAN. WAN seperti halnya LANjuga beroperasi di *layer* 1, 2 dan 3 OSI sehingga *router* yang digunakan untukmenyambungkan LAN dan WAN harus mampu mendukung.

Router memiliki kemampuan melewatkan paket IP dari satu jaringan ke jaringan lain yang mungkin memiliki banyak jalur diantara keduanya. *Router-router* yang saling terhubung dalam jaringan *internet* turut serta dalam sebuah algoritma*routing* terdistribusi untuk menentukan jalur terbaik yang dilalui paket IP dari sistemke sistem lain. Proses *routing* dilakukan secara *hop by hop*. IP tidak mengetahui jalurkeseluruhan *menu*ju tujuan setiap paket. IP *routing* hanya menyediakan IP *address*dari *router* berikutnya yang *menu*rutnya lebih dekat ke *host* tujuan.

Menghubungkan komputer dengan komputer lain dapat dilakukan dengan caralangsung menggunakan kabel jaringan ataupun dengan peralatan tambahan. Jika inginmenyambungkan beberapa komputer di

dalam satu ruangan sudah pasti memerlukanperalatan penyambung seperti *hub* atau *switch*.

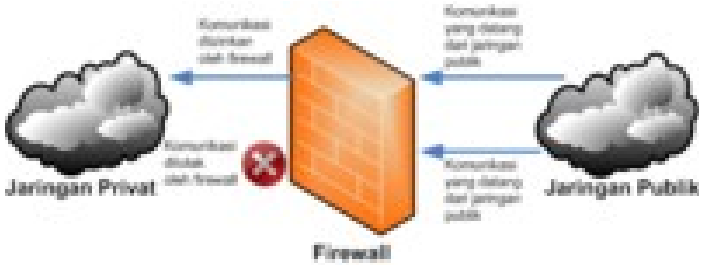
*Hub*ataupun *switch* mempunyai kemampuan untuk menyambungkan pada jarak yang berdekatan berkapasitas *bandwith* mulai dari 10Mbps sampai 1000Mbps. Namun sayang kecepatan tinggi tersebut hanya dapat dinikmati di dalam satu ruangansaja *Local Areal Network* (LAN) . Untuk menyambungkan jaringan dalam saturuangan ke jaringan yang lebih luas memerlukan peralatan yang disebut *router*.

Berhubungan dengan jaringan yang lebih luas atau *internet* berarti akanmenghadapi *internetworking* yang memiliki prinsip dasar sebgai berikut:

a. Pengalamatan secara konsisten
b. Memiliki topologi jaringan mewakili pengalamatan.
c. Pemilihan jalur pengiriman data (terestial, gelombang mikro, satelit, *fiberoptic* dan lainnya).
d. Penggunaan *routerstatic* maupun *dinamic*.
e.Menyambungkan berbagai tempat secara *online* tanpa keterbatasan waktu penyambungan.

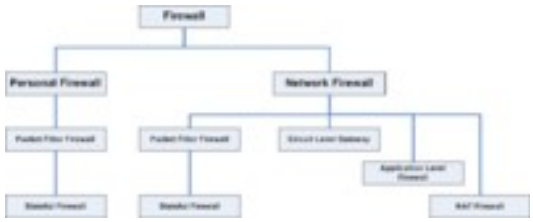
2. *Firewall*

Firewall atau **tembok-api** adalah sebuah sistem atau perangkat yang mengizinkan lalu lintas jaringan yang dianggap aman untuk melaluinya dan mencegah lalu lintas jaringan yang tidak aman. Umumnya, sebuah tembok-api diterapkan dalam sebuah mesin terdedikasi, yang berjalan pada pintu gerbang (*gateway*) antara jaringan lokal dan jaringan lainnya.Tembok-api umumnya juga digunakan untuk mengontrol akses terhadap siapa saja yang memiliki akses terhadap jaringan pribadi dari pihak luar.Saat ini, istilah *firewall* menjadi istilah lazim yang merujuk pada sistem yang mengatur komunikasi antar dua jaringan yang berbeda.Mengingat saat ini banyak perusahaan yang memiliki akses ke *Internet* dan juga tentu saja jaringan berbadan hukum di dalamnya, maka perlindungan terhadap modal *digital* perusahaan tersebut dari serangan para peretas, pemata-mata, ataupun pencuri *data* lainnya, menjadi hakikat.



Gambar 6.1 Ilustrasi mengenai Firewall

- **Jenis-jenis *Firewall***



Gambar 6.2 Taksonomi Firewall

Firewall terbagi menjadi dua jenis, yakni sebagai berikut

- ***PersonalFirewall***: *PersonalFirewall* didesain untuk melindungi sebuah komputer yang terhubung ke jaringan dari akses yang tidak dikehendaki. *Firewall* jenis ini akhir-akhir ini berevolusi menjadi sebuah kumpulan program yang bertujuan untuk mengamankan komputer secara total, dengan ditambahkannya beberapa fitur pengaman tambahan semacam perangkat proteksi terhadap virus, *anti-spyware*, *anti-spam*, dan lainnya. Bahkan beberapa produk *firewall* lainnya dilengkapi dengan fungsi pendeteksian gangguan keamanan jaringan (*Intrusion Detection System*). Contoh dari *firewall* jenis ini adalah Microsoft*WindowsFirewall* (yang telah terintegrasi dalam sistem operasi *Windows* XP *ServicePack* 2, *Windows* Vista dan *WindowsServer* 2003 *ServicePack* 1), Symantec Norton *PersonalFirewall*, Kerio*PersonalFirewall*, dan lain-lain. *PersonalFirewall* secara umum hanya memiliki dua fitur utama, yakni *Packet Filter Firewall* dan *StatefulFirewall*.
- ***NetworkFirewall***: *NetworkFirewall* didesain untuk melindungi jaringan secara keseluruhan dari berbagai serangan. Umumnya dijumpai dalam dua bentuk, yakni sebuah perangkat terdedikasi atau sebagai sebuah perangkat lunak yang diinstalasikan dalam sebuah *server*. Contoh

dari *firewall* ini adalah Microsoft*InternetSecurityandAccelerationServer*(ISA *Server*), Cisco PIX, Cisco ASA, IPTables dalam sistem operasi GNU/Linux, pf dalam keluarga sistem operasi UnixBSD, serta *SunScreen* dari *Sun Microsystems, Inc*. yang dibundel dalam sistem operasiSolaris. *NetworkFirewall* secara umum memiliki beberapa fitur utama, yakni apa yang dimiliki oleh *personalfirewall* (*packetfilterfirewall* dan *statefulfirewall*), *Circuit Level Gateway*, *Application Level Gateway*, dan juga *NAT Firewall*. *NetworkFirewall* umumnya bersifat transparan (tidak terlihat) dari pengguna dan menggunakan teknologi *routing* untuk menentukan paket mana yang diizinkan, dan mana paket yang akan ditolak.

- **Fungsi *Firewall***

Secara *fundamental*, *firewall* dapat melakukan hal-hal berikut:

- Mengatur dan mengontrol lalu lintas jaringan
- Melakukan autentikasi terhadap akses
- Melindungi sumber daya dalam jaringan private
- Mencatat semua kejadian, dan melaporkan kepada *administrator*

- **Mengatur dan Mengontrol Lalu lintas jaringan**

Fungsi pertama yang dapat dilakukan oleh *firewall* adalah *firewall* harus dapat mengatur dan mengontrol lalu lintas jaringan yang diizinkan untuk mengakses jaringan *private* atau komputer yang dilindungi oleh *firewall*.*Firewall* melakukan hal yang demikian, dengan melakukan inspeksi terhadap paket-paket dan memantau koneksi yang sedang dibuat, lalu melakukan penapisan (*filtering*) terhadap koneksi berdasarkan hasil inspeksi paket dan koneksi tersebut.

- **Proses inspeksi Paket**

Inspeksi paket (*packet inspection*) merupakan proses yang dilakukan oleh *firewall* untuk 'menghadang' dan memproses *data* dalam sebuah paket untuk menentukan bahwa paket tersebut diizinkan atau

ditolak, berdasarkan kebijakan akses (*access policy*) yang diterapkan oleh seorang *administrator*. *Firewall*, sebelum menentukan keputusan apakah hendak menolak atau menerima komunikasi dari luar, ia harus melakukan inspeksi terhadap setiap paket (baik yang masuk ataupun yang keluar) di setiap antarmuka dan membandingkannya dengan daftar kebijakan akses. Inspeksi paket dapat dilakukan dengan melihat elemen-elemen berikut, ketika menentukan apakah hendak menolak atau menerima komunikasi:

- Alamat IP dari komputer sumber
- *Port* sumber pada komputer sumber
- Alamat IP dari komputer tujuan
- *Port* tujuan data pada komputer tujuan
- Protokol IP
- In*form*asi *header-header* yang disimpan dalam paket

- **Koneksi dan Keadaan Koneksi**

Agar dua *host* TCP/IP dapat saling berkomunikasi, mereka harus saling membuat koneksi antara satu dengan lainnya. Koneksi ini memiliki dua tujuan:

1. Komputer dapat menggunakan koneksi tersebut untuk mengidentifikasikan dirinya kepada komputer lain, yang meyakinkan bahwa sistem lain yang tidak membuat koneksi tidak dapat mengirimkan *data* ke komputer tersebut. *Firewall* juga dapat menggunakan in*form*asi koneksi untuk menentukan koneksi apa yang diizinkan oleh kebijakan akses dan menggunakannya untuk menentukan apakah paket *data* tersebut akan diterima atau ditolak.
2. Koneksi digunakan untuk menentukan bagaimana cara dua *host* tersebut akan berkomunikasi antara satu dengan yang lainnya (apakah dengan menggunakan koneksi *connection-oriented*, atau *connection-less*).

Gambar 6.3 Ilustrasi mengenai percakapan antara dua buah host

Kedua tujuan tersebut dapat digunakan untuk menentukan keadaan koneksi antara dua *host* tersebut, seperti halnya cara manusia bercakap-cakap. Jika Amir bertanya kepada Aminah mengenai sesuatu, maka Aminah akan meresponsnya dengan jawaban yang sesuai dengan pertanyaan yang diajukan oleh Amir. Pada saat Amir melontarkan pertanyaannya kepada Aminah, keadaan percakapan tersebut adalah Amir *menu*nggu respons dari Aminah. Komunikasi di jaringan juga mengikuti cara yang sama untuk memantau keadaan percakapan komunikasi yang terjadi.

Firewall dapat memantau in*form*asi keadaan koneksi untuk menentukan apakah ia hendak mengizinkan lalu lintas jaringan. Umumnya hal ini dilakukan dengan memelihara sebuah tabel keadaan koneksi (dalam istilah *firewall*: *state table*) yang memantau keadaan semua komunikasi yang melewati *firewall*. Dengan memantau keadaan koneksi ini, *firewall* dapat menentukan apakah *data* yang melewati *firewall* sedang "ditunggu" oleh *host* yang dituju, dan jika ya, aka mengizinkannya. Jika *data* yang melewati *firewall* tidak cocok dengan keadaan koneksi yang didefinisikan oleh tabel keadaan koneksi, maka *data* tersebut akan ditolak. Hal ini umumnya disebut sebagai *Stateful Inspection*.

- ***Stateful Packet Inspection***

Ketika sebuah *firewall* menggabungkan *stateful inspection* dengan *packet inspection*, maka *firewall* tersebut dinamakan dengan ***Stateful Packet Inspection*** (SPI). SPI merupakan proses inspeksi paket yang tidak dilakukan dengan menggunakan struktur paket dan *data* yang terkandung dalam paket, tapi juga pada keadaan apa*host-host* yang

saling berkomunikasi tersebut berada. SPI mengizinkan *firewall* untuk melakukan penapisan tidak hanya berdasarkan isi paket tersebut, tapi juga berdasarkan koneksi atau keadaan koneksi, sehingga dapat mengakibatkan *firewall* memiliki kemampuan yang lebih fleksibel, mudah diatur, dan memiliki skalabilitas dalam hal penapisan yang tinggi.

Salah satu keunggulan dari SPI dibandingkan dengan inspeksi paket biasa adalah bahwa ketika sebuah koneksi telah dikenali dan diizinkan (tentu saja setelah dilakukan inspeksi), umumnya sebuah kebijakan (*policy*) tidak dibutuhkan untuk mengizinkan komunikasi balasan karena *firewall* tahu respons apa yang diharapkan akan diterima. Hal ini memungkinkan inspeksi terhadap *data* dan perintah yang terkandung dalam sebuah paket *data* untuk menentukan apakah sebuah koneksi diizinkan atau tidak, lalu *firewall*akan secara otomatis memantau keadaan percakapan dan secara dinamis mengizinkan lalu lintas yang sesuai dengan keadaan. Ini merupakan peningkatan yang cukup signifikan jika dibandingkan dengan *firewall* dengan inspeksi paket biasa.Apalagi, proses ini diselesaikan tanpa adanya kebutuhan untuk mendefinisikan sebuah kebijakan untuk mengizinkan respons dan komunikasi selanjutnya.Kebanyakan *firewall* modern telah mendukung fungsi ini.

- **Melakukan autentikasi terhadap akses**

Fungsi *fundamentalfirewall* yang kedua adalah *firewall* dapat melakukan autentikasi terhadap akses.Protokol TCP/IP dibangun dengan premis bahwa protokol tersebut mendukung komunikasi yang terbuka. Jika dua *host* saling mengetahui alamat IP satu sama lainnya, maka mereka diizinkan untuk saling berkomunikasi. Pada awal-awal perkembangan *Internet*, hal ini boleh dianggap sebagai suatu berkah.Tapi saat ini, di saat semakin banyak yang terhubung ke *Internet*, mungkin kita tidak mau siapa saja yang dapat berkomunikasi dengan sistem yang kita miliki. Karenanya, *firewall* dilengkapi dengan fungsi autentikasi dengan menggunakan beberapa mekanisme autentikasi, sebagai berikut:

- *Firewall* dapat meminta input dari pengguna mengenai nama pengguna (*user name*) serta kata kunci (*password*). Metode ini sering disebut sebagai *extendedauthentication* atau *xauth*. Menggunakan *xauth* pengguna yang mencoba untuk membuat sebuah koneksi akan diminta input mengenai nama dan kata kuncinya sebelum akhirnya diizinkan oleh *firewall*. Umumnya, setelah koneksi diizinkan oleh kebijakan keamanan dalam *firewall*, *firewall* pun tidak perlu lagi mengisikan input *password* dan namanya, kecuali jika koneksi terputus dan pengguna mencoba menghubungkan dirinya kembali.
- Metode kedua adalah dengan menggunakan sertifikat *digital* dan kunci publik. Keunggulan metode ini dibandingkan dengan metode pertama adalah proses autentikasi dapat terjadi tanpa intervensi pengguna. Selain itu, metode ini lebih cepat dalam rangka melakukan proses autentikasi. Meskipun demikian, metode ini lebih rumit implementasinya karena membutuhkan banyak komponen seperti halnya implementasi infrastruktur kunci publik.
- Metode selanjutnya adalah dengan menggunakan *Pre-SharedKey* (PSK) atau kunci yang telah diberitahu kepada pengguna. Jika dibandingkan dengan sertifikat *digital*, PSK lebih mudah diimplenentasikan karena lebih sederhana, tetapi PSK juga mengizinkan proses autentikasi terjadi tanpa intervensi pengguna. Dengan menggunakan PSK, setiap *host* akan diberikan sebuah kunci yang telah ditentukan sebelumnya yang kemudian digunakan untuk proses autentikasi. Kelemahan metode ini adalah kunci PSK jarang sekali diperbarui dan banyak organisasi sering sekali menggunakan kunci yang sama untuk melakukan koneksi terhadap *host-host* yang berada pada jarak jauh, sehingga hal ini sama saja meruntuhkan proses autentikasi. Agar tercapai sebuah derajat keamanan yang tinggi, umumnya beberapa organisasi juga menggunakan gabungan antara metode PSK dengan xauth atau PSK dengan sertifikat *digital*.

Dengan mengimplementasikan proses autentikasi, *firewall* dapat menjamin bahwa koneksi dapat diizinkan atau tidak. Meskipun jika paket telah diizinkan dengan menggunakan inspeksi paket (PI) atau berdasarkan keadaan koneksi (SPI), jika *host* tersebut tidak lolos proses autentikasi, paket tersebut akan dibuang.

- **Melindungi sumber daya dalam jaringan *private***

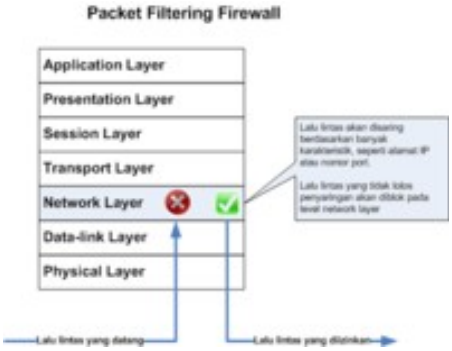
Salah satu tugas *firewall* adalah melindungi sumber daya dari ancaman yang mungkin datang. Proteksi ini dapat diperoleh dengan menggunakan beberapa peraturan pengaturan akses (*accesscontrol*), penggunaan SPI, *applicationproxy*, atau kombinasi dari semuanya untuk mencegah *host* yang dilindungi dapat diakses oleh *host-host* yang mencurigakan atau dari lalu lintas jaringan yang mencurigakan.Meskipun demikian, *firewall* bukanlah satu-satunya metode proteksi terhadap sumber daya, dan mempercayakan proteksi terhadap sumber daya dari ancaman terhadap *firewall* secara eksklusif adalah salah satu kesalahan fatal. Jika sebuah *host* yang menjalankan sistem operasi tertentu yang memiliki lubang keamanan yang belum ditambal dikoneksikan ke *Internet*, *firewall* mungkin tidak dapat mencegah dieksploitasinya *host* tersebut oleh *host-host* lainnya, khususnya jika *exploit* tersebut menggunakan lalu lintas yang oleh *firewall* telah diizinkan (dalam konfigurasinya). Sebagai contoh, jika sebuah *packet-inspectionfirewall* mengizinkan lalu lintas HTTP ke sebuah web *server* yang menjalankan sebuah layanan web yang memiliki lubang keamanan yang belum ditambal, maka seorang pengguna yang "iseng" dapat saja membuat *exploit* untuk meruntuhkan web *server* tersebut karena memang web *server* yang bersangkutan memiliki lubang keamanan yang belum ditambal. Dalam contoh ini, web *server* tersebut akhirnya mengakibatkan proteksi yang ditawarkan oleh *firewall* menjadi tidak berguna.Hal ini disebabkan oleh *firewall* yang tidak dapat membedakan antara *request* HTTP yang mencurigakan atau tidak.Apalagi, jika *firewall* yang digunakan bukan *applicationproxy*.Oleh karena itulah, sumber daya yang dilindungi haruslah dipelihara dengan melakukan penambalan terhadap lubang-lubang keamanan, selain tentunya dilindungi oleh *firewall*.

- **Cara Kerja *Firewall***

Gambar 6.4 Contoh pengaturan akses (access control) yang diterapkan dalam firewall

Pada bentuknya yang paling sederhana, sebuah *firewall* adalah sebuah *router* atau komputer yang dilengkapi dengan dua buah NIC (*NetworkInterfaceCard*, kartu antarmuka jaringan) yang mampu melakukan penapisan atau penyaringan terhadap paket-paket yang masuk. Perangkat jenis ini umumnya disebut dengan *packet-filtering router*.

Firewall jenis ini bekerja dengan cara membandingkan alamat sumber dari paket-paket tersebut dengan kebijakan pengontrolan akses yang terdaftar dalam *AccessControlListfirewall*, *router* tersebut akan mencoba memutuskan apakah hendak meneruskan paket yang masuk tersebut ke tujuannya atau menghentikannya. Pada bentuk yang lebih sederhana lagi, *firewall* hanya melakukan pengujian terhadap alamat IP atau nama domain yang menjadi sumber paket dan akan menentukan apakah hendak meneruskan atau menolak paket tersebut. Meskipun demikian, *packet-filteringrouter* tidak dapat digunakan untuk memberikan akses (atau menolaknya) dengan menggunakan basis hak-hak yang dimiliki oleh pengguna.

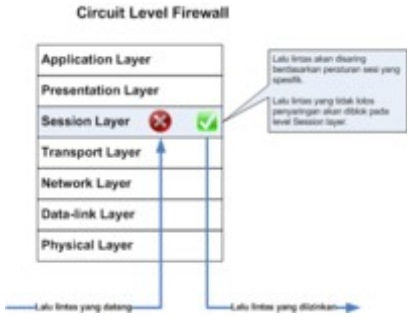


Gambar 6.5 Cara kerja packet filter firewall

Packet-filtering router juga dapat dikonfigurasikan agar menghentikan beberapa jenis lalu lintas jaringan dan tentu saja mengizinkannya.Umumnya, hal ini dilakukan dengan mengaktifkan/menonaktifkan *port* TCP/IP dalam sistem *firewall* tersebut. Sebagai contoh, *port* 25 yang digunakan oleh Protokol SMTP (*Simple Mail Transfer Protocol*) umumnya dibiarkan terbuka oleh beberapa *firewall* untuk mengizinkan surat elektronik dari *Internet* masuk ke dalam jaringan *private*, sementara *port* lainnya seperti *port* 23 yang digunakan

oleh Protokol Telnet dapat dinonaktifkan untuk mencegah pengguna *Internet* untuk mengakses layanan yang terdapat dalam jaringan *private* tersebut. *Firewall* juga dapat memberikan semacam pengecualian (*exception*) agar beberapa aplikasi dapat melewati *firewall* tersebut. Dengan menggunakan pendekatan ini, keamanan akan lebih kuat tapi memiliki kelemahan yang signifikan yakni kerumitan konfigurasi terhadap *firewall*: daftar *AccessControlListfirewall* akan membesar seiring dengan banyaknya alamat IP, nama domain, atau *port* yang dimasukkan ke dalamnya, selain tentunya juga *exception* yang diberlakukan.

- ***CircuitLevelGateway***



Gambar 6.6 Cara kerja circuit level firewall

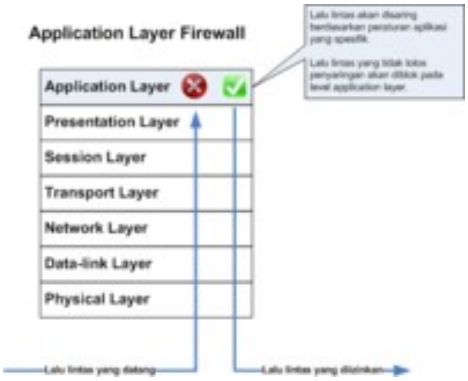
Firewall jenis lainnya adalah *Circuit-LevelGateway*, yang umumnya berupa komponen dalam sebuah *proxy server*. *Firewall* jenis ini beroperasi pada *level* yang lebih tinggi dalam model referensi tujuh lapis OSI (bekerja pada lapisan sesi/session *layer*) daripada *PacketFilterFirewall*. Modifikasi ini membuat *firewall* jenis ini berguna dalam rangka menyembunyikan in*forma*si mengenai jaringan terproteksi, meskipun *firewall* ini tidak melakukan penyaringan terhadap paket-paket individual yang mengalir dalam koneksi.

Dengan menggunakan *firewall* jenis ini, koneksi yang terjadi antara pengguna dan jaringan pun disembunyikan dari pengguna. Pengguna akan dihadapkan secara langsung dengan *firewall* pada saat proses pembuatan koneksi dan *firewall* pun akan membentuk koneksi dengan sumber daya jaringan yang hendak diakses oleh pengguna setelah mengubah alamat IP dari paket yang ditransmisikan oleh dua

belah pihak. Hal ini mengakibatkan terjadinya sebuah sirkuit *virtual* (*virtual circuit*) antara pengguna dan sumber daya jaringan yang ia akses.

Firewall ini dianggap lebih aman dibandingkan dengan *Packet-FilteringFirewall*, karena pengguna eksternal tidak dapat melihat alamat IP jaringan internal dalam paket-paket yang ia terima, melainkan alamat IP dari *firewall*. Protokol yang populer digunakan sebagai *Circuit-LevelGateway* adalah *SOCKS v5.*

- ***ApplicationLevelFirewall***



Gambar 6.7 Application Level Firewall (disebut juga sebagai application proxy atau application level gateway)

Firewall jenis lainnya adalah *ApplicationLevelGateway* (atau *Application-LevelFirewall* atau sering juga disebut sebagai *ProxyFirewall*), yang umumnya juga merupakan komponen dari sebuah *proxyserver*. *Firewall* ini tidak mengizinkan paket yang datang untuk melewati *firewall* secara langsung. Tetapi, aplikasi *proxy* yang berjalan dalam komputer yang menjalankan *firewall*akan meneruskan permintaan tersebut kepada layanan yang tersedia dalam jaringan *private* dan kemudian meneruskan respons dari permintaan tersebut kepada

komputer yang membuat permintaan pertama kali yang terletak dalam jaringan publik yang tidak aman.

Umumnya, *firewall* jenis ini akan melakukan autentikasi terlebih dahulu terhadap pengguna sebelum mengizinkan pengguna tersebut untuk mengakses jaringan. Selain itu, *firewall* ini juga mengimplementasikan mekanisme *auditing* dan pencatatan (*logging*) sebagai bagian dari kebijakan keamanan yang diterapkannya.*ApplicationLevelFirewall* juga umumnya mengharuskan beberapa konfigurasi yang diberlakukan pada pengguna untuk mengizinkan mesin klien agar dapat berfungsi.Sebagai contoh, jika sebuah *proxy*FTP dikonfigurasikan di atas sebuah *applicationlayergateway*, *proxy* tersebut dapat dikonfigurasikan untuk mengizinlan beberapa perintah FTP, dan menolak beberapa perintah lainnya. Jenis ini paling sering diimplementasikan pada *proxy*SMTP sehingga mereka dapat menerima surat elektronik dari luar (tanpa menampakkan alamat *e-mail* internal), lalu meneruskan *e-mail* tersebut kepada *e-mailserver* dalam jaringan. Tetapi, karena adanya pemrosesan yang lebih rumit, *firewall* jenis ini mengharuskan komputer yang dikonfigurasikan sebagai *applicationgateway* memiliki spesifikasi yang tinggi, dan tentu saja jauh lebih lambat dibandingkan dengan *packet-filterfirewall*.

3. Mikrotik

MikroTik RouterOS™, merupakan sistem operasi Linux *base* yang diperuntukkan sebagai *networkrouter*. Didesain untuk memberikan kemudahan bagi penggunanya. Administrasinya bisa dilakukan melalui *WindowsApplication (WinBox)*. Selain itu instalasi dapat dilakukan pada *Standard* komputer PC (*PersonalComputer*). PC yang akan dijadikan *router* mikrotik pun tidak memerlukan *resource* yang cukup besar untuk penggunaan *standard*, misalnya hanya sebagai *gateway*. Untuk keperluan beban yang besar *(network*yang kompleks*, routing*yang rumit) disarankan untuk mempertimbangkan pemilihan*resource* PC yang memadai.

Sejarah MikroTik RouterOS, 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 berimigrasi ke Latvia. Di Latvia ia bejumpa dengan Arnis, seorang sarjana Fisika dan Mekanik sekitar tahun 1995.

John dan Arnis mulai me-*routing* dunia pada tahun 1996 (misi MikroTik adalah me-*routing* 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 *staffResearchandDevelopment* (R&D) MikroTik yang sekarang menguasai dunia *routing* di negara-negara berkembang. Menurut Arnis, selain staf di lingkungan MikroTik, mereka juga merekrut tenega-tenaga lepas dan pihak ketiga yang dengan intensif mengembangkan MikroTik secara *marathon*.

Jenis-jenis mikrotik:

1.MikroTik RouterOS yang berbentuk *software* yang dapat di-*download* di www.mikrotik.com. Dapat diinstal pada komputer rumahan (PC).

2.BUILT-IN *Hardware* MikroTik dalam bentuk perangkat keras yang khusus dikemas dalam *boardrouter* yang didalamnya sudah terinstal MikroTik RouterOS.

Fitur-fitur mikrotik:

1. *AddressList* : Pengelompokan IP *Address* berdasarkan nama
2. *Asynchronous* : Mendukung *serial* PPP *dial-in* / *dial-out*, dengan otentikasi CHAP, PAP, MSCHAPv1 dan MSCHAPv2, Radius, *dial ondemand*, *modempool*hingga 128 *ports*.
3. *Bonding* : Mendukung dalam pengkombinasian beberapa antarmuka *ethernet* ke dalam 1 pipa pada koneksi cepat.

- *Bridge* : Mendukung fungsi *bridgespinningtree*, *multiplebridgeinterface*, *bridgingfirewalling*.

- *DataRateManagement* : QoS berbasis HTB dengan penggunaan *burst*, PCQ, RED, SFQ, FIFO *queue*, CIR, MIR, *limit* antar *peertopeer*
- DHCP : Mendukung DHCP tiap antarmuka; DHCP *Relay*; DHCP *Client*, *multiplenetwork* DHCP; *staticanddynamic* DHCP *leases*.
- *Firewall* dan NAT : Mendukung pemfilteran koneksi *peertopeer*, *source* NAT dan *destination* NAT. Mampu memfilter berdasarkan MAC, IP *address*, *rangeport*, protokol IP, pemilihan opsi protokol seperti ICMP, TCP *Flags* dan MSS.
- *Hotspot* : *Hotspotgateway* dengan otentikasi RADIUS. Mendukung *limitdatarate*, SSL ,HTTPS.
- IPSec : Protokol AH dan ESP untuk IPSec; MODP Diffie-Hellmann *groups* 1, 2, 5; MD5 dan algoritma SHA1 *hashing*; algoritma enkirpsi menggunakan DES, 3DES, AES-128, AES-192, AES-256; *PerfectForwardingSecresy* (PFS) MODP *groups* 1, 2,5

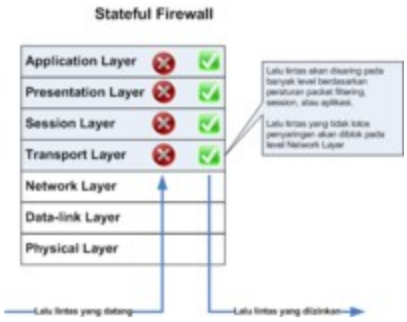
4. ISDN : mendukung ISDN *dial-in/dial-out*. Dengan otentikasi PAP, CHAP, MSCHAPv1 dan MSCHAPv2, Radius. Mendukung 128K *bundle*, *Cisco* HDLC, x751, x75ui, x75bui *line* protokol.
5. M3P : MikroTik Protokol Paket *Packer* untuk *wirelesslinks* dan *ethernet*.
6. MNDP : MikroTik *DiscoveryNeighbour* Protokol, juga mendukung *CiscoDiscoveryProtocol* (CDP).
7. *Monitoring / Accounting* : Laporan *Traffic* IP, *log*, statistik *graph* yang dapat diakses melalui HTTP.
8. NTP :*NetworkTime Protocol* untuk *server* dan *clients*; sinkronisasi menggunakan system GPS.
9. *Poin to PointTunneling Protocol* : PPTP, PPPoE dan L2TP *AccessConsentrator;* protokol otentikasi menggunakan PAP, CHAP, MSCHAPv1, MSCHAPv2; otentikasi dan laporan Radius; enkripsi MPPE; kompresi untuk PPoE; *limitdatarate*.
10. *Proxy* :*Cache* untuk FTP dan HTTP *proxyserver*, HTTPS *proxy*; *transparentproxy* untuk DNS dan HTTP; mendukung protokol *SOCKS*; mendukung *parentproxy*; *static* DNS.
11. *Routing* :*Routingstatic* dan *dinamic*; RIP v1/v2, OSPF v2, BGP v4.
12. SDSL : Mendukung *SingleLine* DSL; *mode* pemutusan jalur koneksi dan jaringan.
13. *SimpleTunnel* : *Tunnel* IPIP dan EoIP (*Ethernetover* IP).

14. SNMP : *SimpleNetworkMonitoringProtocolmode* akses *read-only*.
15. *Synchronous* : V.35, V.24, E1/T1, X21, DS3 (T3) media ttypes; sync-PPP, Cisco HDLC; *FrameRelayline* protokol; ANSI-617d (ANDI atau annex D) dan Q933a (CCITT atau annex A); *FrameRelay* jenis LMI.
16. Tool : *Ping*, *Traceroute*; *bandwidthtest*; *pingflood*; *telnet*; SSH; *packetsniffer*; Dinamik DNS *update*.
17. UPnP : Mendukung antarmuka *UniversalPlugandPlay*.
18. VLAN : Mendukung *Virtual* LAN IEEE 802.1q untuk jaringan *ethernet* dan *wireless*; *multiple* VLAN; VLAN *bridging*.
19. VoIP : Mendukung aplikasi *voiceover* IP.
20. VRRP : Mendukung *VirtualRouterRedudantProtocol*.
21. *WinBox* : Aplikasi *mode* GUI untuk me*remote* dan mengkonfigurasi MikroTik RouterOS.

4. NAT *Firewall*

NAT (*NetworkAddressTranslation*) *Firewall* secara otomatis menyediakan proteksi terhadap sistem yang berada di balik *firewall*karena NAT *Firewall* hanya mengizinkan koneksi yang datang dari komputer-komputer yang berada di balik *firewall*. Tujuan dari NAT adalah untuk melakukan *multiplexing* terhadap lalu lintas dari jaringan internal untuk kemudian menyampaikannya kepada jaringan yang lebih luas (MAN, WAN atau *Internet*) seolah-olah paket tersebut datang dari sebuah alamat IP atau beberapa alamat IP.NAT *Firewall* membuat tabel dalam memori yang mengandung in*form*asi mengenai koneksi yang dilihat oleh *firewall*. Tabel ini akan memetakan alamat jaringan internal ke alamat eksternal. Kemampuan untuk menaruh keseluruhan jaringan di belakang sebuah alamat IP didasarkan terhadap pemetaan terhadap *port-port* dalam NAT *firewall*.

- ***StatefulFirewall***



Gambar6.8Cara kerja statefulfirewall

StatefulFirewall merupakan sebuah *firewall* yang menggabungkan keunggulan yang ditawarkan oleh *packet-filteringfirewall*, NAT *Firewall*, *Circuit-LevelFirewall* dan *ProxyFirewall* dalam satu sistem.*StatefulFirewall* dapat melakukan *filtering* terhadap lalu lintas berdasarkan karakteristik paket, seperti halnya *packet-filteringfirewall*, dan juga memiliki pengecekan terhadap sesi koneksi untuk meyakinkan bahwa sesi koneksi yang terbentuk tersebut diizinlan.Tidak seperti *ProxyFirewall* atau *CircuitLevelFirewall*, *StatefulFirewall* umumnya didesain agar lebih transparan (seperti halnya *packet-filteringfirewall* atau NAT *firewall*). Tetapi, *statefulfirewall* juga mencakup beberapa aspek yang dimiliki oleh *applicationlevelfirewall*, sebab ia juga melakukan inspeksi terhadap *data* yang datang dari lapisan aplikasi (*applicationlayer*) dengan menggunakan layanan tertentu. *Firewall* ini hanya tersedia pada beberapa *firewall* kelas atas, semacam Cisco PIX. Karena menggabungkan keunggulan jenis-jenis *firewall* lainnya, *statefulfirewall* menjadi lebih kompleks.

- ***VirtualFirewall***

VirtualFirewall adalah sebutan untuk beberapa *firewall* logis yang berada dalam sebuah perangkat fisik (komputer atau perangkat *firewall* lainnya).Pengaturan ini mengizinkan beberapa jaringan agar dapat diproteksi oleh sebuah *firewall* yang unik yang menjalankan kebijakan keamanan yang juga unik, cukup dengan menggunakan satu buah perangkat.Dengan menggunakan *firewall* jenis ini, sebuah ISP (*InternetService* Provider) dapat menyediakan layanan *firewall* kepada para pelanggannya, sehingga mengamankan lalu lintas jaringan mereka, hanya dengan menggunakan satu buah perangkat.Hal ini jelas

merupakan penghematan biaya yang signifikan, meski *firewall* jenis ini hanya tersedia pada *firewall* kelas atas, seperti Cisco PIX 535.

- ***TransparentFirewall***

TransparentFirewall (juga dikenal sebagai *bridgingfirewall*) bukanlah sebuah *firewall* yang murni, tetapi ia hanya berupa turunan dari *statefulFirewall*. Daripada *firewall-firewall* lainnya yang beroperasi pada lapisan IP ke atas, *transparentfirewall* bekerja pada lapisan *Data*-Link *Layer*, dan kemudian ia memantau lapisan-lapisan yang ada di atasnya. Selain itu, *transparentfirewall* juga dapat melakukan apa yang dapat dilakukan oleh *packet-filteringfirewall*, seperti halnya *statefulfirewall* dan tidak terlihat oleh pengguna (karena itulah, ia disebut sebagai *TransparentFirewall*).

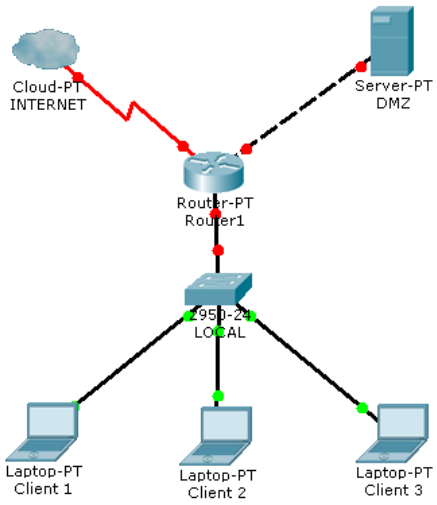
Intinya, *transparentfirewall* bekerja sebagai sebuah *bridge* yang bertugas untuk menyaring lalu lintas jaringan antara dua segmen jaringan.Dengan menggunakan *transparentfirewall*, keamanan sebuah segmen jaringan pun dapat diperkuat, tanpa harus mengaplikasikan NAT *Filter*.*TransparentFirewall* menawarkan tiga buah keuntungan, yakni sebagai berikut:

- Konfigurasi yang mudah (bahkan beberapa produk mengklaim sebagai "*ZeroConfiguration*"). Hal ini memang karena *transparentfirewall* dihubungkan secara langsung dengan jaringan yang hendak diproteksinya, dengan memodifikasi sedikit atau tanpa memodifikasi konfigurasi *firewall* tersebut. Karena ia bekerja pada *data-linklayer*, pengubahan alamat IP pun tidak dibutuhkan. *Firewall* juga dapat dikonfigurasikan untuk melakukan segmentasi terhadap sebuah *subnet* jaringan antara jaringan yang memiliki keamanan yang rendah dan keamanan yang tinggi atau dapat juga untuk melindungi sebuah *host*, jika memang diperlukan.
- Kinerja yang tinggi. Hal ini disebabkan oleh *firewall* yang berjalan dalam lapisan *data-link* lebih sederhana dibandingkan dengan *firewall* yang berjalan dalam lapisan yang lebih tinggi. Karena bekerja lebih sederhana, maka kebutuhan pemrosesan pun lebih kecil dibandingkan dengan *firewall* yang berjalan pada lapisan yang tinggi, dan akhirnya per*forma* yang ditunjukannya pun lebih tinggi.

- Tidak terlihat oleh pengguna (*stealth*). Hal ini memang dikarenakan *TransparentFirewall* bekerja pada lapisan *data-link*, dan tidak membutuhkan alamat IP yang ditetapkan untuknya (kecuali untuk melakukan manajemen terhadapnya, jika memang jenisnya *managed firewall*). Karena itulah, *transparentfirewall* tidak dapat terlihat oleh para penyerang. Karena tidak dapat diraih oleh penyerang (tidak memiliki alamat IP), penyerang pun tidak dapat menyerangnya.

C. Permasalahan

Buatlah konfigurasi komputer seperti gambar dibawah ini.



Gambar 6.9 konfigurasi jaringan untuk permasalahan

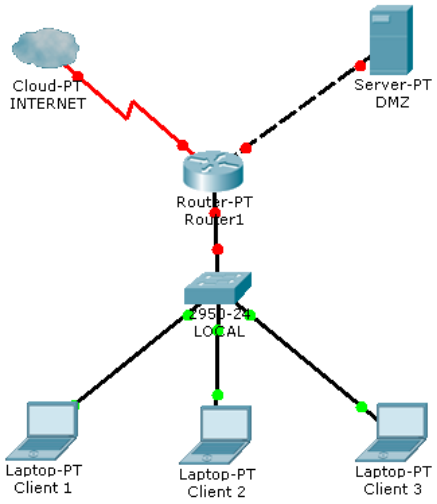
Buatlah konfigurasi sesuai permasalahan dibawah ini

- Semua *client* bisa mengakses *internet* menggunakan DHCP *client* dan setiap melakukan koneksi ke mikrotik akan mendapatkan IP yang selalu sama
- *Client* 1 bisa mengakses *internet* dan hanya bisa mengakses web pada DMZ namun tidak bisa melakukan *ping* ke DMZ.
- *Client* 2 tidak bisa melakukan *browsing* ke *internet* namun bisa mengakses FTP pada DMZ
- *Client* 3 hanya bisa mengakses winbox pada *router* namun tidak bisa mengakses jaringan yang lain.

D. Analisa Data

Dari permasalahan diatas dapat di buat analisa data sebagai berikut :

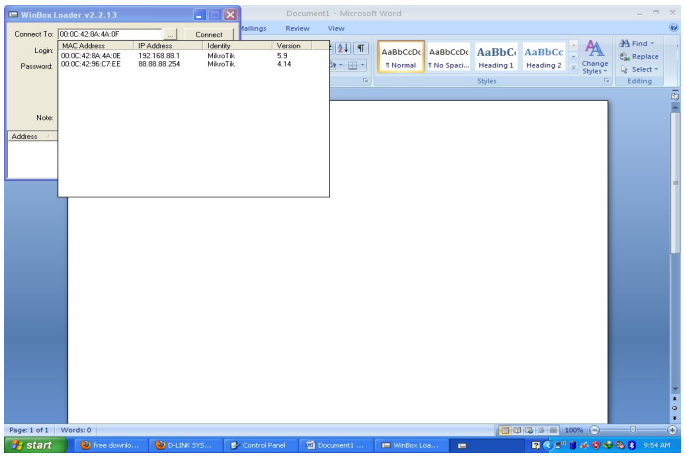
1. Membuat konfigurasi komputer seperti pada gambar :



Gambar 6.10 Konfigurasi jaringan permasalahan

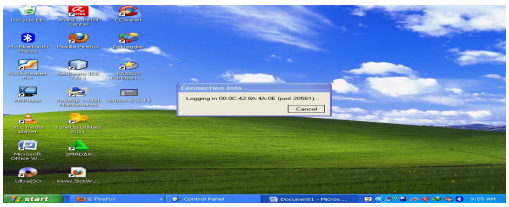
- Me*reset router* mikrotik

  Untuk memulai praktikum, perlu digunakan *router* mikrotik dalam kondisi belum terkonfigurasi. Sehingga mikrotik perlu di*reset* dengan cara sebagai berikut menggunakan *software* konfigurasi mikrotik yakni winbox.



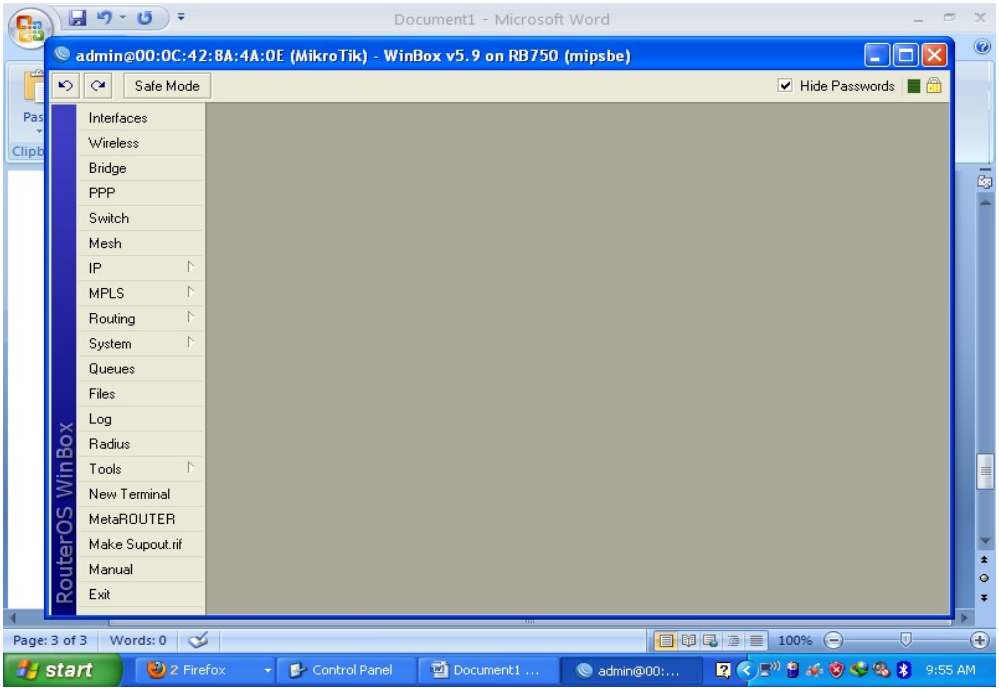
Gambar 6.11 software winbox

Dari gambar diatas, untuk terhubung dengan *router* mikrotik harus mencari MAC *address* dari *router* mikrotik. Pada praktikum kali ini *router* mikrotik memiliki MAC *address* 00:0C:42:8A:4A:0E. setelah memilih MAC *address*, kemudian menekan tombol *connect*.



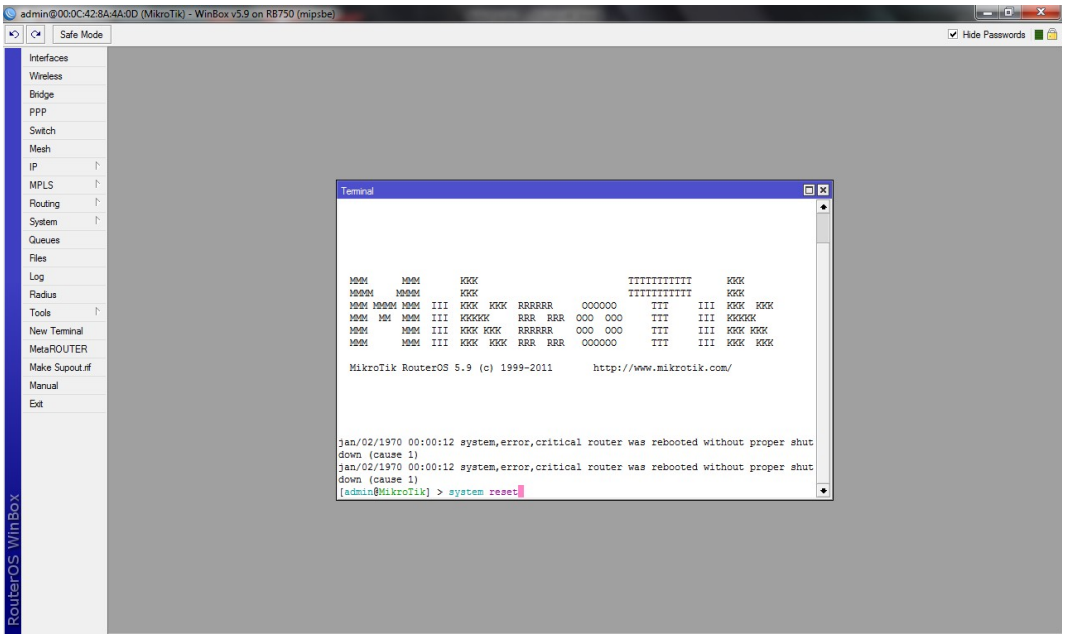
Gambar 6.12 melakukan koneksi ke router mikrotik

Setelah berhasil terhubung ke *router* mikrotik maka akan muncul *menu* konfigurasi seperti pada gambar berikut :



Gambar 6.13 Menu konfigurasi pada Winbox

Kemudian memilih *menu*□*NewTerminal*, mengetikkan perintah `sistem reset` untuk me*reset* konfigurasi pada *router*.

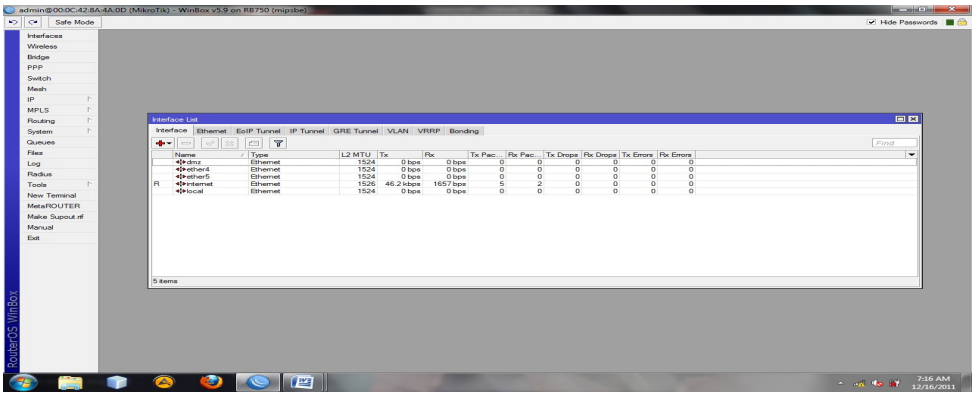


Gambar 6.14 Terminal pada winbox

- Konfigurasi *interface*

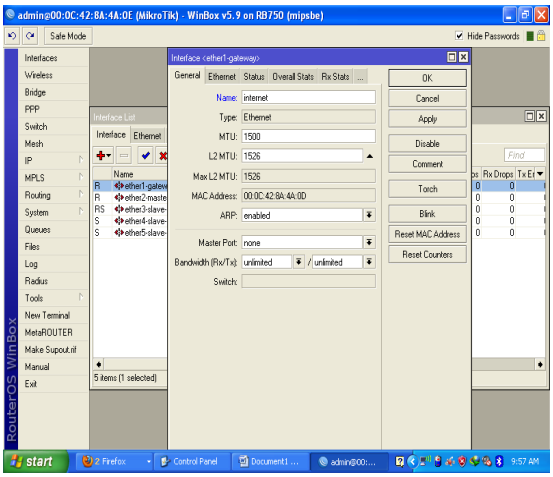
Interface adalah *port-port Ethernet* pada *router* mikrotik. Pada RB750 terdapat 5 buah *port Ethernet*. Pada saat praktikum, hanya port 1, 2 dan 3 saja yang digunakan. *Port* 1 digunakan sebagai jalur *internet*, *port* 2 digunakan sebagai jalur DMZ (*DemilitarizationZone*) dan *port* 3 digunakan sebagai jalur untuk *client* (*local*).

Untuk mengkonfigurasi *interface* diatas, dapat dilakukan dengan cara meng-klik *menu*□*interface*s, maka akan muncul tampilan pada gambar 6.15 :



Gambar 6.15 konfigurasi interfacerouter

Pada gambar diatas dapat dilihat terdapat 5 *interface* pada *router* mikrotik yang belum dikonfigurasi. Untuk mengkonfigurasinya dengan mengklik `ether1`, akan tampil *form* pada gambar 6.16 :

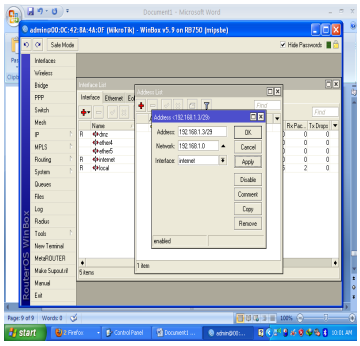


Gambar 6.16 konfigurasi interfacerouter mikrotik

Dari gambar diatas, yang dilakukan hanyalah mengganti nama `ether1` pada *textbox name* menjadi *internet*. Hal yang sama dilakukan untuk `ether2` menjadi DMZ dan `ether3` menjadi *local*.

- Konfigurasi IP *address* untuk tiap *interface*

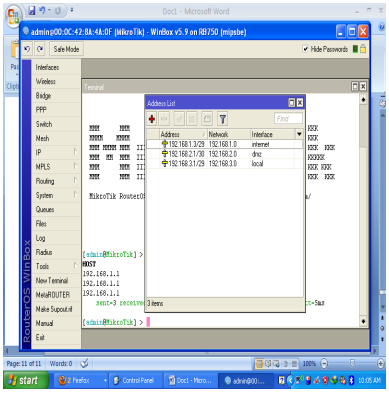
  IP *address* perlu ditentukan agar komunikasi yang diinginkan dapat terbentuk. Tiap *interface* pada *router* mikrotik perlu di konfigurasi IP *address*nya dengan cara memilih *menu* IP□Addresses, kemudian menekan tombol `add` untuk membuat konfigurasi IP *address* seperti pada gambar berikut:



Gambar 6.17 konfigurasi IP address

Gambar diatas adalah konfigurasi untuk *interface internet* pada *port Ethernet* 1. IP *address* yang digunakan adalah `192.168.1.3` dengan CIDR `29`. Untuk menerapkan konfigurasi tersebut menekan tombol *apply* dan *network* akan terisi secara otomatis.

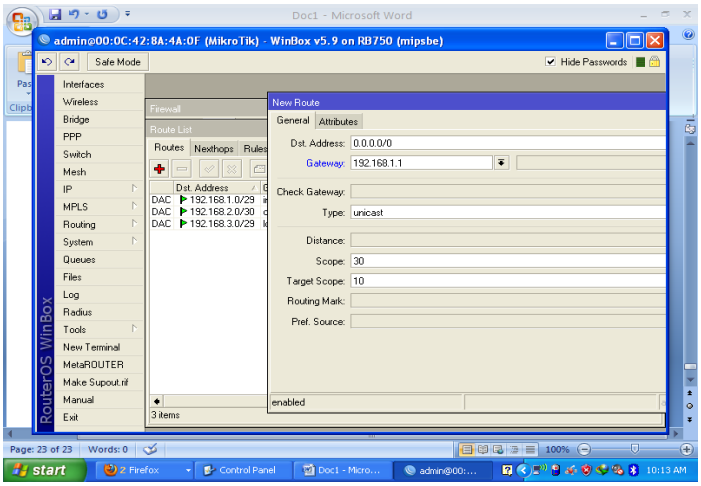
Dengan cara yang sama *interface* DMZ pada *port Ethernet* 2 menggunakan IP *address* `192.168.2.1` dengan CIDR `30`. Sedangkan *interface local* yang digunakan sebagai jalur *client* menggunakan IP *address* `192.168.3.1` dengan CIDR `29`.



Gambar 6.18 konfigurasi IP address untuk tiap interface

- Konfigurasi *routing table*

Routing table adalah daftar pengarahan paket yang melewati *router* agar bisa sampai pada tujuan. Penambahan *routingtable* dapat dilakukan dengan cara menekan tombol `add`, maka akan muncul *form* gambar 6.19 sebagai berikut :



Gambar 6.19 routingtable pada router mikrotik

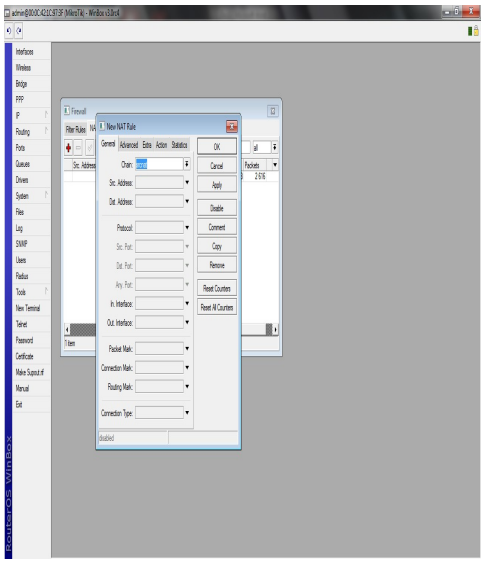
Dari gambar diatas dapat dilihat terdapat *box* untuk *destination address* atau alamat tujuan dan *gateway*. Pada praktikum kali ini *destination address* digunakan `0.0.0.0/0` yang berarti bahwa semua tujuan IP akan diarahkan ke *gateway* `192.168.1.1` bila tidak terdapat pada jaringan. Berikut adalah rute-rute yang terdapat pada routing table:

Table 6.01 Routing Table

Destination	Gateway	Keterangan
0.0.0.0/0	192.168.1.1	Internet
192.168.2.2/30	192.168.2.1	DMZ

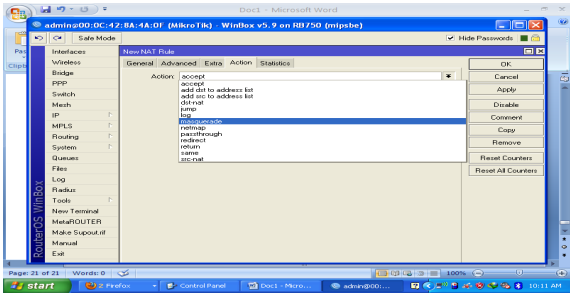
- Konfigurasi NAT

NAT atau *Network Address Translation* adalah suatu sistem yang menerjemahkan IP address pada suatu network agar dapat digunakan secara bersama oleh IP diluar network tersebut atau sebaliknya. Untuk mengkonfigurasi NAT dengan cara memilih *menu* `IP`→`firewall` kemudian tab NAT lalu menekan tombol `add` untuk menambahkan NAT *rule*. Tampilan dari NAT *rule* dapat dilihat pada gambar 6.20.



Gambar 6.20 Form NAT rule

Pada gambar diatas dapat dilihat pada tab *general*. Pada *box chain* diisi `srcnat` agar *client* dapat menggunakan IP *public* secara bersama-sama untuk terhubung dengan jaringan *internet*. Sedangkan `dstnat` berfungsi agar sebuah *host* pada jaringan *local* dapat diakses dari *internet*.



Gambar 6.21 masquerade untuk NAT jaringan

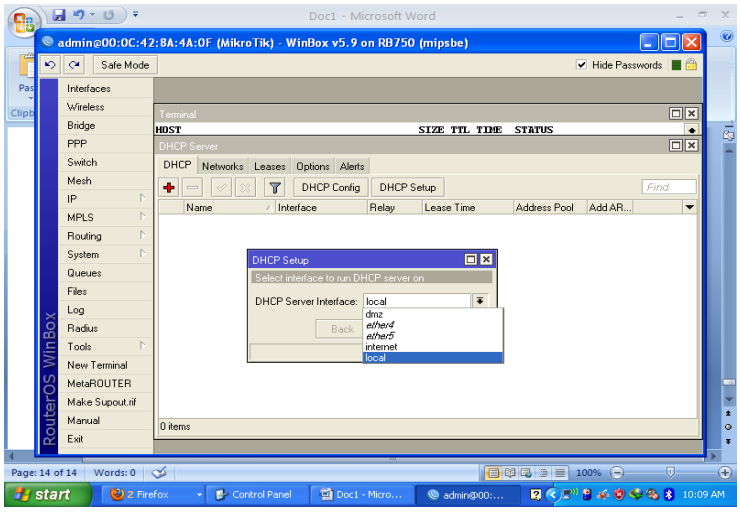
Pada tab *action* dari *form new* NAT *rule* terdapat sejumlah pilihan. Pada saat praktikum, praktikan menggunakan *action masquerede* yang artinya merubah paket-paket data IP *address* asal dan *port* dari *network* kemudian selanjutnya diteruskan ke jaringan *internet global*. pada umumnya terdapat beberapa *action* antara lain dapat dilihat pada table sebagai berikut:

Table 6.02 action Form New NAT rule

NO	ACTION	KETERANGAN
1	*Accept*	paketditerima dandilewatkanmelalui NATtanpa mengambiltindakan apapun
2	*Jump*	melompat kejalan yangditentukan olehnilai dariargumenjump target
3	*Log*	logpaketyang cocok
4	*Return*	kembali ke jalan sebelumnya, dari mana tempat lompatan terjadi
5	*Passthrough*	mengabaikan aturan ini dan pergi ke yang berikutnya
6	*Adddstaddresslist*	menambahkan alamat tujuan paket ke daftar alamat yang ditentukan
7	*Addsrcaddresslist*	menambahkan alamat paket sumber ke daftar alamat yang ditentukan
8	*dstnat and redirect*	Mengganti alamat tujuan dari paket data
9	*Srcnat and masquerade*	Mengganti alamat pengirim dari paket data
10	*Netmap*	

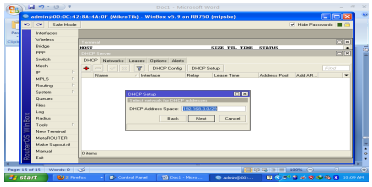
- Konfigurasi DHCP *Server*

DHCP *server* adalah *server* yang menangani pembagian IP dinamis untuk para *client* (*host*) yang terhubung dengan *router*. Untuk mengkonfigurasinya dapat dilakukan pada *menu* `IP DHCP Server` kemudian menekan tombol `add` dapat dilihat pada gambar 6.22.



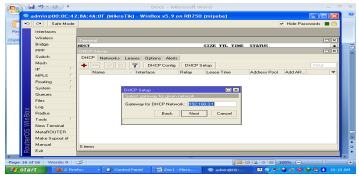
Gambar 6.22 Menu DHCP server

Pada gambar diatas dapat dilihat *form* untuk mengatur *interface* mana yang akan dipasang menjadi DHCP*server*. Terdapat 5 pilihan sesuai dengan *interface*nya yaitu `dmz`, `ether4`, `ether 5`, `internet` dan `local`. Setelah memilih *interface* kemudian klik `Next`.



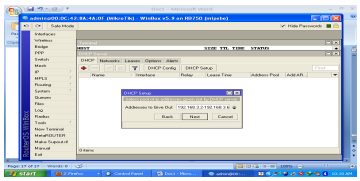
Gambar 6.23 Pemilihan network DHCP

Pada gambar diatas dapat dilihat *form* untuk mengatur *network* pada *interface* yang dipilih. Secara *default network* akan disesuaikan dengan IP *address interface* jika telah dikonfigurasi sebelumnya. Klik `next` untuk melanjutkan.



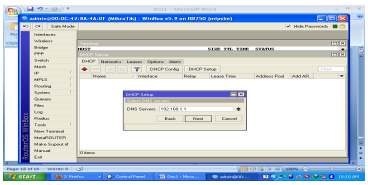
Gambar 6.24 Pengaturan gateway DHCP

Gateway yaitu jalur yang digunakan untuk berkomunikasi dengan IP *host* yang berada diluar jaringan *local*. *Gateway* `192.168.3.1` adalah IP *router* mikrotik untuk *interface local*, sehingga semua paket yang memiliki *network* diluar dari *network interface local* akan diteruskan oleh *router* mikrotik melalui dirinya.



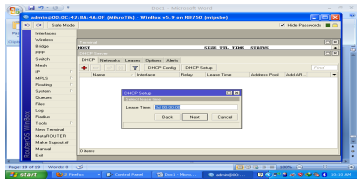
Gambar 6.25 konfigurasi range IP DHCP

Pada gambar diatas, *box* secara otomatis berisi *range* IP dari `192.168.3.2` sampai `192.168.3.7`. hal ini dikarenakan CIDR yang digunakan adalah `/29` yang berisi maksimal 8 *host*. IP `192.168.3.1` telah digunakan pada *router*, sehingga IP *valid* pada jaringan adalah `192.168.3.2 – 192.168.3.7`.



Gambar 6.26 konfigurasi DNS untuk DHCP

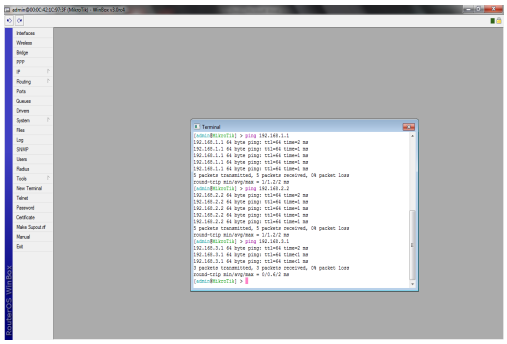
Pada gambar diatas dapat dilihat *box* untuk DNS *Server* yang akan diberikan kepada setiap *client* dari DHCP yaitu `192.168.1.1`.



Gambar 6.27 konfigurasi lease time DHCP

Lease time yaitu waktu yang dialokasikan kepada *client* untuk menggunakan IP dari DHCP *server* dan terhubung ke jaringan, pada praktikum kali ini *lease time* menggunakan *setting default* 3 hari.

Untuk mengetahui apakah *router* dapat berkomunikasi dengan setiap *host* pada *interface*nya, maka dilakukan *ping test* sebagai berikut :



Gambar 6.28 Ping test dari server

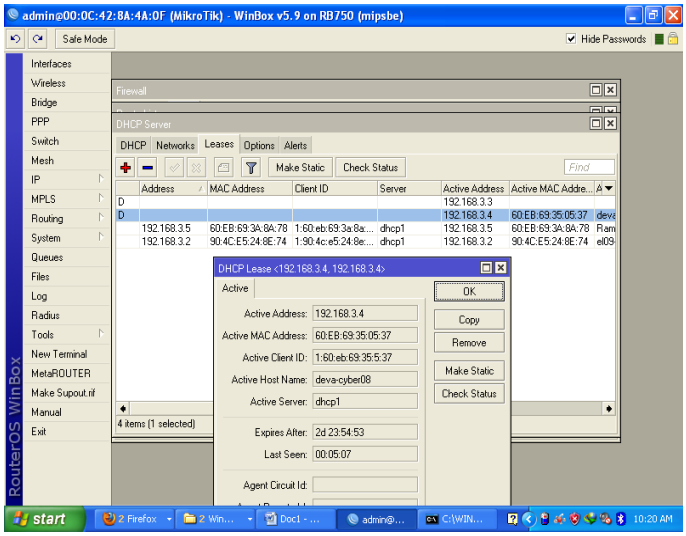
Dari gambar diatas dapat dilihat hasil *ping* dari *server* ke setiap *interface.* `192.168.1.1` adalah IP *address* untuk jalur *internet* dan `192.168.2.2` adalah IP *address* untuk jalur DMZ sedangkan `192.168.3.1` adalah IP *address* untuk jalur *client*.

2. Konfigurasi *Firewall*

2.a. Semua *client* bisa mengakses *internet* menggunakan DHCP *client* dan setiap melakukan koneksi ke mikrotik akan mendapatkan IP yang selalu sama

- Setiap *client* mendapatkan sebuah IP tetap

Setelah kofigurasi DHCP *server* selesai diterapkan, kemudian dilanjutkan membuat *static* IP untuk setiap *client* dengan cara sebagai berikut :



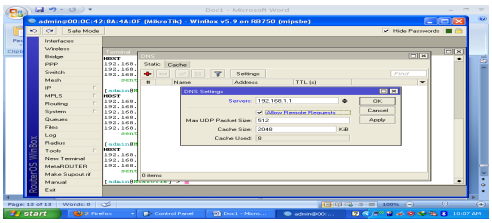
Gambar 6.29 Konfigurasi static IP

Dari gambar diatas, untuk membuat IP tetap untuk setiap *client*, dapat dilakukan dengan memilih DHCP *server* dari *menu* `IP`. Kemudian memilih tab `leases`, klik 2 kali pada komputer yang ingin dibuat mendapatkan IP *static*, lalu klik *makestatic*.

2.b. *Client* 1 bisa mengakses *internet* dan hanya bisa mengakses web pada DMZ namun tidak bisa melakukan *ping* ke DMZ.

- Konfigurasi DNS *Server* untuk DMZ

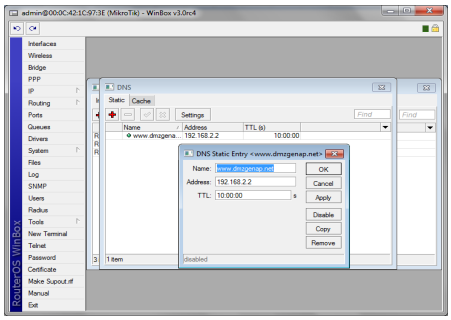
  DNS *server* diperlukan untuk memudahkan *client* mengakses DMZ cukup dengan menggunakan alamat dalam bentuk nama. Untuk mengkonfigurasinya dapat dilakukan dengan cara memilih `IP`□`DNS` kemudian meng-klik DNS *setting* untuk mengkonfigurasi DNS *server* pada *router*, kemudian meng-klik *allow remote requests* maksudnya adalah mengizinkan agar DNS server dapat diakses dari luar jaringan.



Gambar 6.30 Setting DNS Server

Pada gambar diatas, *textboxserver* dapat diisi alamat DNS *Server* yang akan dituju oleh *router* jika *client* mengakses web yang tidak terdaftar IPnya pada DNS *server* milik *router*. *Allow remote request* berarti mengizinkan *remote client* mengirimkan permintaan dari jarak jauh. kemudian klik `OK` untuk menerapkan konfigurasi.

Selanjutnya membuat DNS *Server* untuk jaringan pada *router* mikrotik dengan cara menekan tombol `add` kemudian akan muncul tampilan sebagai berikut :

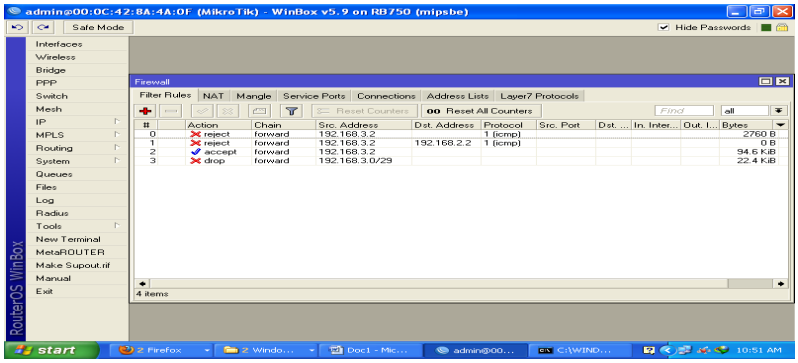


Gambar 6.31 penambahan DNS untuk IP tertentu

Pada *form* diatas, *box name* adalah nama yang diinginkan agar dapat diakses oleh *client* yaitu www.dmzgenap.net. *Box address* adalah alamat IP yang akan dituju jika *client* menghubungi www.dmzgenap.net yaitu `192.168.2.2` yang berlaku sebagai DMZ.

- Memberikan izin *client* 1 mengakses web dan membatasi *client* 1 untuk melakukan *ping* ke web DMZ

Untuk memberikan izin *client* 1 mengakses web dan membatasi *ping* ke DMZ dapat dilakukan dengan cara membuat *firewall* sebagai berikut :



Gambar 6.32 konfigurasi firewall untuk client 1

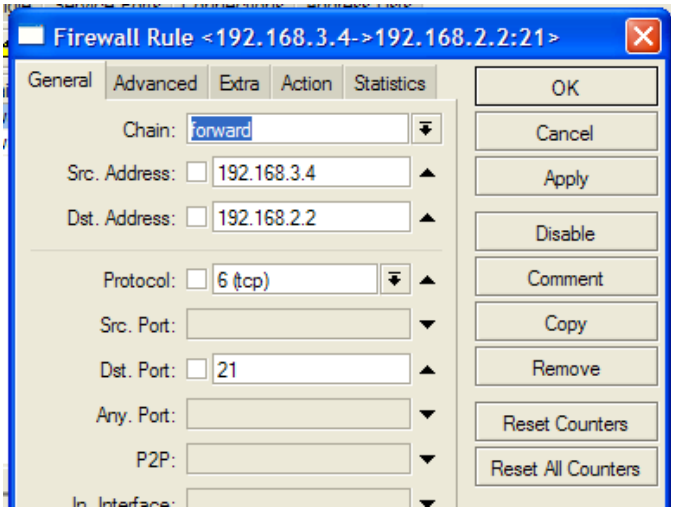
Dari gambar diatas, pada *firewall* nomor 0 berfungsi untuk mengizinkan *host* yang berada pada *network* `192.168.3.0` untuk mengakses seluruh jaringan yang ada (`0.0.0.0/0`) sehingga *client* 1 dapat mengakses *internet* dan DMZ.

Pembatasan *ping* dari *client* 1 ke DMZ terdapat pada *rule* nomor 1, yaitu semua paket dengan protocol ICMP dari *client*`192.168.3.2` *menu*ju `192.168.2.2` akan di*reject* dengan pesan *destination host unreachable*.

2.c. *Client* 2 tidak bisa melakukan *browsing* ke *internet* namun bisa mengakses FTP pada DMZ

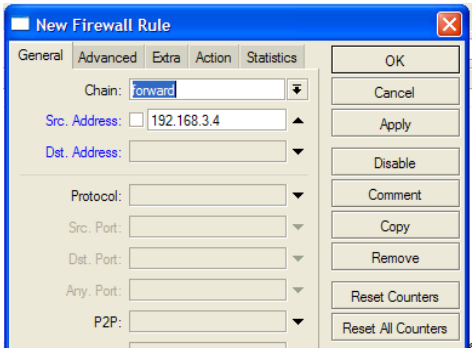
- Membatasi *client* 2 untuk melakukan *browsing* ke *internet*
- Mengizinkan *client* 2 mengakses FTP pada DMZ

Agar *client* 2 tidak bisa melakukan *browsing*, maka dibuat *rule* baru dengan *source* 192.168.3.4 (*client* 2), *destination* 192.168.2.2 (DMZ), *destination port* 21 (FTP) dengan *action* adalah *accept*.



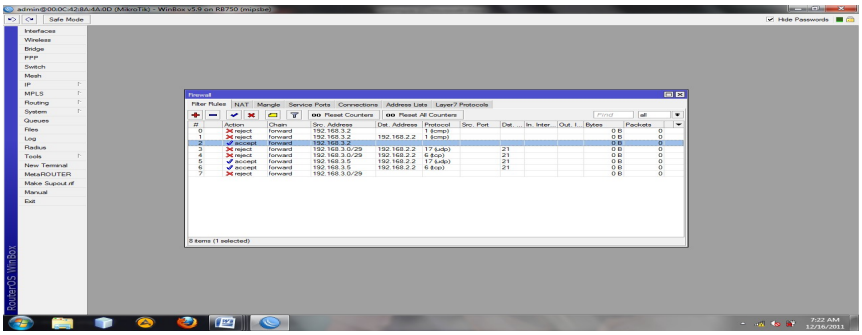
Gambar 6.33 Form Firewall Rule

Selanjutnya *rule* untuk memblok agar *client* 2 tidak bisa melakukan *browsing*, yaitu dengan *source* 192.168.3.4 (*client* 2) dan *action reject*.



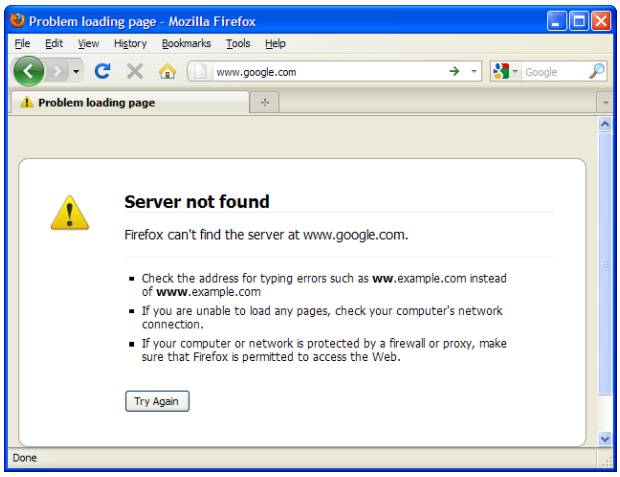
Gambar 6.34 Form Firewall Rule

Setelah semua *rule* yang diatas dimasukkan, akan terlihat *list rule* seperti gambar dibawah ini :



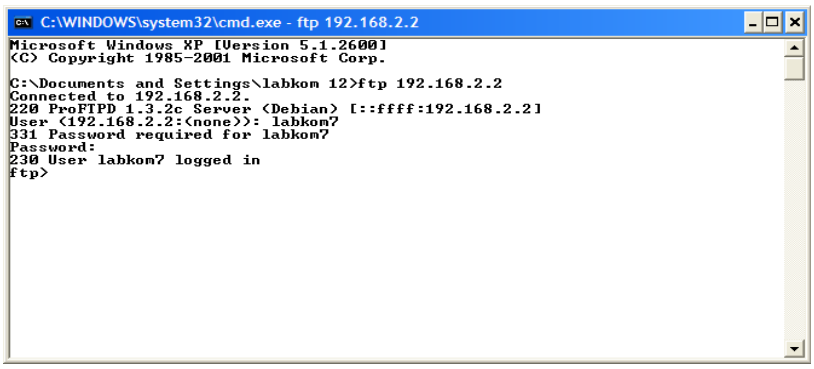
Gambar 6.35 Form Firewall Rule

Selanjutnya melakukan pengujian terhadap *rule* yang sudah dibuat, yang pertama adalah melakukan *browsing internet.*



Gambar 6.36 browsing internet

Dari gambar diatas terlihat bahwa *client* 2 tidak bisa melakukan *browsing internet* dengan adanya pemberitahuan problem *loading* page. Kemudian selanjutnya melakukan perintah FTP ke 192.168.2.2 (DMZ)

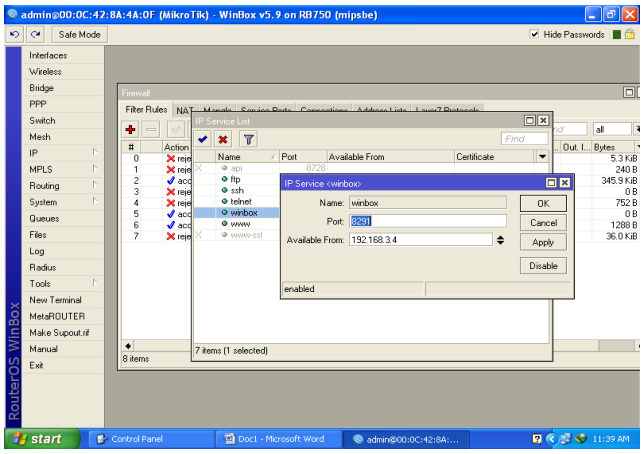


Gambar 6.37 FTP ke DMZ

Dari gambar diatas terlihat bahwa *client* 2 bisa melakukan FTP ke `192.168.2.2` (DMZ) dan sudah berhasil melakukan *login*.

2.d. *Client* 3 hanya bisa mengakses winbox pada *router* namun tidak bisa mengakses jaringan yang lain.

Untuk membatasi winbox hanya dapat diakses oleh *client* 3, dapat dilakukan dengan cara memilih `IP Services`



Gambar 6.38form IP service

Untuk mensetting pembatasan winbox, klik 2 kali pada *rule* winbox. Pembatasan hak akses dapat diisi pada *box available form* dengan IP dari *client* 3 (192.168.3.4).

E. KESIMPULAN

1. *Network Address Translation* atau yang lebih biasa disebut dengan NAT adalah suatu metode untuk menghubungkan lebih dari satu komputer ke jaringan *internet* dengan menggunakan satu alamat IP.

2. Akses mirotik yang digunakan pada praktikum ini adalah winbox. Namun ada beberapa cara lain yakni menggunakan ftp, telnet, ssh dan http.

3. Peletakan *rules* pada *firewall* mikrotik sangatlahlah penting karena dapat mempengaruhi pembatasan akses. Terkadang diperlukan aksi *enable/disable* atau pengaturan posisi *rules* agar tiap *service* itu dapat dibatasi. Misalkan untuk membatasi sebuah client dengan IP 192.168.3.5 agar dapat mengakses DMZ sedangkan client yang lain tidak bisa mengakses maka rule yang berada pada posisi teratas adalah rule untuk menerima semua paket dari client dengan IP 192.168.3.5 yang menuju ke DMZ (192.168.2.2) kemudian rule kedua adalah rule yang me-*reject* semua paket dari network 192.168.3.0.

4. Pada praktikum ini kami menggunakan modul RB750 sebagai mikrotik. Memiliki 5 buah *port ethernet* 10/100, dengan *processor* baru Atheros 400MHz.

DAFTAR PUSTAKA

Anonim.2007.*Jaringan Komputer*. Jurusan Teknik Elektro. Fakultas Teknik. Universitas Mataram.

Rahmawati, Linda.2009.*Laporan Praktikum Jaringan Komputer*.Jurusan Teknik Elektro.Fakultas Teknik.Universitas Mataram

Ramdan.2007.*Installasi Microsoft Windows 2000 Server dan Konfigurasi Pada Jaringan (clientserver).*SMK Plus Bina Teknik

Riyanto,Agus Men.2010.*Pengenalan jaringan.*LIPI.Bandung.

TUGAS

1. Carilah fungsi masing-masing opsi pada *firewall chain*

Jawab:

IP *Tables* memiliki tiga macam daftar aturan bawaan dalam table penyaringan, daftar tersebut dinamakan rantai *firewall* (*firewall chain*) atau sering disebut *chain*. Terdapat 3 buah opsi pada chain yakni *input, forward* dan *output.*

1.a.i.a) *Input* untuk paket yang disiapkan untuk soket lokal atau komputer kita sendiri. *Input* berguna untuk mengatasi paket data yang masuk.

1.a.i.b) *Forward* untuk paket yang diarahkan / routing ke box. *Forward* berguna untuk mengalihkan paket yang datang.

1.a.i.c) *Output* untuk paket yang di *generate* / dibuat sendiri. *Output* berguna untuk menghasilkan paket data yang akan diteruskan nantinya.

2. Buatlah sebuah rule untuk pembatasan akses terhadap situs dan konten tertentu.

Jawab:

Untuk melakukan pembatasan akses situs ataupun konten dapat dilakukan dengan 2 cara yakni menggunakan *firewall* ataupun *web proxy* dari mikrotik.

Dalam kasus ini, akan dibuat sebuah rule untuk membatasi konten “facebook”. Misalkan terdapat 3 buah interface, yakni internet, dmz dan local.

- Firewall

Untuk membuat rule tersebut, sebelumnya harus berhasil login ke dalam mikrotik menggunakan winbox. Kemudian menampilkan menu *firewall* dengan cara memilih menu IP*firewall*. Kemudian pada *filter rule* klik tanda + untuk menambahkan rule baru. Pada tab General setting chain dengan forward dan Out. Interface dengan internet. Dan pada tab Advance bagian content masukkan kata yang akan di blok, dalam hal ini kata facebook. Kemudian pada tab Action, pilihlah action drop kemudian

tekan tombol OK. Maka rule tersebut akan berhasil ditambahkan pada list rure firewall.

Tab	Field	Isi
General	Chain	Forward
	Out. Interface	internet
Advance	Content	facebook
Action	Action	drop

Begitu juga untuk memblok situs, langkah-langkahnya seperti di atas tetapi untuk konten diisi dengan nama situs yang akan diblok.