MIKROTIK
FUNDAMENTAL AND MEDIUM

AKROM MUSAJID

BAB I DASAR NETWORKING

Sebelum menginjak ke inti pembahasan buku ini, kita terlebih dahulu akan mengupas materi dasar jaringan sebagai bekal pemahaman bab berikutnya.

Bab ini akan memperkenalkan model komunikasi OSI, dasar TCP/IP, IP Addressing, dan Subnetting.

1.1 Model Komunikasi OSI

Model Open System Interconnection (OSI) oleh International Organization for Standardization (ISO) yang menyediakan kerangka logika terstruktur bagaimana proses komunikasi data berintaraksi melalui jaringan. Standard ini dikembangkan untuk industri komputer agar antar komputer yang berbeda vendor dan platform dapat berkomunikasi secara efisien.

Model Layer OSI

Terdapat 7 Layer pada model OSI. Setiap layer memiliki tugas tersendiri pada proses komunikasi data antar dua atau lebih perangkat jaringan. Misalnya, satu layer bertanggung jawab untuk membentuk koneksi antar perangkat, sedangangkan layer lainnya mempunyai tugas untuk mengoreksi terjadinya *error* selama proses komunikasi/transfed data berlangsung.

Gambar 1.1 *7 Layer OSI*

Model layer OSI dibagi menjadi dua group : *upper layer* dan *lower layer*. *Upper layer* khusus bertanggungjawab pada aplikasi pengguna dan bagaimana data ditunjukkan di komputer. Untuk seorang *Network Engineer* bagian utama yang harus diperhatikan adalah pada *lower layer*. *Lower layer* adalah inti dari terjadinya komunikasi data melalui jaringan.

Kata *"Open"* dalam OSI adalah untuk menyatakan model jaringan yang melakukan komunikasi tanpa memandang perangkat keras yang digunakan, selama perangkat lunak komunikasi sesuai dengan standard.

OSI Model memiliki pembagian tugas berkaitan dengan proses pengiriman informasi antar perangkat sehingga tugas dapat lebih mudah dikelola. Setiap layer memiliki fungsi dan karakteristik sendiri.

Tabel 1.1 *Fungsi dan karakteristik layer OSI*

Layer	Nama	Fungsi	Aplikasi
7	*Application*	• Menetapakan antarmuka proses user untuk mengirim data dan komunikasi dalam jaringan.	Telnet SSH FTP Mail
6	*Presentation*	• Menangani perbedaan format data di antara sistem – sistem yang tidak sama. • Mengatur encode dan decode data; encrypt dan decrypt data; compress dan decompress data.	
5	*Session*	• Melaporkan error ke layer yang lebih tinggi. • Mengatur sesi dan dialog user. • Mengontrol pembentukan link antar user	
4	*Transport*	• Mengelola pengiriman pesan *end to end* dalam jaringan. • Memberikan penghantaran paket yang reliable dengan memberikan	TCP UDP

		mekanisme recovery error dan flow control.	
3	*Network*	• Menetapkan prosedur data ditransfer diantara perangkat. • Merutekan paket mengikuti addresss unik perangkat.	IP ARP
2	*Data Link*	• Menetapkan prosedur untuk operasi link komunikasi. • Menyusun frame untuk paket. • Mendeteksi dan mengoreksi *error* transmisi paket	Ethernet ARP
1	*Physical*	• Merubah data menjadi besaran sinyal. • Memberikan interface di antara media dan device jaringan. • Melakukan komunikasi *peer to peer*	Ethernet

1.2 MODEL TCP/IP

TCP/IP adalah protokol internet yang paling banyak digunakan saat ini. TCP/IP (Transmission Control Protcol/Internet Protocol) memiliki beberapa keunggulan, antara lain :

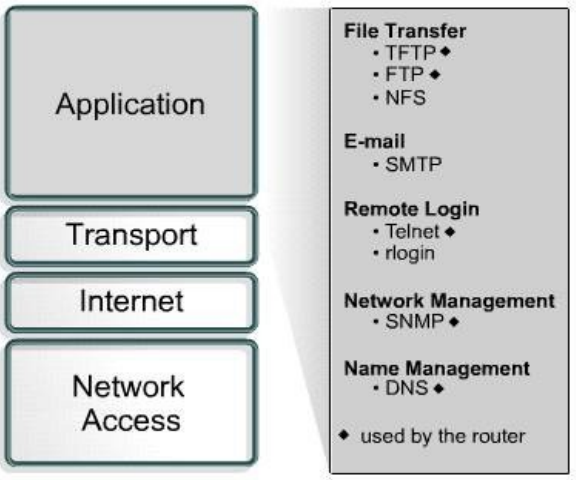
1. *Open Protocol Standard,* yaitu tersedia secara bebas dan dikembangkan independen terhadap komputer hardware ataupun sistem operasi apapun. Karena didukung secara meluas di dunia komunikasi, TCP/IP sangat ideal untuk menyatukan bermacam hardware dan software, walaupun tidak berkomunikasi lewat internet bisa pada jaringan lokal.

2. Independen dari physical network hardware, ini menyebabkan TCP/IP dapat mengitegrasikan bermaca, network, baik melalui ethernet, token ring, dial-up, X.25/AX.25 dan media transmisi fisik lainnya.

3. Skema pengalamatan yang umum menyebabkan device yang menggunakan TCP/IP dapat menghubungi alamat device–device lain di seluruh network, bahkan internet sekalipun.

4. High level protocol standard, yang dapat melayani user secara luas.

Model TCP/IP



Gambar 1.2 *Model TCP/IP*

TCP/IP didefinisikan sebagai koleksi (suit) protokol jaringan yang berperan dalam membangun environment jaringan global seperti Internet. Nama TCP/IP diambil dari dua 'keluarga' prtocol fundamental, yaitu TCP dan IP. Meskipun demikian suit masih memiliki protokol utama lainnya, seperti UDP dan ICMP. Protokol bekerja sama dalam memberikan framework networking yang digunakan oleh banyak protokol aplikasi berbeda, dimana masing-masing digunakan untuk tujuan berbeda.

1.2.1 TCP

TCP (Transmission Control Protocol) merupakan protocol transport yang populer saat ini. Berbeda dengan UDP dan IP yang tergolong *"connectionless",* TCP dikenal dengan protokol *"connection oriented",* artinya protokol membentuk koneksi terlebih dahulu untuk mengirim pesan sampai terjadi proses pertukaran antar aplikasi.

TCP juga bekerjasama dengan Internet Protocol (IP) untuk mengirimkan data antar perangkat jaringan melewati jaringan atau Internet. Data berbentuk unit pesan *(packet)*. Jika IP menangani pengiriman data, maka TCP bertugas mengawasi atau menjaga jalur data paket. Sebuah data akan dipecah menjadi beberapa bagian paket untuk efisiensi routing. Ketika data yang dikirim hilang selama transmisi, TCP dapat mentransmisikan ulang hingga kondisis *timed out* atau pengiriman sukses diterima.

1.2.2 IP

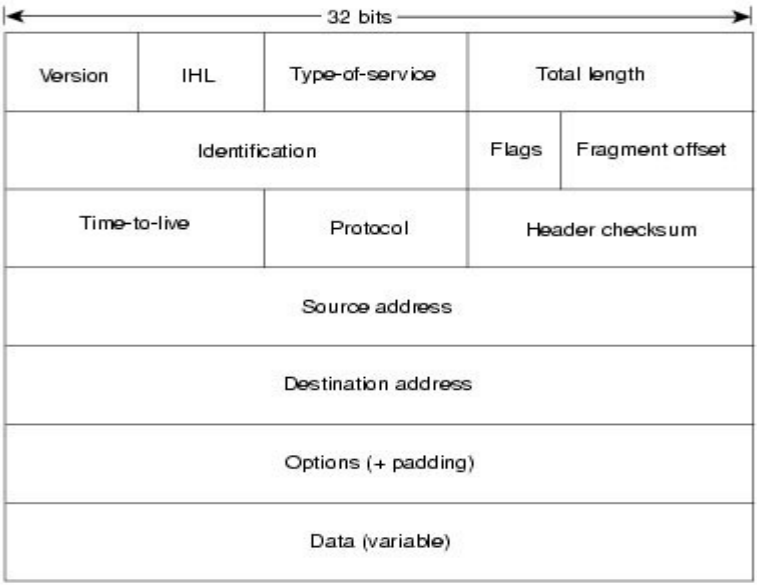
IP (Internet Protocol) merupakan metode yang digunakan unuk mengirim data dari satu perangkat jaringan ke perangkat jaringan lainnya. Setiap perangkat jaringan (host) memiliki paling minimal satu IP address yang berfungsi untuk memperkenalkan dirinya ke host lain di Internet.

IP mempunyai tanggung jawab sebagai :

1. Memberikan layan connectionless atas pengiriman data melalui internetwork.

2. Memberikan fragmantasi dan reassembly datagram untuk mendukung link daa dengan ukuran Maximum Transmission Unit (MTU) berbeda – beda.

Jika diilustrasikan, sebuah paket IP dapat digambarkan sebagai berikut :



Gambar 1.3 *Paket IP*

Tabel 1.2 *Field dan deskripsinya*

Field	Deskripsi
Version	Mengindikasikan versi IP yang dipakai.
IP Header Length (IHL)	Mengindikasikan panjang header datagram (dalam 32 bit).
Type of Service	Menetapkan bagaimana protokol upper layer menangani datagram dan menugaskannya berdasarkan pada tingkatan pada tingkatan terpenting.

Total Length	Menetapkan panjang keseluruhan paket IP (dalam byte), termasuk data dan headernya.
Identification	Memuat sebuah bilangan yang mengidentifikasikan datagram saat ini. Field diuanakan untuk membantu menyatukan fragmen datagram.
Flag	Tediri atas file 3 bit yang mengontrol fragmentasi.
Fragment Offset	Mengindikasikan posisi data fragmen yang relatif terhadap pemulaan data dalam datagram orisinal dan memungkinkan proses IP tujuan tepat mengontruksi ulang datagram orisinal.
Time to Live	Merawat nilai hitungan *(counter)* yang berangsur – angsur berkurang hingga mencapai nol, dimana datagram akan dibuang. *Time to Live* menjaga paket dari pengulangan terus menerus.
Protocol	Mengindikasikan protokol *upper layer* yang akan menerima paket setelah proses IP selesai.
Header Cheksum	Membantu meyakinkan integritas IP header.
Source Address	Menentukan node pengirim.

Destination Address	Menentukan node penerima.
Options	Memungkinkan IP pendukung beragam opsi lainnya seperti keamanan.
Data	Memuat informasi *upper layer*.

IP merupakan connectionless, yang berarti tidak ada kesepakatan koneksi terlebih dahulu antara dua perangkat yang melakukan komunikasi. Setiap paket yang melintasi Internet diperlakukan sebagai unit data independen, tanpa ada keterkaitan dengan unit data lainnya.

1.3 IP ADDRESS

IP Address adalah deretan angka biner antar 32-bit sampai 128-bit yang digunakan sebagai alamat id untuk setiap komputer dalam jaringan. Sistem pengalamatan IP terbagi menjadi dua, yaitu :

a. IP versi 4 (IPv4)

b. IP versi 6 (IPv6)

1.3.1. Pembagian Kelas IP Address

Kumpulan komputer dalam satu jaringan TCP/IP dikelompokkan ke dalam kelas.

a. Apabila tiga dari tiga blok terahir berubah, termasuk kelas A :

   xxx.aaa.bbb.ccc.

b. Apabila dua dari tiga blok terahir berubah, termasuk kelas B : xxx.xxx.aaa.bbb.

c. Apabila blok terahir yang berubah, maka termasuk kelas C : xxx.xxx.xxx.aaa.

d. Kelas D dan E akan dijelaskan kemudian.

IP Address kelas A

0	Network ID	Alamat Host
Biner	7 Bit	24 Bit

Ketentuan kelas A :

0.0.0.0	Tidak boleh digunakan
1.0.0.0 s/d 126.0.0.0	Network ID yang tersedia dan boleh digunakan
127.0.0.0	Tidak boleh digunakan karena dialokasikan untuk keperluan loopback

Contoh IP kelas A :

10.0.0.1 = 00001010.00000000.00000000.00000001

IP Address Kelas B

10	Network ID	Alamat Host
Biner	14 Bit	16 Bit

Ketentuan kelas B :

128.0.0.0 s/d 192.254.0.0	Network ID yang boleh digunakan
192.255.0.0	Tidak boleh dipakai

Contoh IP kelas B :

172.16.0.1 = 10101100.00010000.00000000.00000001

IP Address Kelas C

110	Network ID	Alamat Host
Biner	21 Bit	8 Bit

Ketentuan kelas C :

192.0.0.0	Tidak boleh digunakan
192.0.1.0 s/d 223.255.254.0	Network ID yang boleh digunakan

223.255.255.0	Tidak boleh digunakan

Contoh IP kelas C

192.168.0.1 = 11000000.10101000.00000000.00000001

IP Address Kelas D

1110	Multicast
Biner	28 Bit

Ketentuan kelas D :

224.0.0.0 s/d 239.255.255.255	Kelompok multicast

Alamat IP kelas D semuanya digunakan untuk multicasting dan selalu diawali dengan bit 1110.

IP Address Kelas E

1111	Network ID
Biner	24 Bit

Ketentuan kelas E :

224.0.0.0	Tidak boleh digunakan

255.255.255.255	IP broadcast

IP address kelas E ditandai dengen nilai biner 1111 pada bagian awal alamat yang sebenarnya tidak boleh digunakan oleh host, IP ini digunakan sebagai media sesearch teknologi masa depan.

1.3.2 Subnet Mask

Subnetmask biasanya digunakan untuk menentukan bagian mana yang merupakan alamat jaringan dan bagian mana yang merupakan alamat host. Subnetmask terdiri dari 32 bit seperti IP address yang juga ditulis dalam notasi desimal bertitik. Untuk menentukan network ID biasanya digunakan proses AND dimana bit-bit subnet mask di-AND terhadap bit-bit IP address yang ada.

Contoh :

IP address : 180.20.5.9

Subnet mask : 255.255.0.0

Network ID

180.20.5.9 : 10110100.00010100.00000101.00001001

255.255.0.0 : 11111111.11111111.00000000.00000000

: 10110100.00010100.00000000.00000000

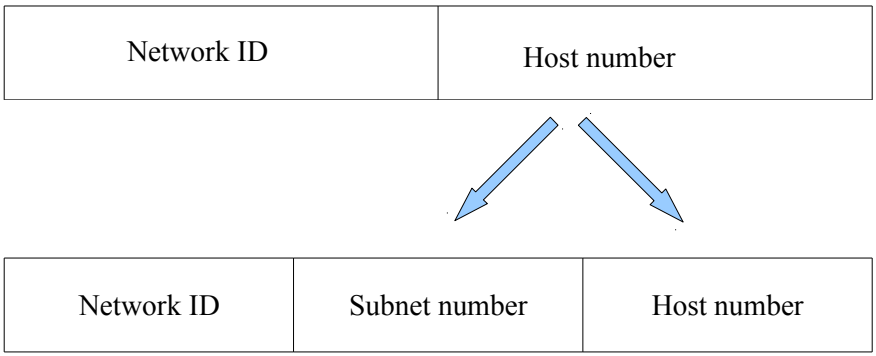
: 180.20.0.0

1.3.3 Subnetting

Subnetting adalah cara membagi satu jaringan menjadi beberapa sub jaringan. Beberapa bit dari bagian host ID dialokasikan menjadi bit tambahan pada bagian network ID. Cara ini menciptakan sejumlah Network ID tambahan dan mengurangi jumlah maksimum host yang ada dalam tiap jaringan tersebut.

Untuk gambaran dari seubnetting dapat diterangkan melalui gambar dibawah ini.

Two-level classful hierarchy



Gambar 1.4 Hirarki Subnet Address

Jumlah bit yang dipindahkan ini dapat bervariasi yang ditentukan oleh nilai subnet mask. Sebagai contoh, network ID kelas B yaitu 172.16.0.0, subnetting dapat dilakukan dengan cara sebagai berikut.

Address kelas B (sebelum subnetting)

Network ID	Network ID	Host ID	Host ID
172	16	0	0

Address kelas B (setelah subnetting

Network ID	Network ID	Host ID	Host ID
172	16	2	0

Gambar 1.5 Bit-bit yang dipindahkan dari Host ID yang membuat alamat subnet

Beberapa alasan membangun subnetting adalah sebagai berikut :

Mereduksi trafik jaringan

Alasan utama menggunakan subnetting yaitu untuk mereduksi ukuran broadcast domain.

a. Mengoptimasi performasi jaringan

Sebagai hasil dari reduksi jaringan, maka optimasi akan diperoleh performasi jaringan yang lebih baik.

b. Memudahkan manajemen

Dengan membagi-bagi jaringan yang diharapkan akan memudahkan administrator dalam mengatur jaringan terutama untuk keperluan identifikasi.

c. Mengefektifkan jaringan yang dibatasi area geografis yang luas.

Sebuah jaringan tunggal dan besar yang dibatasi oleh area geografis yang luas dapat menimbulkan berbagai masalah, terutama dari sisi kecepatan. Dengan mengkoreksikan multi jaringan yang lebih kecil maka diharapkan dapat membuat sistem lebih efisien.

Hal yang harus diketahui untuk melakukan subnetting adalah mengingat nila dari bit-bit subnet mask. Nilai iniakrom@akrom-Satellite-L645:~$ yang akan dijadikan panduan dalam proses subnetting. Perhatikan tabel dibawah ini.

Tabel 1.3 Bit-bit subnet mask

128	**64**	**32**	**16**	**8**	**4**	**2**	**1**		
2^7	**2^6**	**2^5**	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**		
1	0	0	0	0	0	0	0	=	128
1	1	0	0	0	0	0	0	=	192
1	1	1	0	0	0	0	0	=	224
1	1	1	1	0	0	0	0	=	240
1	1	1	1	1	0	0	0	=	248
1	1	1	1	1	1	0	0	=	252

1	1	1	1	1	1	1	0	=	254
1	1	1	1	1	1	1	1	=	255

Berdasarkan tabel diatas nilai subnet mask yang diguanakan untuk subnetting adalah 128, 192, 224, 240, 240, 248, 252, 254, dan 255.

Tabel 1.4 Nilai subnet mask yang mungkin untuk subnetting

Subnet mask	**CIDR**	**Subnet mask**	**CIDR**
255.128.0.0	/9	255.255.240.0	/20
255.192.0.0	/10	255.255.248.0	/21
255.224.0.0	/11	255.255.252.0	/22
255.240.0.0	/12	255.255.254.0	/23
255.248.0.0	/13	255.255.255.0	/24
255.252.0.0	/14	255.255.255.128	/25
255.254.0.0	/15	255.255.255.192	/26

255.255.0.0	/16	255.255.255.224	/27
255.255.128.0	/17	255.255.255.240	/28
255.255.192.0	/18	255.255.255.248	/29
255.255.224.0	/19	255.255.255.252	/30

Contoh subnetting kelas C

Apabila sebuah network ID 192.168.10.0/30, maka untuk menentukan kelas dan subnet mask dari network ID adalah sebagai berikut :

IP 192.168.10.0 termsuk IP dari kelas C. Subnet mask /30 berarti 11111111.11111111.11111111.**111111**00

(128 + 64 + 32 + 16 +8 +4 = 252)

Sehingga subnet mask adalah 255.255.255.252.

Perhitungan tentang subnetting akan terfokus pada 4 hal, jumlah subnet, jumlah host per subnet, blok subnet, alamat host dan broadcast yang valid.

a. Jumlah subnet = 2^x , dimana x adalah banyaknya bit 1 pada

oktet terahir subnet mask (2 oktet terahir untuk kelas B dan 3 oktet terahir untuk kelas A). Jadi 2^6 = 64 subnet.

b. Jumlah host per subnet = 2^y - 2, dimana y adalah banyaknya bit 0 pada oktet terahir subnet. Jadi jumlah host per subnet adalah 2^2 - 2 = 2 host.

c. Blok subnet = 256 – 252 (nilai oktet terahir subnet mask) = 4. jadi blok subnet lengkapnya adalah 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, ..., 252.

d. Alamat host dan broadcast yang valid dapat dilihat pada tabel di bawah ini. Sebagai catatan, host pertama adalah 1 angka setelah subnet dan broadcast adalah 1 angka sebelum subnet berikutnya.

Tabel 1.5 Tabel hasil subnetting 192.168.10.0/30

Network ID	192.168.10.0	192.168.10.4	.....	192.168.10.252
Host Pertama	192.168.10.1	192.168.10.5	.....	192.168.10.253
Host Terakhir	192.168.10.2	192.168.10.6	.....	192.168.10.254
Broadcast	192.168.10.3	192.168.10.7	.....	192.168.10.255

Dengan konsep dan teknik yang sama, subnet mask yang bisa digunakan untuk kelas C adalah sebagai berikut.

Tabel 1.6 Subnet mask yang dapat digunakan untu subnetting kelas C

Subnet Mask	CIDR
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30

Contoh subnetting kelas B

Subnet mask yang bisa digunakan untuk subnetting kelas B seperti pada tabel bawah ini.

Tabel 1.7 Subnet mask yang digunakan subnetting kelas B

Subnet Mask	CIDR	Subnet Mask	CIDR
255.255.128.0	/17	255.255.255.128	/25

255.255.192.0	/18	255.255.255.192	/26
255.255.224.0	/19	255.255.255.224	/27
255.255.240.0	/20	255.255.255.240	/28
255.255.248.0	/21	255.255.255.248	/29
255.255.252.0	/22	255.255.255.252	/30
255.255.254.0	/23		
255.255.255.0	/24		

Contoh subnetting kelas B adalah sebagai berikut. Apabila alamat jaringan 172.16.0.0/18, maka subnetting dapat dilakukan sebagai berikut.

IP 172.16.0.0/18 merupakan IP kelas B, subnet mask /18 berarti :

11111111.11111111.11000000.00000000

(128 + 64 = 192 (Oktet ke 3))

Sehingga subnet mask adalah 255.255.192.0.

Perhitungan :

a. Jumlah subnet = 2^x , dimana x adalah banyak bit 1 pada oktet

2 terakhir. Jadi jumlah subnet adalah 2^2 = 4 subnet.

b. Jumlah host per subnet adalah 2^y - 2, dimana y adalah banyaknya bit 0 pada 2 oktet terakhir. Jadi jumlah host per subnet adalah 2^{14} - 2 = 16.382 host.

c. Blok subnet 256 – 192 = 64. Subnet lengkapnya adalah 0, 64, 128 dan 192.

d. Alamat host dan broadcast yang valid seperti tabel di bawah ini.

Tabel 1.8 Tabel hasil subnetting 172.16.0.0/18

Subnet	172.16.0.0	.... .	172.16.192.0
Host Pertama	172.16.0.1	.... .	172.16.192.1
Host Terakhir	172.16.63.254	.... .	172.16.255.254
Broadcast	172.16.63.255	.... .	172.16.255.254

Contoh subnetting kelas A

Konsep subnetting kelas A sama dengan kelas B dan C, hanya berbeda oktet mana pada blok subnet yang akan dimainkan. Kalai kelas C di oktet 4, kelas B di oktet 3 dan 4 (2 oktet terakhir), kalai A di oktet 2, 3 dan 4 (3 oktet terakhir). Kemudian subnet mask yangbisa digunakan untuk subnetting kelas A adalah semua subnet mask dari

CIDR /8 sampai /30.

Contoh alamat jaringan 10.0.0.0/10, maka dapat ditentukan IP 10.0.0.0 tergolong IP kelas A. Subnet mask /10 adalah 11111111.11000000.00000000.00000000 (255.192.0.0)

Perhitungan :

a. Jumlah subnet 2^2 = 4 subnet.

b. Jumlah hostper subnet 2^{22} - 2 = 4.194.302 host.

c. Blok subnet 256 – 192 = 64, jadi subnet lengkapnya adalah 0, 64, 128, 192.

d. Alamat host dan broadcast yang valid seperti tabel di bawah ini.

Tabel 1.9 Tabel hasil subnetting 10.0.0.0/10

Subnet	10.0.0.0	.....	10.192.0.0
Host Pertama	10.0.0.1	.....	10.192.0.1
Host Terakhir	10.63.255.254	.....	10.255.255.254
Broadcast	10.63.255.255	.....	10.255.255.255

BAB II Mikrotik RouterOS

RouterOS

Mikrotik RouterOS adalah salah satu distro Linux yang didesain khusus untuk fungsi *routing system*. Perusahaan berkantor pusat di Latvia ini diprakarsai oleh John Trully dan Arnis Reaktins. Mikrotik mengeluarkan produk berupa *RouterBoard* yang berbentik *hardware router* dan berupa *RouterOS* yang dapat diinstall di sebuah PC.

Kehandalan Mikrotik sudah terbukti dan tidak diragukan lagi, baik dari segi keamanan maupun kemudahan penggunannya. Karena itulah hampir semua ISP (Internet Service Provideri) mengenal dan menggunakan Mikroik untuk layanan ke pelanggan maupun dalam manajamen network.

Mikrotik RouterOS memiliki berbagai fitur jaringan, adapaun fitur dari Mikrotik RouterOS itu sendiri adalah :

1. *Routing*

   Static routing, policy routing, ECMP, RIP, OSPF, BGP

2. *Firewall*

   Mangle, filter, layer 7 filtering, address list, NAT

3. *Quality of Service*

   Simple queue, HTB, PFIFO, BFIFO, PCQ, SFQ, RED

4. *Wireless Network*

PTP, PTMP, *nstream, dual nstrea,* WDS

5. IP *Tunnel*

   PPTP, IPIP, Ipsec, EoIP, L2TP, MPLS, OpenVPN

6. *Authentication*

   PPPoE, *Hotspot, Radipengenalan mikrotikus*

7. *Interface*

   Gigabit ethernet, wireless, V35, G703, ISDN, *dial up, bridge, bonding,* STP, RSTP

8. *Service*

   DHCP *server,* IP *poll, web proxy,* DNS *cache*

Lisensi MikroTik

Sebelum melakukan instalasi hal yang perlu diperhatikan adalah level lisenis, perhatikan manual lisensi atau daftar list harga *software.* Level tertinggi adalah level 6 yang memiliki semua modul yang bisa digunakan secara maksimum. Perbedaan dari tiap lisensi adalah pada harga dan kelengkapan paket. Sekarang Mikrotik menerapkan sistem level lisensi yang baru. Dengan adanya sistem level lisensi yang baru ini, diharapkan pengguna lebih diuntungkan, karena harganya yang lebih murah dan jangka waktu *free upgrade* yang lebih lama (sekarang menjadi 3 tahun untuk level 5 dan 6). Untuk lebih jelasnya lihat tabel lisensi Mikortik pada tabel 2.1.

Tabel 1.1 Lisensi Mikrotik

Level number	4 (WISP AP)	5 (WISP AP 3Y)	6 (Controller 3Y)
Software only	$45 (SW/L4)	$95 (SW/L5)	$250 (SW/L6)
Installed on IDE Flash	$85 (SW/FL4)	$135 (SW/FL5)	$290 (SW/FL6)
Features			
Upgrade time	1y	3y	3y
Initial Config Support	15d	30d	30d
Wireless Client and Bridge	yes	yes	yes
Wireless AP	**yes**	yes	yes
Synchronous interfaces	yes	yes	yes
EoIP tunnels	unlimited	unlimited	unlimited
PPPoE tunnels	200	500	unlimited
PPTP tunnels	200	unlimited	unlimited
L2TP tunnels	200	unlimited	unlimited
VLAN interfaces	unlimited	unlimited	unlimited
P2P firewall rules	unlimited	unlimited	unlimited
NAT rules	unlimited	unlimited	unlimited
HotSpot active users	200	500	unlimited
RADIUS client	yes	yes	yes
Queues	unlimited	unlimited	unlimited
Web proxy	yes	yes	yes
RIP, OSPF, BGP protocols	yes (2.10 = no)	yes	yes
Upgrade	For one year or license purchase (configuration saved)	For one year or license purchase (configuration saved)	For one year or license purchase (configuration saved)

Instalasi Mikrotik RouterOS

Ada 2 macam cara instalasi yang sering digunakan antara lain :

1. ISO *image*, yaitu menggunakan CD instalasi. Download terlebih dahulu file mikortik berekstensi .iso kemudian *burn* ke dalam CD kosong.

2. *NetInstall*, melalui jaringan LAN menggunakan *ethernet* yang mendukung proses *booting* komputer melalui *ethernet card.*

Sedangkan untuk instalasi di dalam PC syarat minimal sebuah komputer untuk dapat menjalanka Mikrotik adalah :

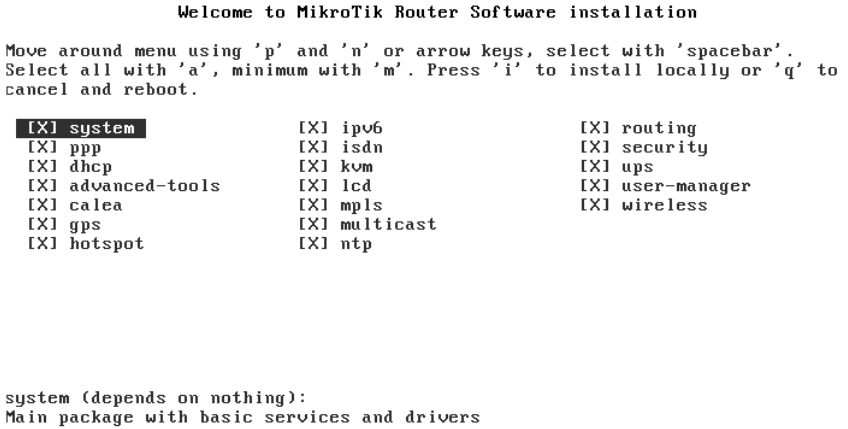
1. Menggunakan prosesor setidaknya 100 MHz atau lebih seperti Intel Pentium, Cyrix 6X86, AMD K5 atau prosesor yang lebih baru dari Intel IA-32 (i386), untuk penggunaan lebih dari satu prosesor belum diperbolehkan.

2. Memori (RAM) minimal 64 MB dan maksimal 1 GB.

3. Media penyimapanan (*Hardisk*) menggunakan sistem standar *controller* IDE dan ATA. Penggunaan SATA, SCSI dan USB tidak didukung. Minimal sisa media penyimpanan adalah 64 MB.

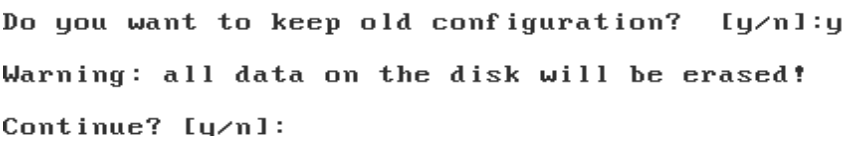
Install menggunakan file ISO Image

Untuk instalasi menggunakan ISO *image* pada PC setting boot melalui CD-ROM terlebih dahulu. Pada saat setelah booting akan muncul proses awal install mikrotik tekan tombol 'a' untuk memilih semua paket untuk diinstall seperti gambar 2.1.



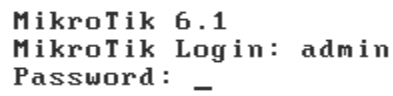
Gambar 2.1 Install paket mikrotik

Untuk melanjutkan proses ke tahap install tekan tombol 'i', kemudian akan muncul beberapa konfigurasmi tekan tombol 'y' untuk melanjutkan proses seperti gambar 2.2 dibawah ini.



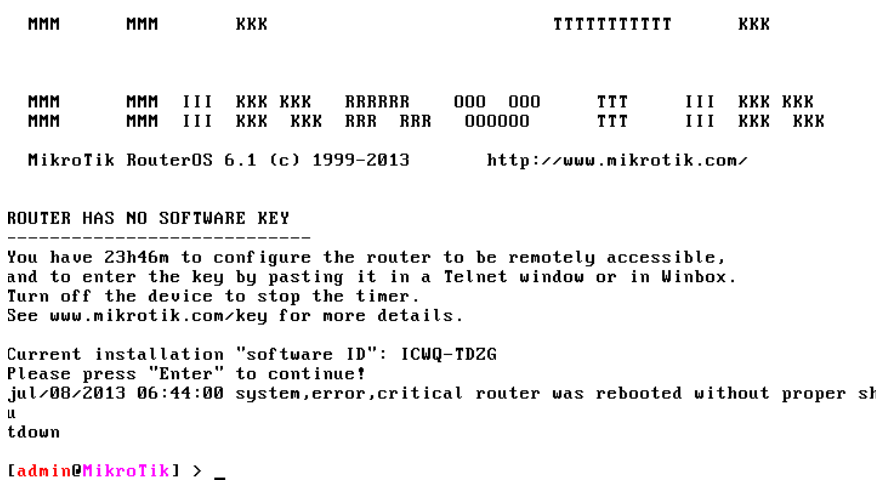
Gambar 2.2 Konfirmasi Install

Setelah proses install selesai lepaskan CD dari komputer kemudian lakukan restart, akan muncul halaman *login* Mikrotik isikan pada user *login* dengan nama 'admin' kemudian *password* dikosongkan seperti pada gambar 2.3 dibawah ini.



Gambar 2.3 Halaman login Mikrotik

Setelah *login* berhasil barulah masuk ke halaman terminal Mikrotik dimana segala konfigurasi dilakukan dengan *text*, lihat gambar 1.4



Gambar 1.4 *Terminal* Mikrotik

Install Via NetInstall

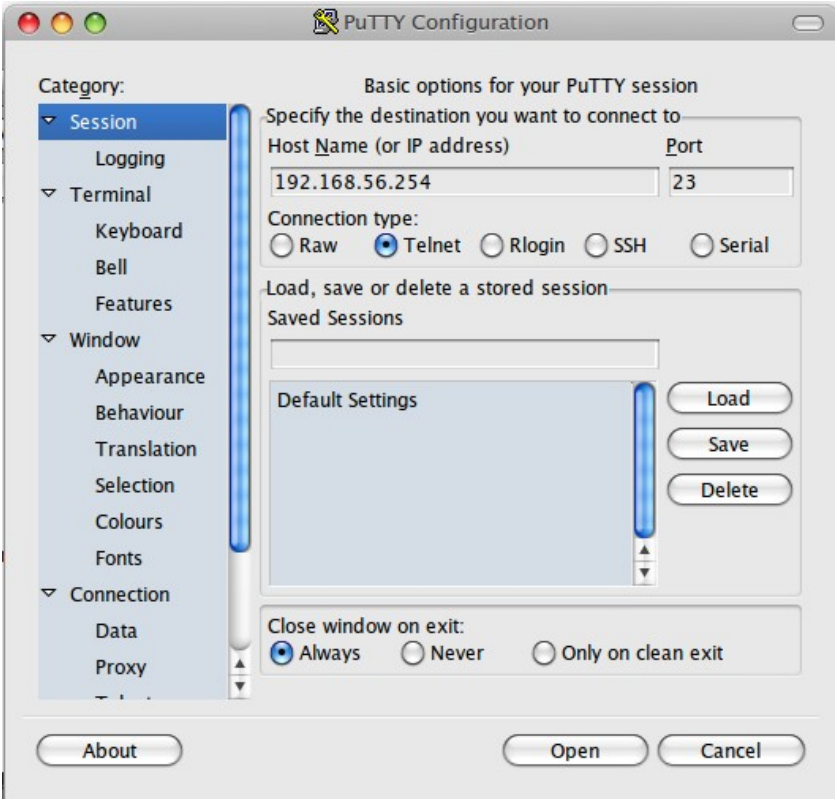
Install melalui NetInstall akan dibahas di lain Bab.

Akses Mikrotik

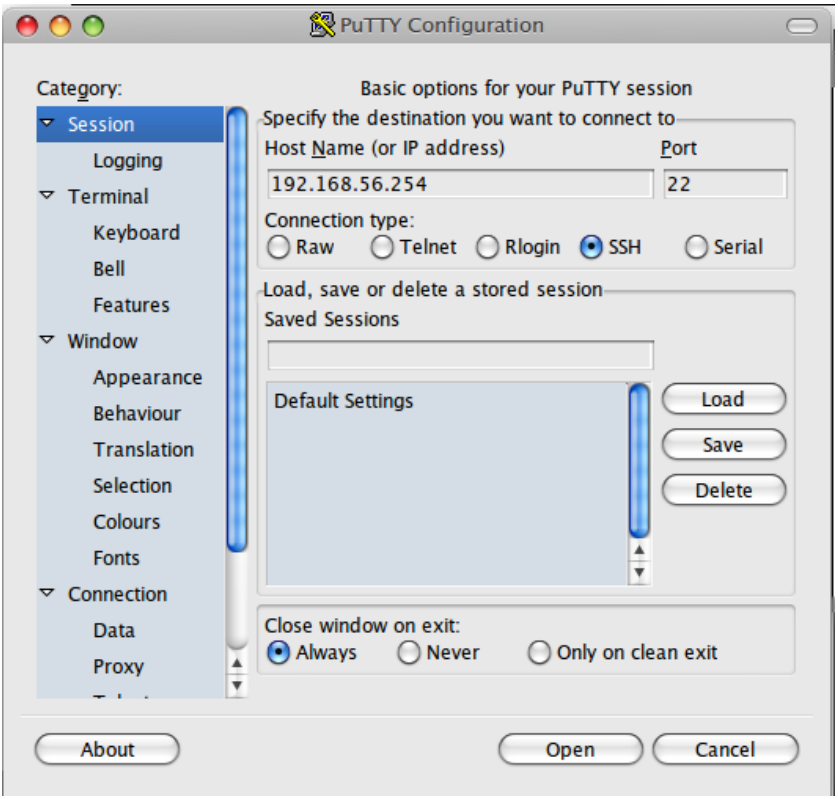
Untuk mengakses Mikrotik ada 3 cara yang dapat dilakukan antara lain :

1. Via *console* Mikrotik

   Ada dua cara apabila menggunakan *console* untuk mengakses Mikrotik yaitu menggunakan *telnet* atau *ssh* dengan menyertakan IP *address* Mikrotik, gunakan *software* Putty untuk melakukan *remote* Mikrotik. Lihat gambar 1.5 dan gambar 1.6



Gambar 1.5 Telnet

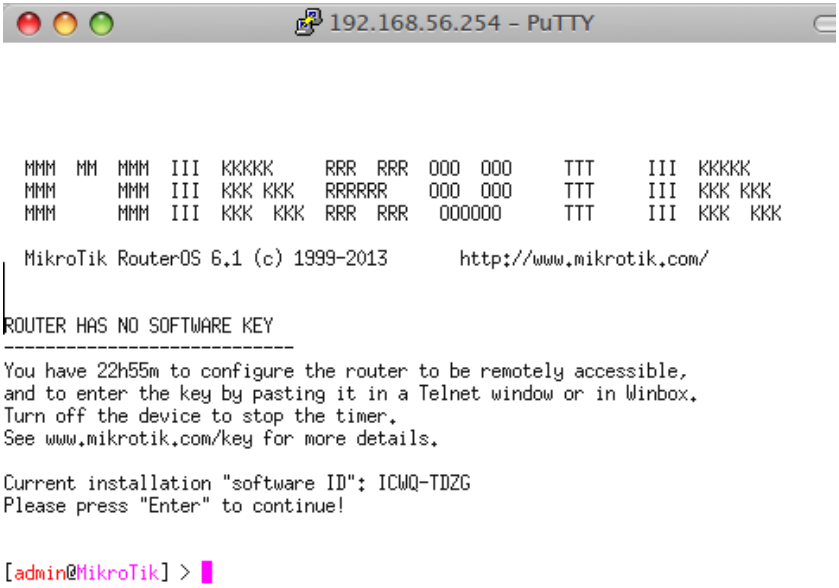


Gambar 1.6 SSH

Dianjurkan menggunakan tipe *remote* SSH karena lebih aman karena terenkripsi.

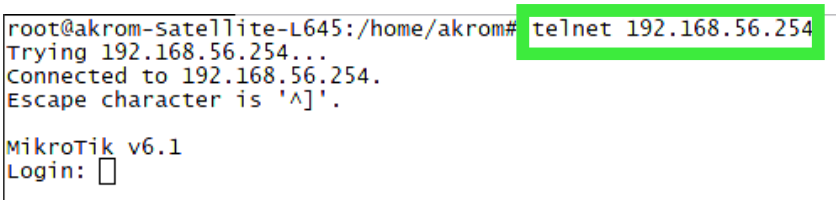
Setelah login akan muncul halaman login seperti gambar 1.7 dan gambar 1.8, masukan *user* dan *password* Mikrotik.

Gambar 1.7 *Login* Putty



Gambar 1.8 *Terminal* Putty

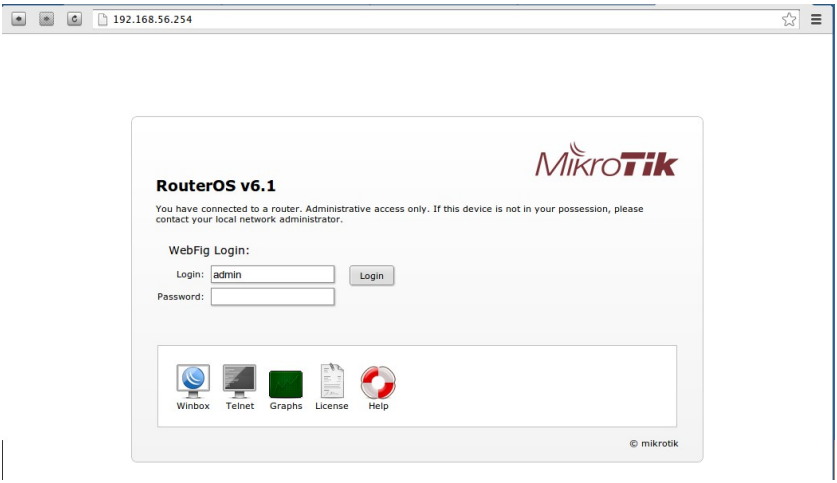
Tanpa menggunakan Putty pun dapat digunakan *terminal/commandpromt* yang terdapat pada komputer untuk melakukan akses Mikrotik dengan mengetikkan perintah seperti pada gambar 1.9.



Gambar 1.9 telnet *terminal/commandprompt*

2. Via web browser

Mikrotik juga dapat diakses melalui web*/port* 80 pada browser. Caranya cukup mudah dengan mengetikkan IP *address* Mikrotik pada kolom *address browser.*

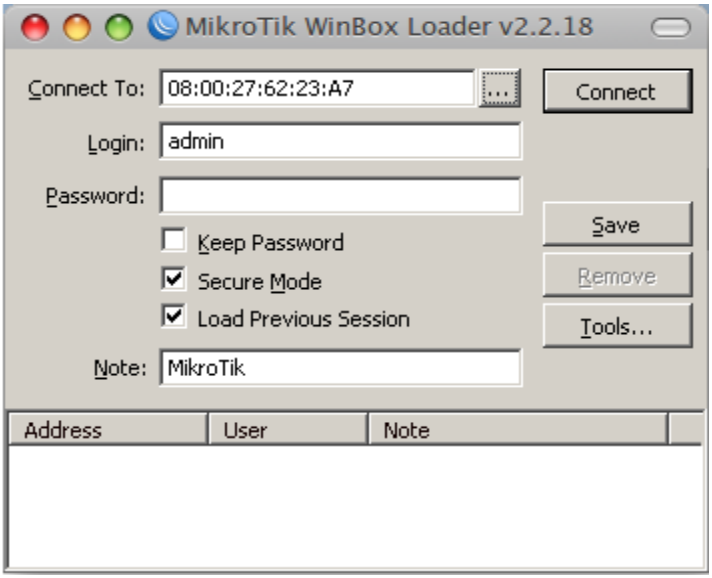


Gambar 1.20 Via web

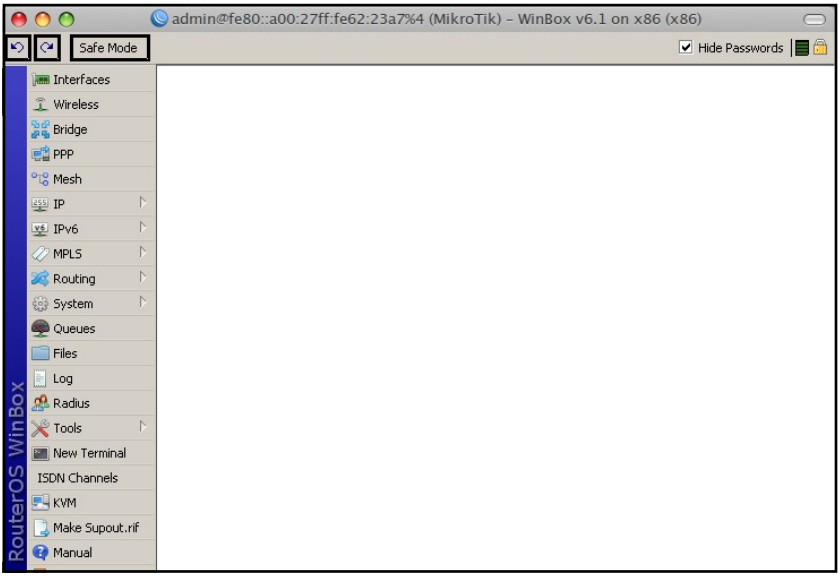
3. Via Winbox

Biasanya cara yang sering dipakai adalah menggunakan *software* Winbox, merupakan aplikasi yang disediakan khusus oleh Mikrotik

digunakan khusus untuk *remote*. Winbox dapat diunduh di http://www.mikrotik.com/download. Untuk menjalankan Winbox dilakukan dengan cara buka file Winbox yang berekstensi .exe maka muncul jendela seperti pada gambar 1.11 dan halaman Mikrotik pada winbox akan tampil seperti pada gambar 1.12. Pada kolom '*Connect To'* dianjurkan menggunakan MAC *address* dari Mikrotik agar apabila ada perubahan IP address tidak terjadi putus koneksi.



Gambar 1.11 *Login* Winbox

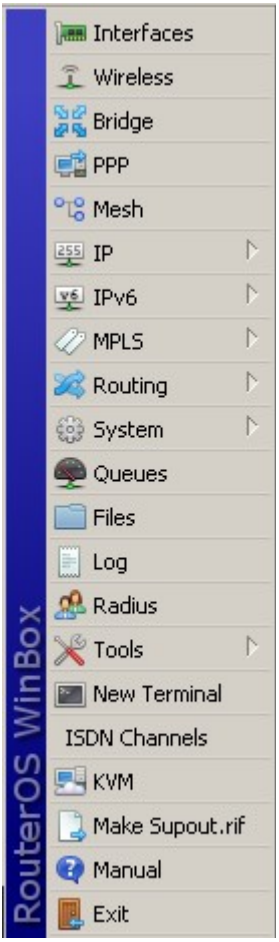


Gambar 1.12 Winbox Mikrotik

Winbox

Karena pada praktiknya dalam buku ini kebanyakan menggunakan Winbox maka akan dibahas terlebih dahulu untuk pengenalan secara detail menu-menu Mikrotik di Winbox.

1. Menu Mikrotik di Winbox, ada banyak menu pada Mikrotik dan masing-masing memiliki fungsi masing-masing, lihat gambar 1.13.



Gambar 1.13 Menu Mikrotik

Interface, berisi daftar interface yang dapat digunakan oleh Mikrotik dalam suatu jaringan. Terdiri dari *Ethernet, EoIP Tunnel, IP Tunnel, GRE Tunnel, VLAN, VRRP, Bonding, LTE*

Wireless, menu *wireless* berisi banyaknya interface *wireless* yang dimiliki sebuah Mikrotik beserta konfigurasinya yang terdiri dari

Nstreme Dual, Access List, Registration, Connect List, Security Profiles, Channels.

Bridge, menu ini digunakan untuk melakukan teknik *bridging* dalam beberapa *interface*.

PPP, merupakan menu yang digunakan untuk melakukan koneksi VPN (*Virtual Private Network*) antara lain: PPP, PPTP, SSTP, L2TP, OpenVPN, PPPoE, ISDN.

Mesh, digunakan untuk melakukan implementasi topologi Mesh.

IP, merupakan menu yang digunakan untuk manajemen network dengan menggunakan teknologi IPv4. Beberapa sub menu dalam menu IP antara lain : ARP, *Accounting, Addresses,* DHCP *Client,* DHCP *Relay*, DHCP *Server*, DNS, *Firewall*, *Hotspot*, Ip*sec, Neighbors, Packing, Pool, Routes,* SMB, SNMP, *Services, Settings, Socks,* TFTP, *Traffice Flow,* UpnP, *Web Proxy*.

IPv6, digunakan untuk manajemen network dengan menggunakan teknologi IPv6. Beberapa sub menu yang ada dalam IPv6 antara lain, *Addresses,* DHCP *Client,* DHCP *Server*, *Firewall*, ND, *Neighbors*, *Pool, Routes.*

MPLS, menu yang digunakan untuk membentuk jaringan yang menggunakan teknologi MPLS.

Routing, menu yang digunakan untuk membentuk rute antar router. Mikrotik mendukung jenis *protocol routing* antara lain : BFD, BGP, *Filters,* IGMP *Proxy,* MME, OSPF, OSPFv3, PIM, *Prefix Lists,* RIP, RIPng.

System, digunakan untuk pengaturan pada sistem Mikrotik antara lain, *Auto Upgrade, Certificates, Clock, Console, Drivers, Health, History, Identify, LCD, LEDs, License, Loggin, NTP Client, NTP Server, Packages, Password, Ports, Reboot, Reset Configuration, Resources, Routerboard, Scheduler, Scripts, Shutdown, Special Login, Stores, UPS, Users, Watchdog.*

Queues, adalah menu yang digunakan untuk melakukan manajemen *bandwidth* baik *upload* maupun *download*.

Files, merupakan tempat dimana semua file (*backup, packages,* dll) tersimpan

Log, merupakan *history* segala aktifitas konfigurasi didalam Mukrotik.

Radius, digunakan untuk konfigurasi radius pada *Hotspot*.

Tools, merupakan kumpulan *tool* yang digunakan untuk keperluan maintenance jaringan antara lain *Btest Server, Bandwidth Test, Email, Flood Ping, Graphing, IP Scan, MAC Server, Netwatch, Packet Sniffer, Ping, Ping Speed, Profile, SMS, Telnet, Torch, Traceroute, Traffic Generator, Traffic Monitor.*

New Terminal, digunakan untuk memunculkan terminal Mikrotik pada Winbox.

ISDN Channels, menampilkan jalur ISDN yang terinstall pada Mikrotik.

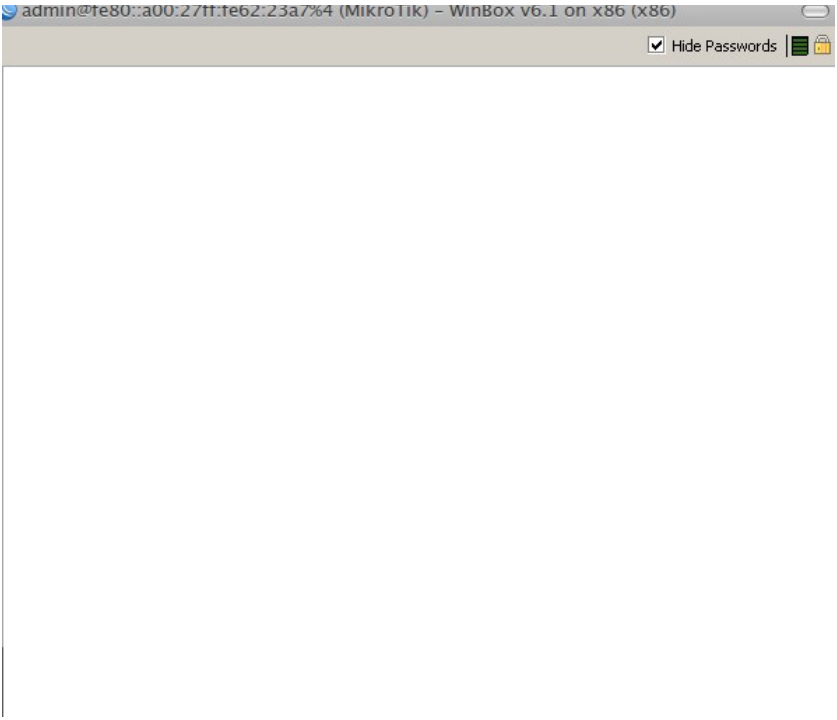
KVM, digunakan untuk membuat *Virtual Machine* berupa *Virtual Router.*

Exit, digunakan bila ingin keluar dari jendela Winbox

2. Undo/Redo, digunakan untuk mengubah konfigurasi ke sebelumnya atau sesudahnya jika terjadi ketidaksesuaian. Lihat gambar 1.14

Gambar 1.13 *Undo/Redo*

3. Area kerja, adalah area dimana jendela konfigurasi Mikrotik berada. Lihat gambar 1.15.



Gambar 1.15 Area kerja Mikrotik

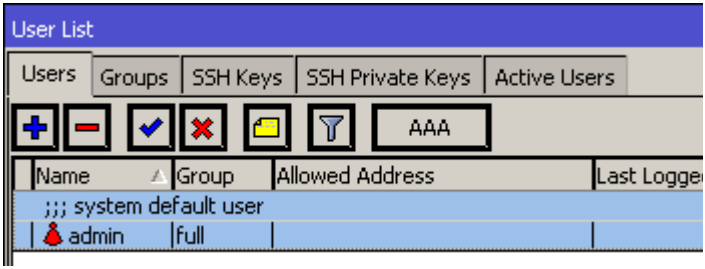
4. *Hide password* dan *Traffic Load*, yaitu bagian pada Winbox yang berfungsi menampilkan atau tidaknya segala text password pada Mikrotik dan menginformasikan traffic *resource* yang digunakan Mikrotik. Lihat gambar 1.16

Gambar 1.16 *Hide password* dan *traffic load*

BAB II Konfigurasi Dasar Mikrotik

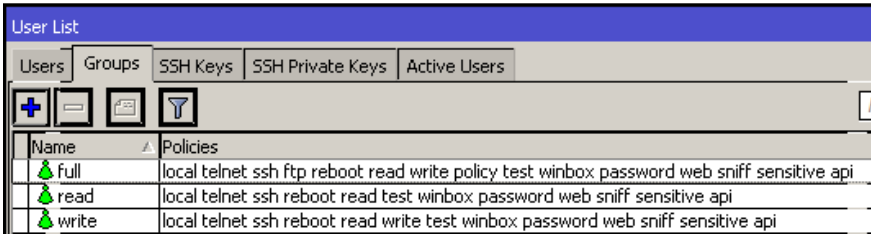
User Management Mikrotik

Secara default Mikrotik memiliki user yang bernama 'admin' yang bisa dilihat di Menu *System* > *Users,* seperti pada gambar 2.1. Kita bisa menambahkan user kita sendiri.



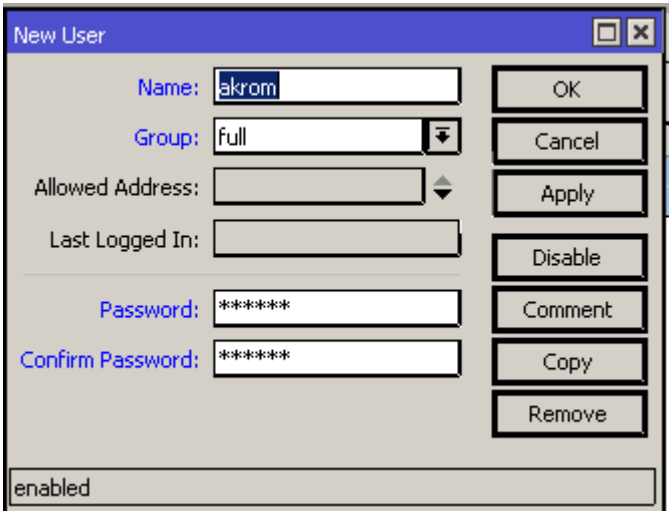
Gambar 2.1 *User* Mikrotik

Untuk menambah user kita sendiri klik *Add,* isikan pada kolon *Name, Group* dan *Password*. Ada 3 jenis *group* secara *default* yaitu *full, write* dan *read.* Untuk lebih jelas perbedaan dari ketiga *group* dapat kita lihat di tab menu *groups.* Lihat gambar 2.2



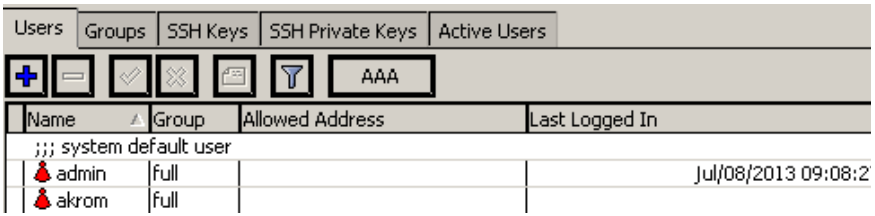
Gambar 2.2 *Groups*

Kemudian melanjutkan untuk penambahan user baru adalah seperti gambar 2.3 dibawah ini.



Gambar 2.3 *User* baru

Setelah OK, akan muncuk user baru sesuai dengan yang ditambahkan yang bernama 'akrom', lihat gambar 2.4



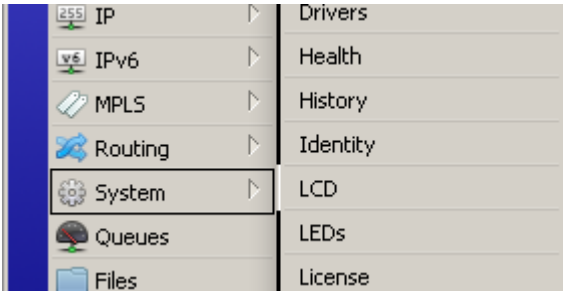
Gambar 2.4 *User* baru

Identitas Mikrotik

Secara defaukt router Mikrotik memiliki identitas dengan nama 'MikroTik', seperti pada gambar 2.5.

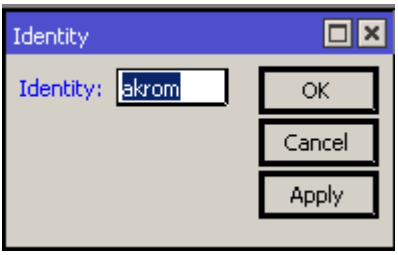
Gambar 2.5 Identitas *default*

Untuk merubah identitas dapat diubah dengan cara lebih mudah menggunakan Winbox ke menu ***system*** **–** ***identity***. Untuk lebih jelasnya lihat gambar 2.6



Gambar 2.6 Mengubah identitas router

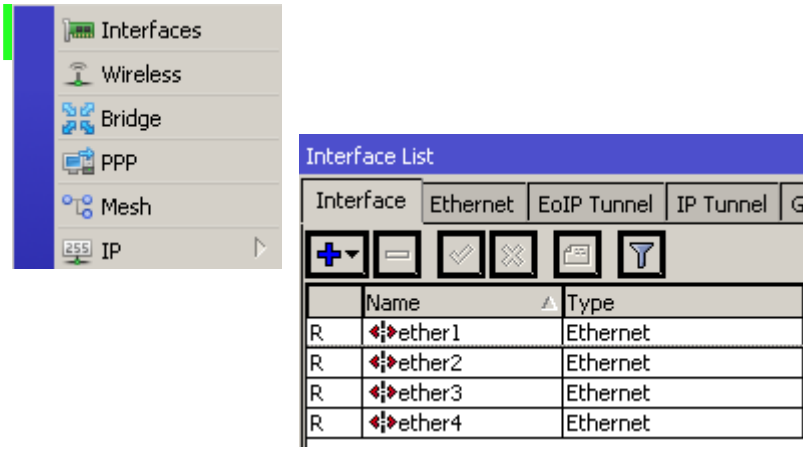
Ubahlah identitas tersebut sesuai dengan keinginan, dalam contoh kali ini akan diubah dengan nama 'akrom'. Seperti pada gambar 2.7



Gambar 2.7 *Identity*

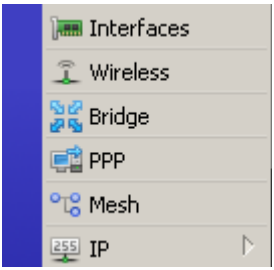
Interface Mikrotik

Untuk melihat banyaknya *interface* jaringan router sebenarnya kita hanya perlu melihat pada bagian portnya saja, namun untuk interface tersebut dapat juga kita lihat di Winbox pada menu *interface*. Pada menu tersebut akan terlihat beberapa interface jaringan dengan tipe dan nama masing-masing. Pada umumnya tipe yang ada dalam menu *interface* berupa *ethernet.* Seperti pada gambar 2.8



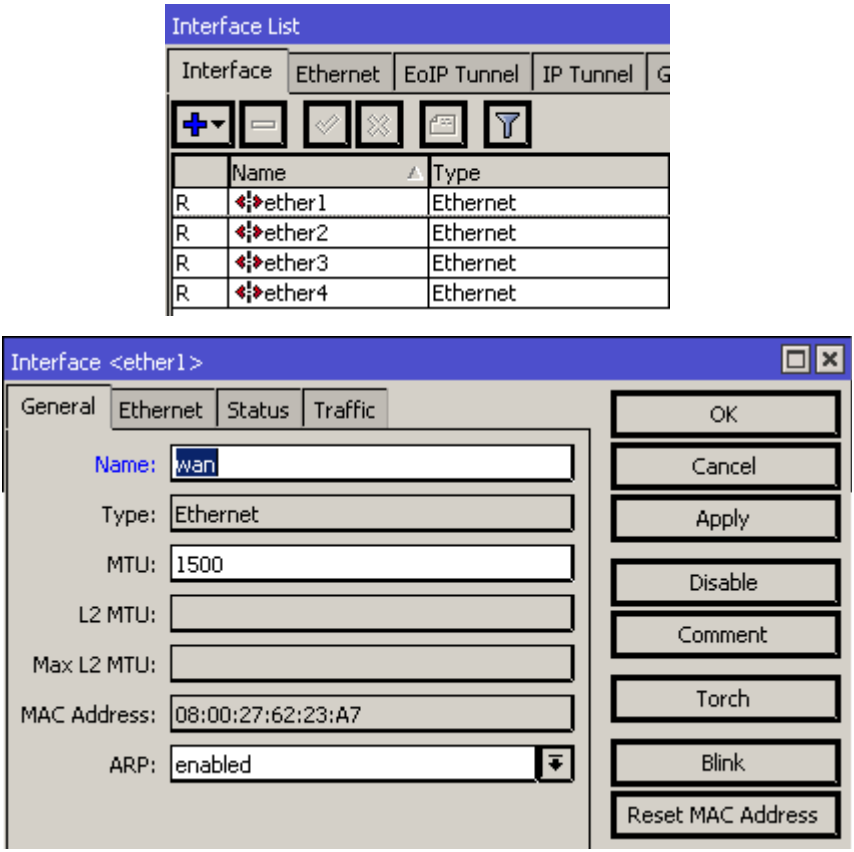
Gambar 2.8 *Interface* Mikrotik

Bila router terpasang *wireless card* dapat dilihat di menu *Wireless*. Lihat gambar 2.9



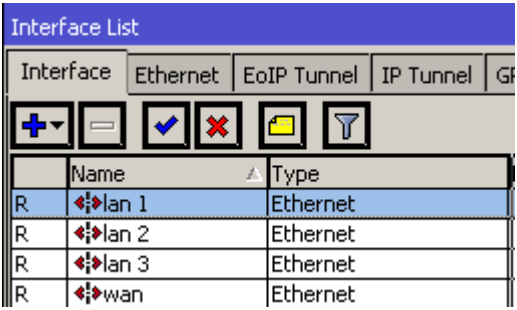
Gambar 2.9 Menu *wireless*

Untuk memudahkan kita manajemen jaringan sebuah nama dari *interface* dapat juga kita ganti dengan nama yang lain. Misalkan *ether 1* diubah dengan nama *wan, ether 2* diubah dengan nama *lan 1, ether 3* diubah dengan nama *lan 2, ether 4* diubah dengan nama *lan 3*. Mengubah nama *interface* lakukan dengan cara *double click* pada *interface* yang akan diubah, kemudian ketik nama yang sesuai dengan yang ditentukan. Seperti pada gambar 2.10



Gambar 2.11 Ubah nama *interface*

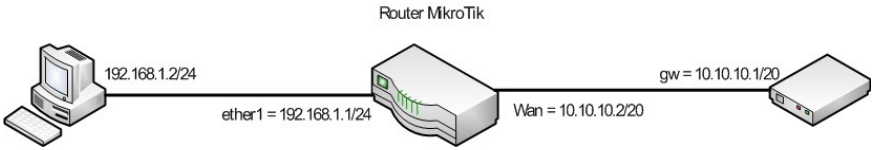
Lakukan pada semua *interface* dengan mengubah namanya sesuai dengan ketentuan, sehingga seperti gambar 2.12



Gambar 2.12 Daftar *interface*

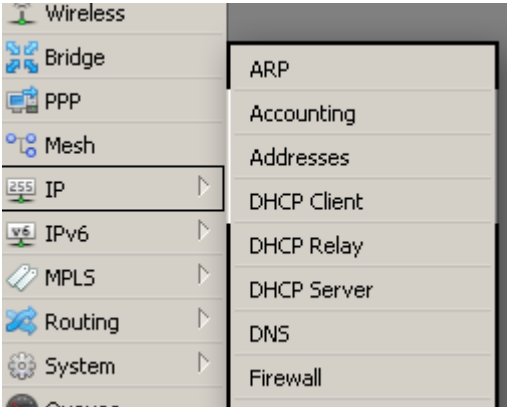
Menambahkan IP Address

Setiap perangkat yang terlibat dalam suatu jaringan pasti butuh sebuah alamat untuk dapat berkomunikasi dengan perangkat lainnya dalam jaringan tersebut. Dalam perangkat jaringan khususnya router dapat mengenali alamat berupa alamat IP (IP *Address*). Mikrotik dapat mengenali IP dengan versi IPv4 dan IPv6. IPv4 masih digunakan hingga saat ini, karena ketersediaannya yang hampir habis maka akan digantikan oleh IPv6. Untuk konfigurasi IP didalam Mikrotik cukup sederhana, dalam sebuah kasus misalkan ada topologi jaringan seperti pada gambar 2.13. Kemudian akan dilakukan konfigurasi IP address seperti berikut.



Gambar 2.13 Topologi jaringan

Untuk konfigurasi IP *address* menggunakan Winbox di Mikrotik adalah dengan klik pada menu IP > *address*. Seperti pada gambar 2.14



Gambar 2.14 Setting IP *address*

Maka akan muncul halaman daftar IP *address* yang masih kosong seperti pada gambar 2.15. Untuk menambahkan klik *add* atau simbol plus pada jendela tersebut.

Gambar 2.15 *Address list*

Untuk interface *lan1* tambahkan IP *address* 192.168.1.1/24 seperti berikut, lihat gambar 2.16.

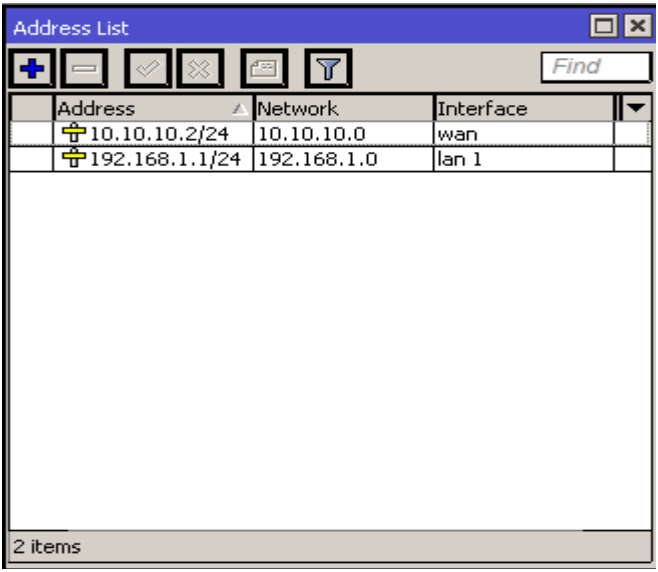
Gambar 2.16 *Setting* IP *address interface* lan1

Klik *Apply*, kemudian OK

Kemudian tambahkan juga IP *address* pada *interface* wan dengan IP 10.10.10.2/20. Lihat gambar 2.17

Gambar 2.17 *Setting* IP *address interface* wan

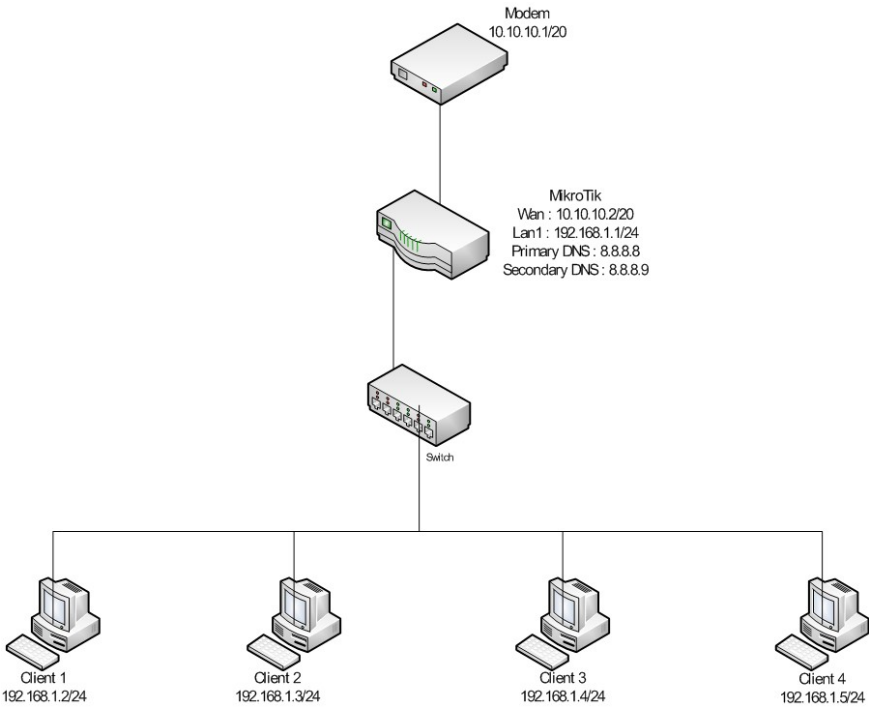
Klik *Apply,* kemudian OK. Lihat hasil konfigurasi seperti pada gambar 2.18 berikut.



Gambar 2.18 Lihat IP *address*

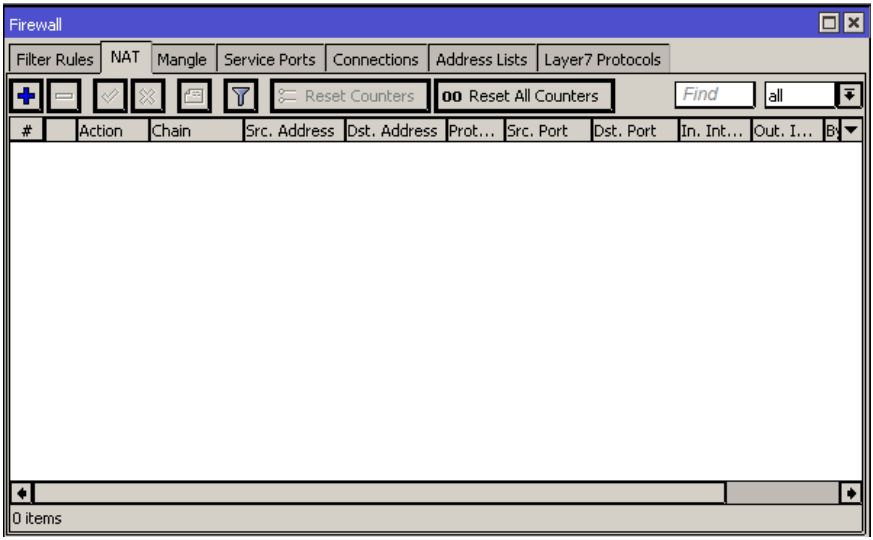
Konfigurasi Internet Mikrotik

Salah satu fungsi router adalah sebagai *gateway* suatu jaringan LAN kali ini kita akan mencoba konfigurasi Mikrotik sebagai *gateway* untuk meneruskan koneksi internet dari ISP ke jaringan LAN menggunakan NAT (*Network Address* Translation) dengan menggunakan topologi jaringan yang sama pada materi sebelumnya.



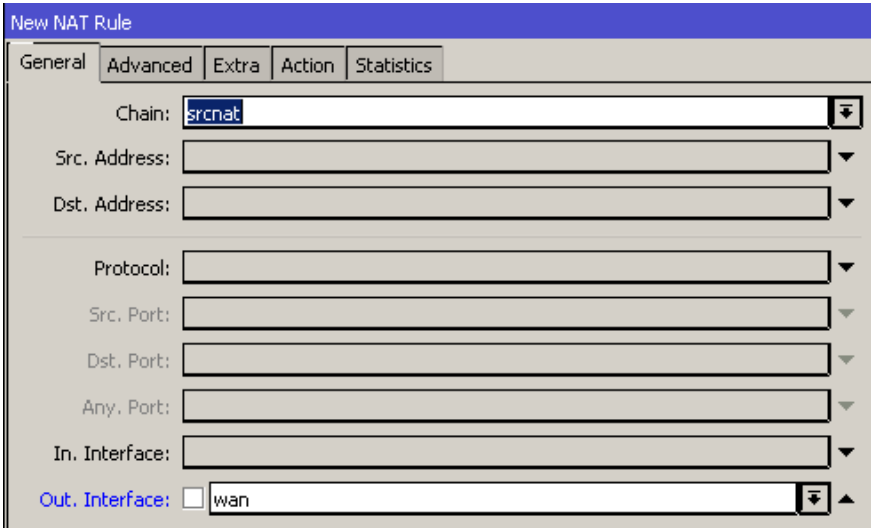
Gambar 2.19 Topologi Mikrotik Internet

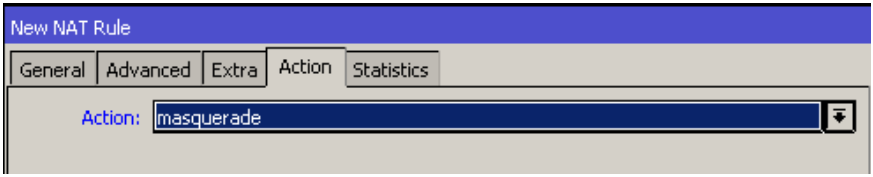
Untuk konfigurasi NAT dapat menggunakan cara klik menu Winbox pada IP > *Firewall* kemudian klik pada tab NAT seperti pada gambar 2.20



Gambar 2.20 Jendela *Firewall*

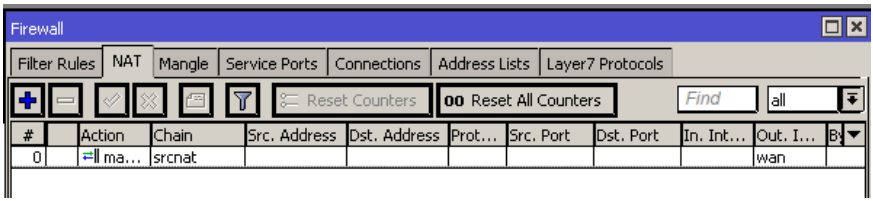
Tambahkan NAT dengan klik *add* pada tab *general* tambahkan konfigurasi kolom *chain* dengan *srcnat* dan *out interface* dengan *interface wan*. Kemudian masuk ke tab *action* isikan kolom *action* dengan pilihan *masquerade*. Untuk lebih jelasnya lihat gambar 2.21





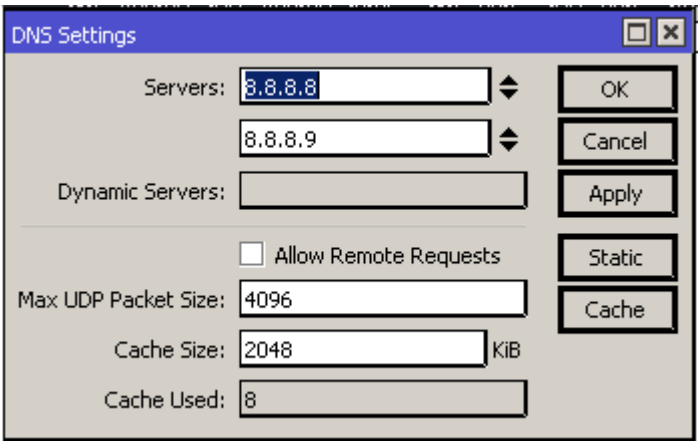
Gambar 2.21 Konfigurasi NAT

Kemudian klik *Apply* dan OK. Lihat hasil konfigurasi NAT seperti pada gambar 2.22



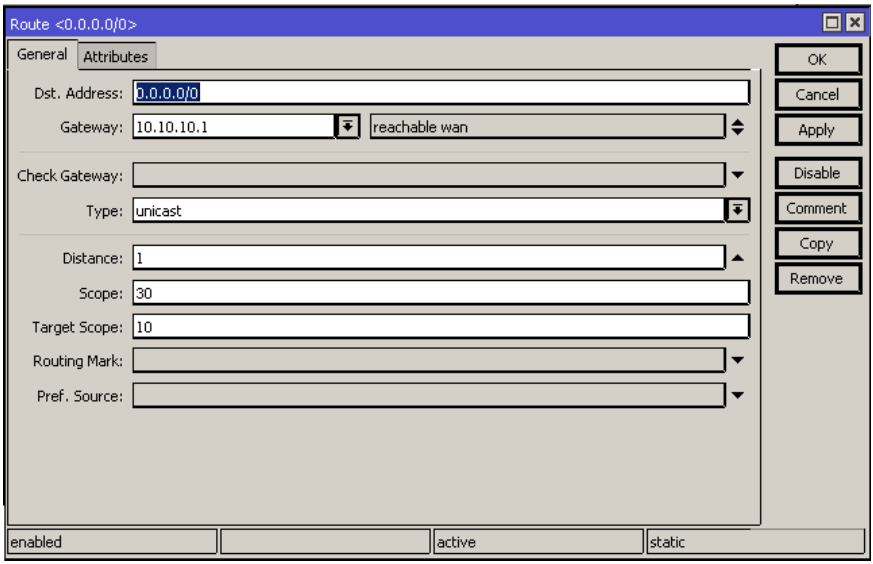
Gambar 2.22 Hasil konfigurasi NAT

Langkah berikutnya adalah setting DNS pada router Mikrotik dan default route untuk bisa mengkases internet. Untuk setting DNS klik pada menu IP > DNS. Isikan pada kolom *Servers* dengan IP DNS.



Gambar 2.23 Setting DNS

Kemudian setting routing untuk menentukan *default gateway* dengan cara klik IP > *route* dan tambahka konfigurasi seperti gambar 2.24



Gambar 2.24 Konfigurasi *default gateway*

Dst. Address : 0.0.0.0/0

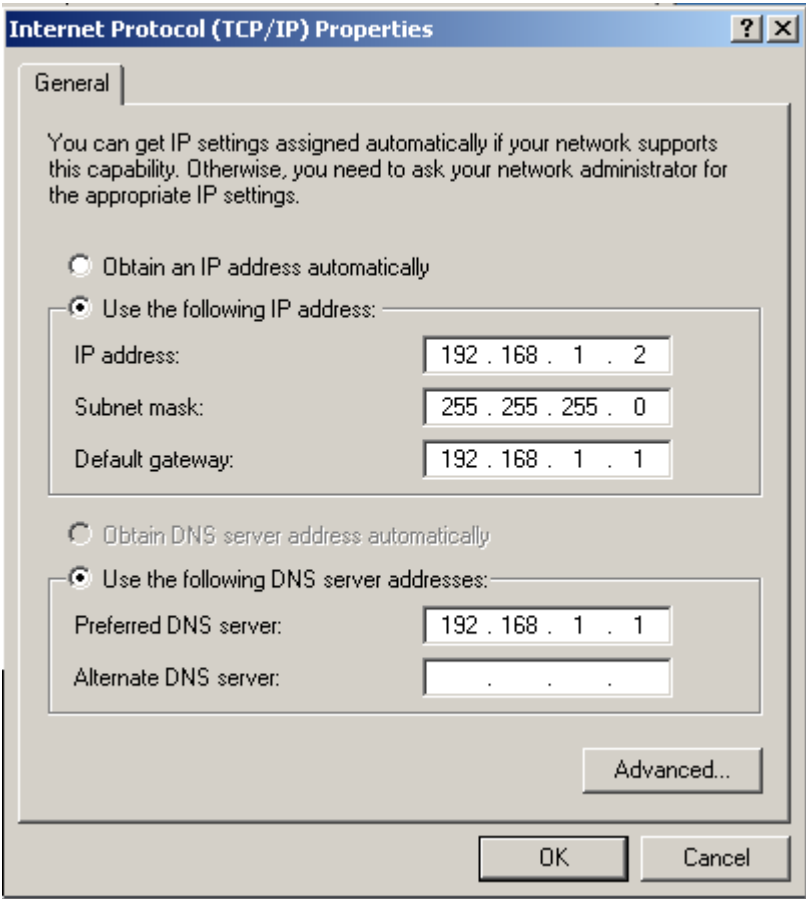
Gateway : 10.10.10.1

Untuk menguji koneksi internet cobalah ping ke www.google.com dari terminal Mikrotik. Bila muncul pesan balasan berupa TTL dan *time* maka setting internet berhasil. Lihat gambar 2.25

```
[admin@akrom] > ping google.com
HOST                                     SIZE TTL TIME  STATUS
173.194.72.139                             56  44 895ms
173.194.72.139                             56  44 589ms
173.194.72.139                             56  44 591ms
173.194.72.139                             56  44 554ms
173.194.72.139                             56  44 664ms
173.194.72.139                             56  44 613ms
173.194.72.139                             56  44 561ms
173.194.72.139                             56  44 591ms
    sent=8 received=8 packet-loss=0% min-rtt=554ms avg-rtt=632ms
   max-rtt=895ms
```

Gambar 2.25 Tes ping

Kemudian sebagai contoh setting IP pada client sehingga client tersebut mendapatkan koneksi internet, seperti contoh gambar 2.26 dibawah ini.



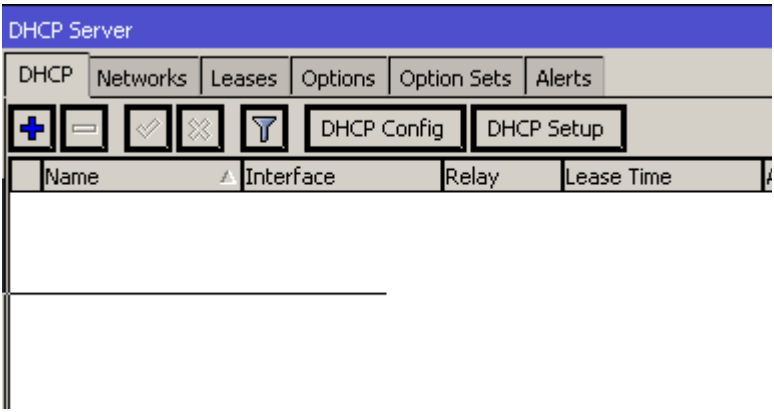
Gambar 2.26 IP client

Setelah selesai konfigurasi IP *client* sekarang cobalah buka browser di *client* tersebut dengan membuka salah satu situs di internet, pastikan berhasil.

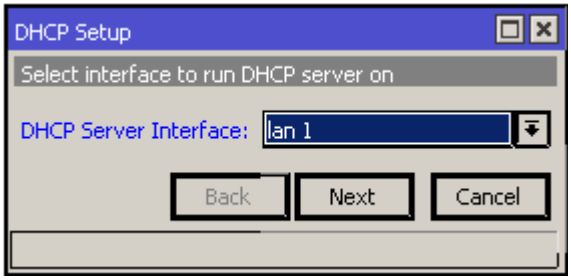
DHCP Server Mikrotik

DHCP *server* adalah sebuah layanan yang memberikan konfigurasi IP secara otomatis dari sebuah *server/router* client. Hal ini biasanya digunakan untuk mempermudah pemberian IP pada jaringan dengan skala besar.

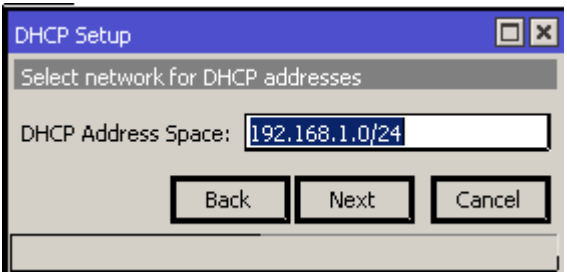
Untuk setting DHCP *server* pada Mikrotik adalah dengan klik pada menu IP > DHCP *servers* kemudian klik pada DHCP *setup,* maka akan muncul keluar urutan konfigurasi seperti gambar 2.27 dibawah ini.



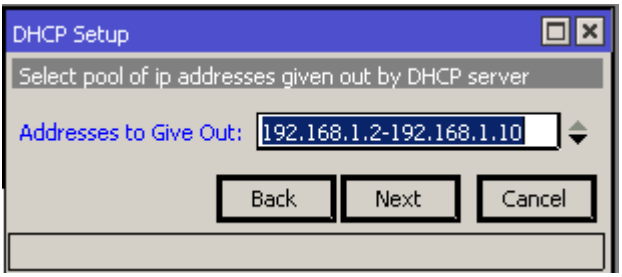
(a)



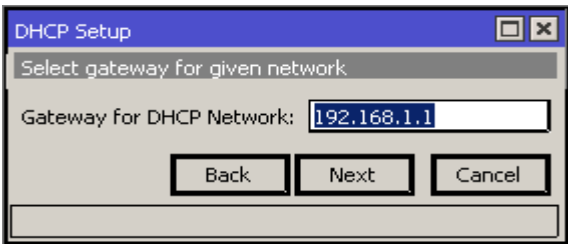
(b)


DHCP Setup
Select network for DHCP addresses
DHCP Address Space:
192.168.1.0/24
Back
Next
Cancel

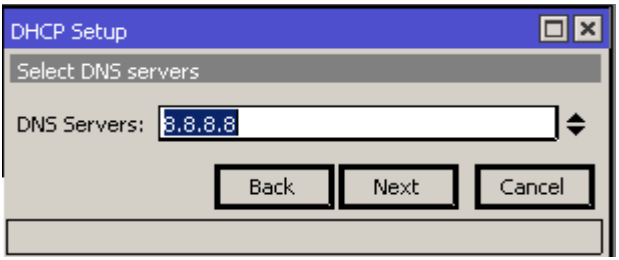
(c)


DHCP Setup
Select pool of ip addresses given out by DHCP server
Addresses to Give Out:
192.168.1.2-192.168.1.10
Back
Next
Cancel

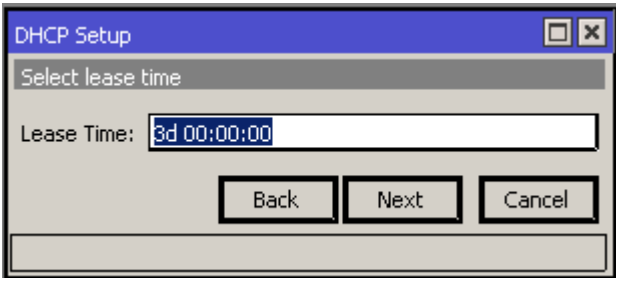
(d)


DHCP Setup
Select gateway for given network
Gateway for DHCP Network:
192.168.1.1
Back
Next
Cancel

(e)


DHCP Setup
Select DNS servers
DNS Servers:
8.8.8.8
Back
Next
Cancel

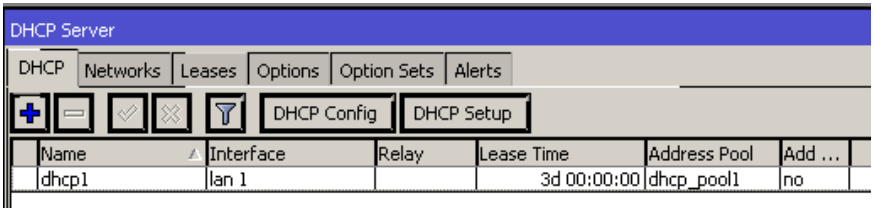
(f)



(g)

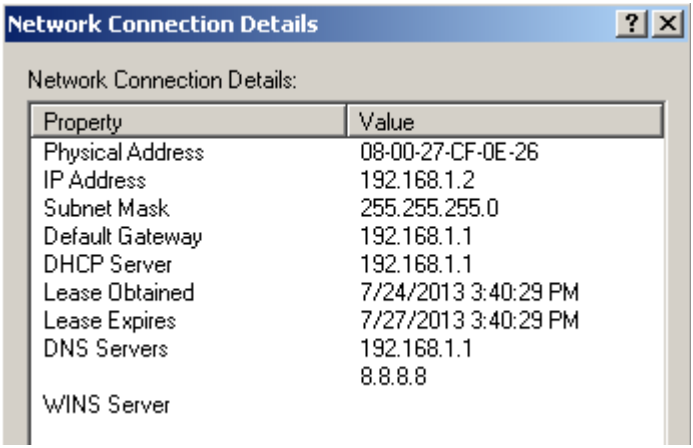
Gambar 2.27 (a) DHCP *setup* (b) DHCP *serve interface* (c) DHCP *address space* (d) *Addresses to give out* (e) *Gateway for DHCP network* (f) DNS *server* (g) *Lease time*

Lihat hasil konfigurasi DHCP *server* seperti pada gambar 2.28 dibawah ini.



Gambar 2.28 Hasil Konfigurasi DHCP

Untuk melakukan testing DHCP server maka pada *client* ubah setting IP menjadi *obtain* apabila menggunakan OS Windows. Apabila dicheck maka IP client adalah seperti gambar 2.29 berikut

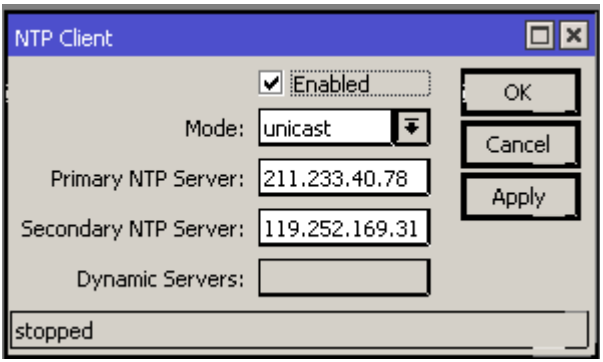


Gambar 2.29 DHCP *client*

Network Time Protocol

Kebanyakan *router* Mikrotik tidak memiliki battery di *board*nya. Sistem waktu yang akurat dan aktual sangat dibutuhkan ketika kita menggunakannya untuk monitoring log dengan menggunakan tool *scheduler* dan *netwatch*. Maka dari itu NTP (*Network Time Protocol*) memungkinkan *router* untuk melakukan sinkronisasi dengan *server* lainnya dalam jaringan. Mikrotik dapat mendukung NTP *server* dan NTP *client.*

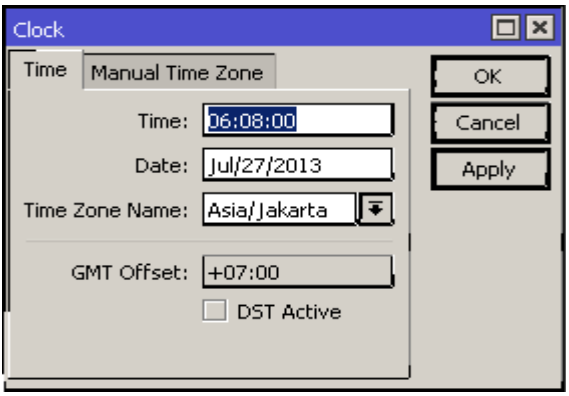
Cobalah setting NTP *client* pada router sehingga waktu mengacu pada waktu internasional (GMT+7), seperti contoh NTP diarahkan ke *public* NTP server **asia.pool.ntp.org** atau **id.pool.ntp.org.** Klik menu *system* > NTP *client.* Lihat gambar 2.30a



Gambar 2.30 NTP *client*

Mode : *unicast*
Primary NTP Server : 211.233.40.78
Secondary NTP Server : 119.252.169.31

Kemudian ubahlah *timezone* menjadi Asia/Jakarta pada menu *system* > *clock* seperti pada gambar 2.31



Gambar 2.31 *Clock*

BAB II Firewall

Firewall

Firewall diperlukan untuk melindungi *router* dari akses yang tidak diinginkan baik dari jaringan lokal maupun jaringan internet. Firewall juga digunakan untuk menyaring akses antar *network* yang melalui router. Dalam Mikrotik *firewall* diimplementasikan dalam fitur *rule* dan NAT.

Chain

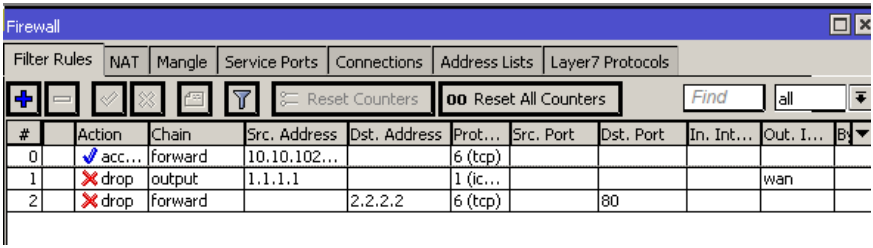
Dalam *firewall filter* ada 3 *chain* utama dalam Mikrotik, yaitu :

Chain input, digunakan untuk memproses paket yang memasuki router melalui salah satu *interface router* dengan alamat IP yang merupakan salah satu IP *router.*

Chain forward, lain halnya dengan *input, chain forward* digunakan untuk memfilter paket yang masuk ke router kemudian diterukan ke suatu tujuan tertentu.

Chain output, aturan ini digunakan untk memproses paket yang berasal dari *router* yang kemudian dikeluarkan ke tujuan tertentu.

Pada daftar *filrewall* yang telah dibuat pada tabel setiap aturan *chain* yang dibuat akan dibaca oleh *router* dari atas ke bawah. Seperti pada gambar 3.1 dibawah ini.



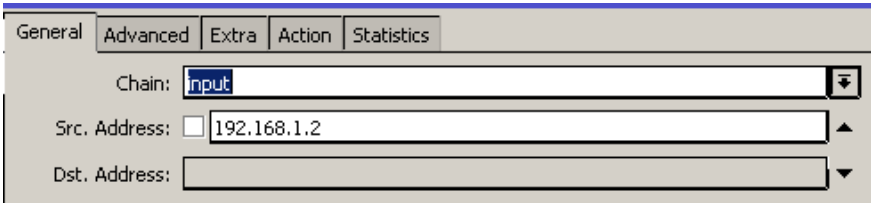
Gambar 3.1 Tabel *firewall*

Chain memiliki prinsip kerja "jika maka". "Jika" kondisi memenuhi syarat pada *rule* yang kita buat pada tab *general*. "Maka" akan diaksikan paket tersebut pada tab *action.*

Firewall Strategy

Kali ini kita kan mencoba buat *firewall* sederhana yang memperbolehkan sebuah *client* saja dengan sebuah alamat IP yang bisa mengakses ke router. Kita akan membuat rule dengna strategy *accept few and drop any* yang artinya memperbolehkan akses pada beberapa dan menolak semuanya.

Buat aturan *firewall accept few* dengan klik menu IP > *firewall* > *filter rules.* Misalkan IP client adalah 192.168.1.2. Lihat gambar 3.2 untuk menentukan kondisi "jika".

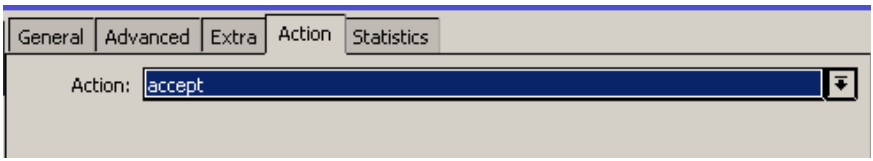


Gambar 3.2 *firewall input*

Gambar diatas mengartikan pada kondisi *input* ke router dengan alamat IP sumber 192.168.1.2.

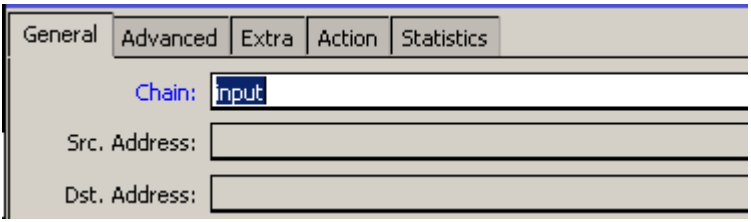
Kemudian dengan kondisi "jika" tersebut ditentukan kondisi "maka"

seperti pada gambar 3.3 dibawah ini.



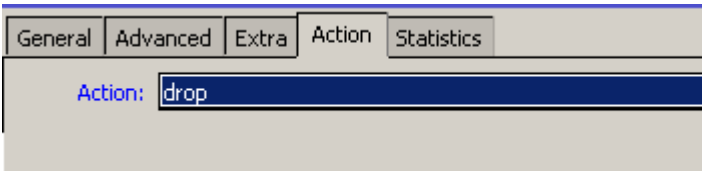
Gambar 3.3 Tab *action*

Langkah selanjutnya adalah membuat strategi *drop any traffic*. Buat kembali kondisi "jika" seperti gambar 3.4 berikut.



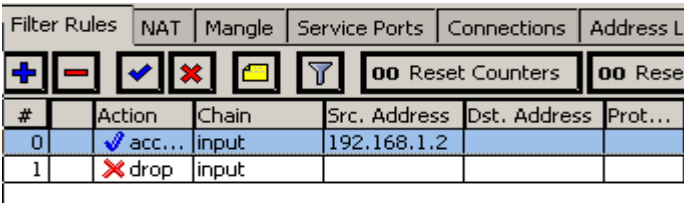
Gambar 3.4 Tab *general*

Kemudian buat kondisi "maka" dengan *action drop* seperti pada gambar 3.5



Gambar 3.5 Tab *action*

Pada tabel *firewall* akan muncul 2 buah rule seperti pada Gambar 3.6

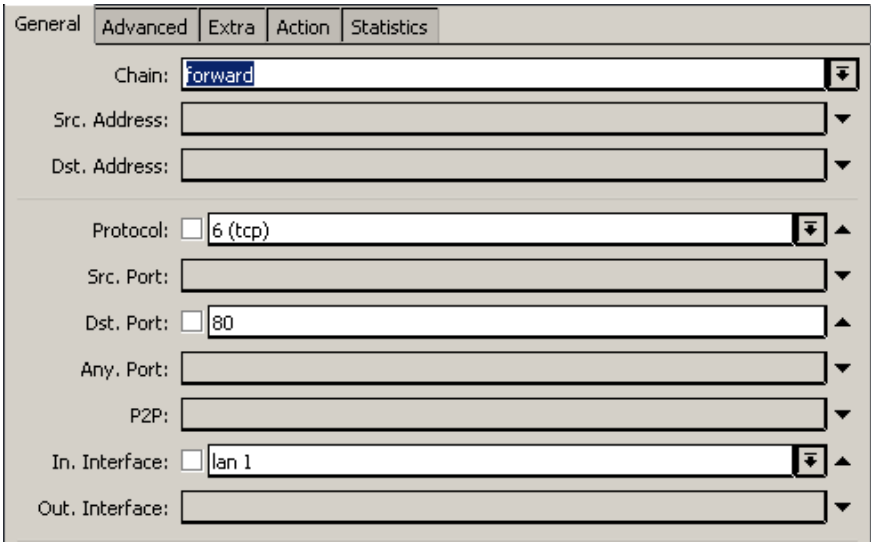


Gambar 3.6 Tabel *firewall*

Cobalah ping dari client yang memiliki IP *address* 192.168.1.2 seharusnya mendapat balasan *reply* kemudian rubahlah IP address *client* menjadi 192.168.1.3 atau yang lainnya maka akan mendapat balasan *request time out*.

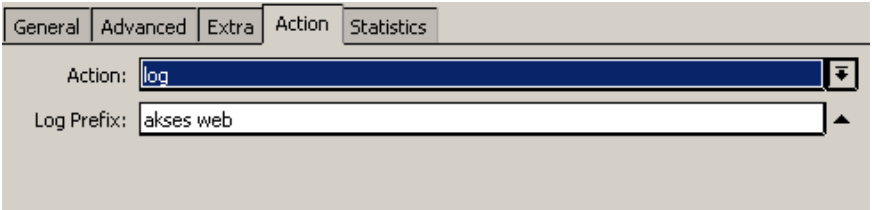
Firewall Logging

Firewall logging adalah salah satu fitur yang berfungsi mencatat segala aktifitas jaringan tertentu, semisal dalam hal ini kita akan mencatat segala akses *website* dari client yang melewati *router* maka dapat kita buat sebuat aturan seperti berikut. Lihat Gambar 3.7



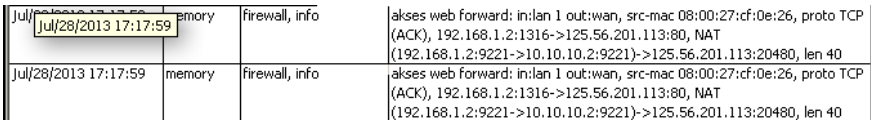
Gambar 3.7 *firewall web log*

Konfigurasi seperti gambar diatas mempunyai maksud yaitu pada *chain forward* dengan *protocol* TCP dan tujuan *port* 80 yaitu *port* http (web) yang melalui *interface* lan1 sebagai *input interface router* akan dikenakan aturan pada tab *action.* Lihat gambar 3.8



Gambar 3.8 Tab *action firewall log*

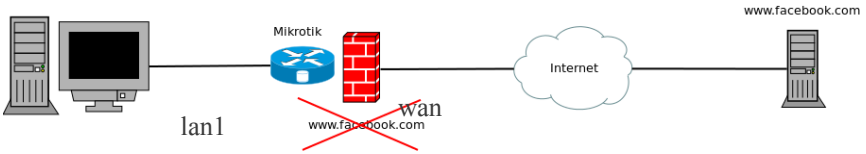
Isikan *action* tersebut adalah dengan '*log*' yaitu memasukkan aktifitas jaringan yang cocok pada rule yang telah dibuat sebelumnya ke dalam aktifitas *log router*. Untuk melihat hasil log yang masuk pada *router* klik pada menu *Log* di Winbox. Seperti pada gambar 3.9 dibawah ini. Sebelumnya cobalah akses sebuah *website* dari *client.*



Gambar 3.9 *Log firewall*

Firewall Blocking Host

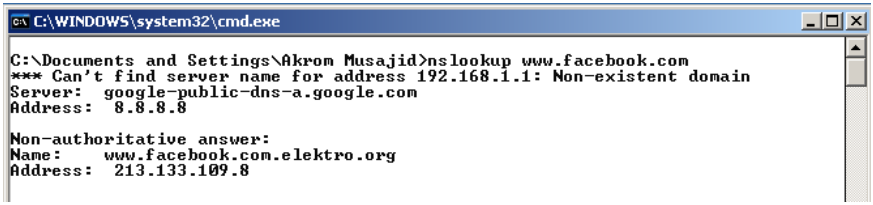
Firewall blocking host adalah teknik *firewall* dimana kita akan memlokir beberapa IP dari suatu jaringan yang melewati *router*. Pada contoh kasus kali ini kita akan coba blok akses ke suatu *website* menggunakan IP. Lihat gambar 3.10



Gambar 3.10 Diagram *firewall block host*

Seperti pada gambar diatas menjelaskan bahwa pada jaringan lokal tidak akan diijinkan mengakses www.facebook.com. Dalam contoh kali ini akan menggunakan teknik *firewall* dengan cara meng-*drop* menggunakan alamat IP. Yang harus pertama kali dilakukan adalah kita harus mengetahui alamat IP dari *facebook*. Hal tersebut dapat dilakukan dengan

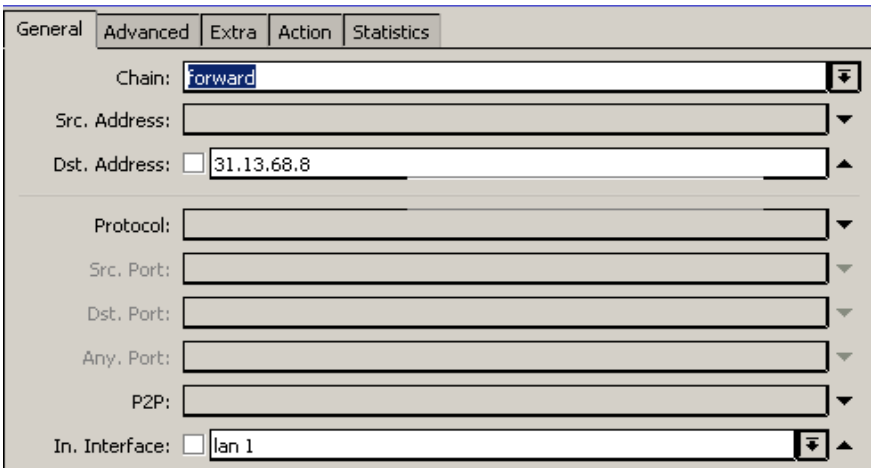
cara *nslookup*. Perintah ini dapat kita ketikkan pada *terminal/command prompt* client. Lihat Gambar 3.11



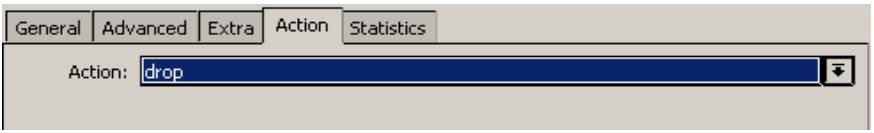
Gambar 3.11 *nslookup*

Setelah dilihat dengan *nslookup, facebook* memiliki IP address 31.13.68.8 dan 31.13.68.16. Maka dari itu kita dapat membuat aturan yang akan memblok akses www.facebook.com dari jaringan lokal.

Klik pada menu IP > *Firewall* > *Tab Filter Rules.* Klik *add* dan tambahkan aturan seperti pada gambar 3.12 (a) dan (b) di bawah ini.



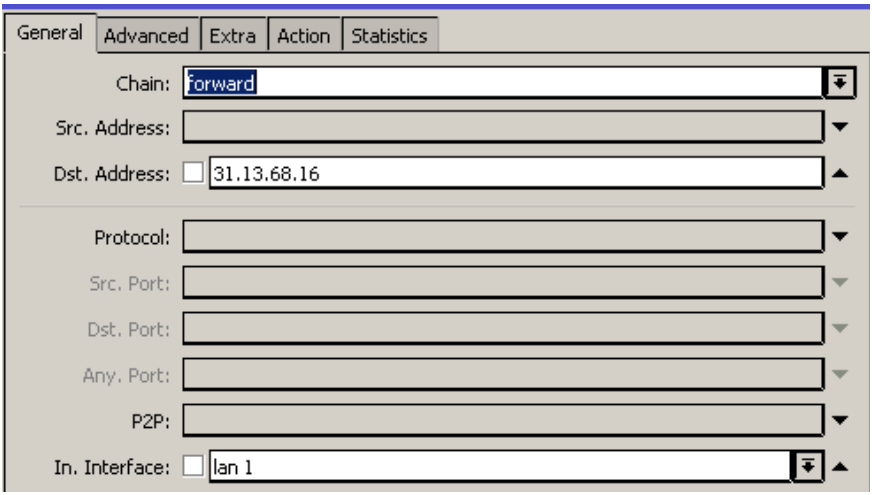
(a)



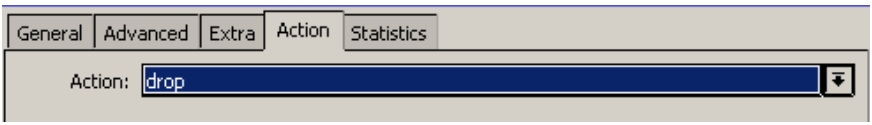
(b)

Gambar 3.12 (a) *Tab general (b) Tab action*

Kemudian tambahkan lagi dengan IP *address* kedua *facebook* dengan konfigurasi yang sama. Lihat Gambar 3.13 (a) dan (b).



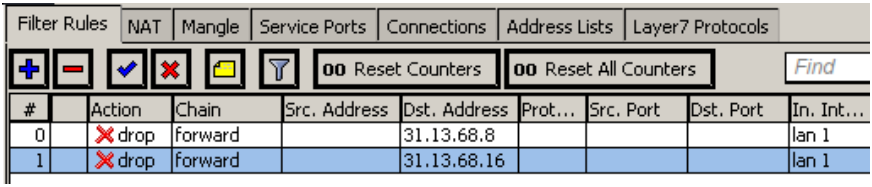
(a)



(b)

Gambar 3.13 (a) *Tab general* (b) *Tab action*

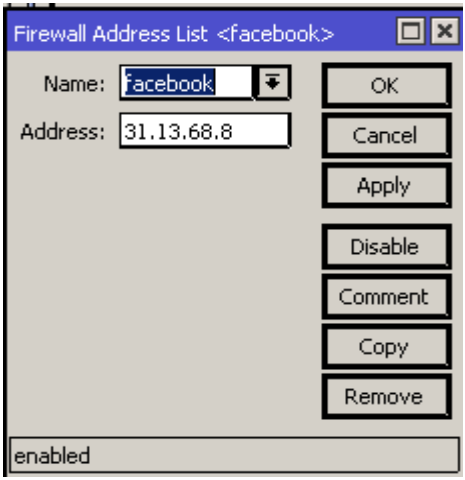
Sehingga menghasilkan tabel *firewall* seperti gambar 3.14 dibawah ini.



Gambar 3.14 Tabel *firewall* blok *facebook*

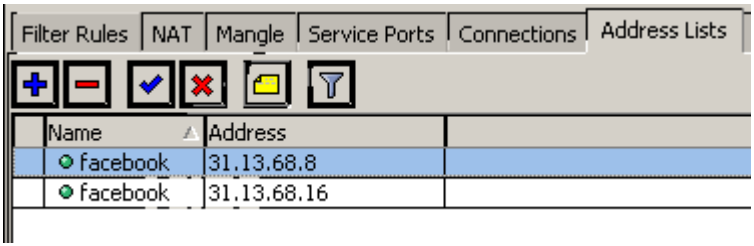
Cobalah akses *facebook* dari client, seharusnya koneksi tidak akan berhasil karena terblock oleh *firewall* Mikrotik.

Untuk menghemat tabel *firewall,* kita juga dapat membuat sebuah *address list* terlebih dahulu dengan cara klik menu IP > *Firewall* > *Tab Address Lists*. Seperti pada gambar 3.15 di bawah ini.



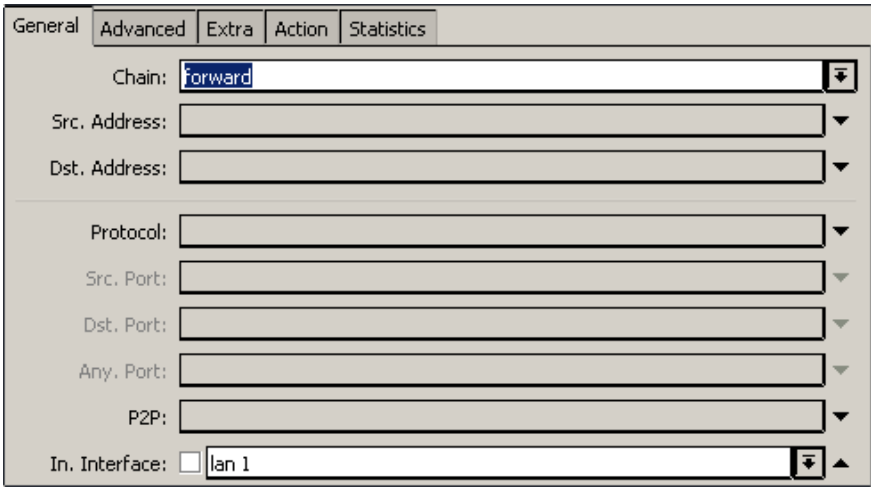
Gambar 3.15 *Address list facebook*

Klik OK, kemudian tambahkan lagi IP *facebook* yang lain. Bila dilihat tabel *address list* dari *facebook* adalah sebagai berikut. Lihat gambar 3.16.

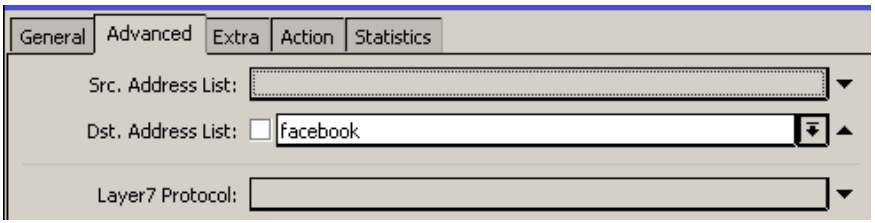


Gambar 3.16 Tabel *address list facebook*

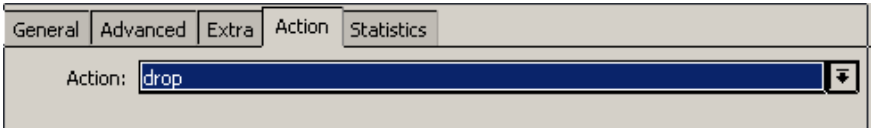
Kemudian untuk menambahkan *rule* nya adalah seperti pada gambar 3.17 (a), (b) dan (c).



(a)



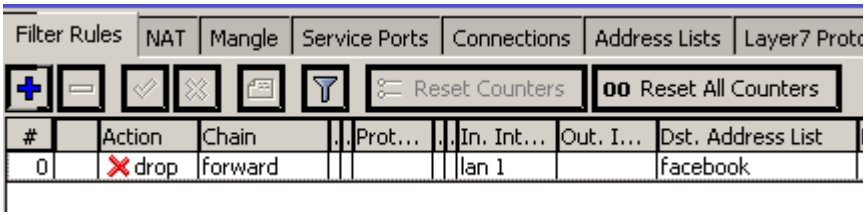
(b)



(c)

Gambar 3.17 (a) *Tab general* (b) *Tab advanced* (c) *Tab action*

Kemudian bila dilihat tabel *rule firewall* adalah seperti pada gambar 3.18.



Gambar 3.18 Tabel *firewall* blok *facebook*

Connection Tracking & Connection State

Connection Tracking mempunyai kemampuan untuk menyimpan dan menjaga informasi koneksi seperti koneksi baru atau koneksi yang sudah ada yang disertai dengan jenis *protocol*, alamat IP asal dan alamat IP tujuan. Dengan menggunakan fitur ini, para *administrator* dapat menolak atau mengijinkan berbagai macam koneksi. *Connection Tracking* mempunyai beberapa keadaan, antara lain :

New, sebuah *client* me-*request* koneksi melalui *firewall,* seperti ada suatu keadaan perangkat 1 menghubungi perangkat 2 dengan mengirimkan paket SYN (*synchronize*).

Established, merupakan sebuah koneksi yanng sudah diketahui sebelumnya.

Related, paket memulai koneksi baru pada koneksi sebelumnya, seperti transfer data pada FTP atau pesan eror ICMP

Invalid, sebuah keadaan dimana tidak ada keadaan seperti 3 keadaan daiatas.

Untuk membuat rule *Connection Tracking* adalah dengan menggunakan *connection state,* seperti berkut :

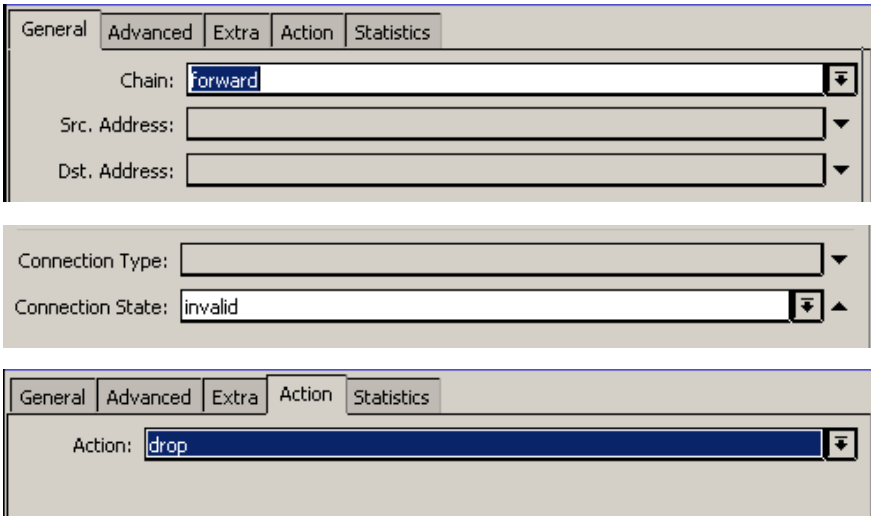
Connection state invalid → drop

Connection state established → accept

Connection state related → accept

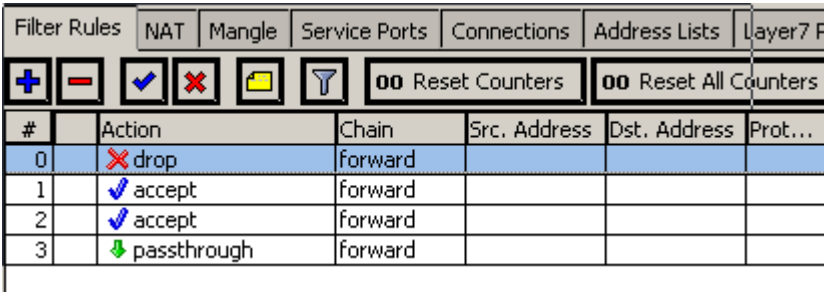
Connection state new → passtrough

Masuk menu IP > *Firewall* > *Filter Rules*. Buatlah rule seperti pada gambar 3.19 dibawah ini.



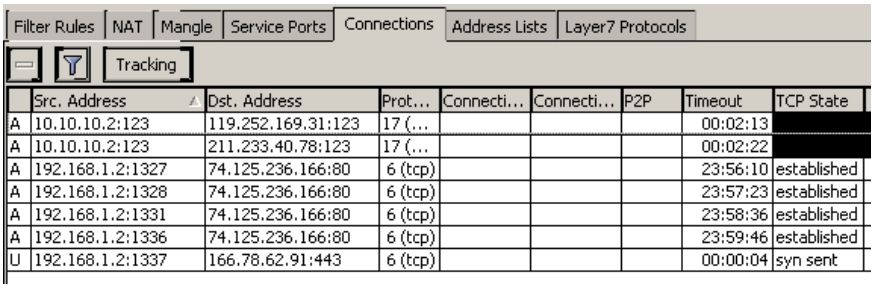
Gambar 3.19 *Connection state rule*

Gambar diatas adalah konfigurasi untuk *state invalid*, selanjutnya lakukan konfigurasi untuk *state* yang lainnya. Sehingga hasil tabel *firewall* yang dibuat adalah seperti pada gambar 3.20.



Gambar 3.20 Tabel *firewall connection state*

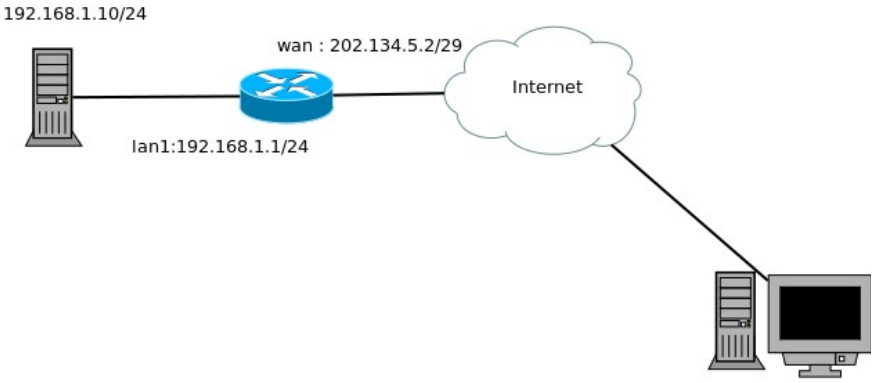
Untuk melihat hasil dari konfigurasi *connection tracking* dapat dilihat di menu IP > *Firewall* > *Connections.* Lihat gambar 3.21



Gambar 3.21

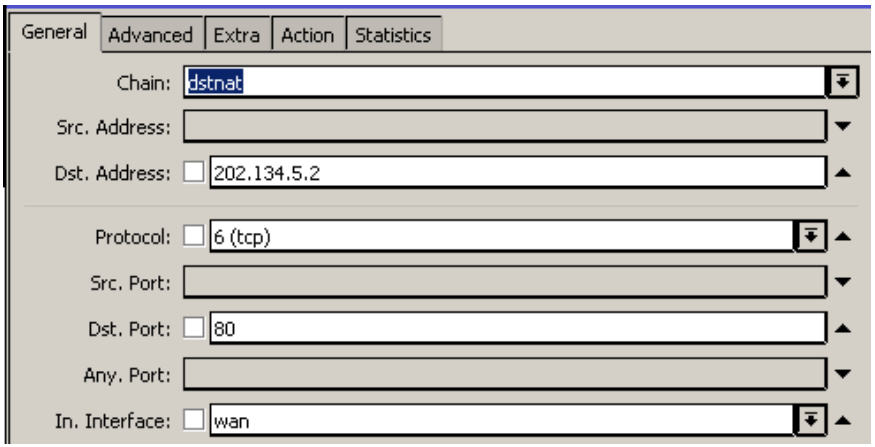
Mikrotik DMZ

DMZ adalah singkatan untuk *Dimilitarized Zone,* istilah berasal dari penggunaan militer, yang berarti daerah penyangga antara dua musuh. Bila diterapkan didalam *network* artinya komputer atau *subnetwork* kecil yang berada di antara jaringan internal yang terpercaya. DMZ dapat dibuat menggunakan Mikrotik. Biasanya DMZ berisi perangkat yang dapat diakses dari internet seperti web (http) server, FTP server, SMTP server dan DNS server. Dalam contoh kali ini akan dibuat sebuah DMZ dari jaringan lokal yang memiliki *service* web server yang akan dapat diakses dari internet menggunakan IP *public router.* Lihat gambar 3.22



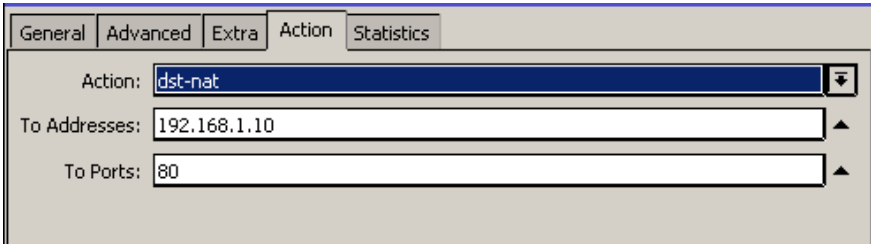
Gambar 3.22 Topologi DMZ

Dari gambar di atas pada *router* memiliki IP *public* 202.134.4.2/29 yang akan diakses dari internet kemudian akan di*forward* ke web server. Sehingga web server dalam jaringan lokal. Untuk konfigurasi di Mikrotik adalah sebagai berikut. Lihat gambar 3.23 untuk aturan setiap request http (port 80) ke IP *public* 202.134.5.2.



Gambar 3.23 dst-nat *general*

Kemduian setiap request ke IP 202.134.5.2 dengan port 80 (web) akan diteruskan ke komputer dengan IP 192.168.1.10 dengan port 80 yang memiliki service web server sebenarnya. Lihat gambar 3.24



Gambar 3.24 dst-nat *action*

BAB III Proxy MikroTik

Proxy

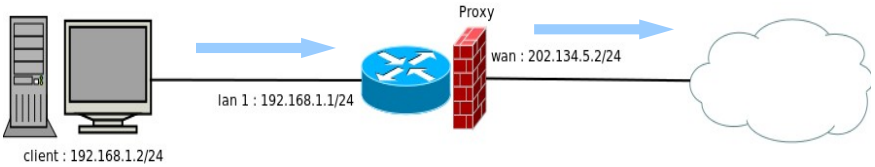
Salah satu fungsi *proxy* adalah menyimpan *cache*. Apabila sebuah LAN menggunakan *proxy* untuk berhubungan dengan internet, maka yang dilakukan oleh *browser* ketika user mengakses sebuah URL adalah mengambil *request* tersebut ke di server *proxy*. Sedangkan jika data belum terdapat di server *proxy* maka *proxy* akan mengambil dulu dari web server. Kemudian *request* tersebut disimpan di *cache server proxy.* Selanjutnya jika ada *client* yang melakukan *request* ke URL yang sama, maka *request* akan diambilkan dari *cache* server *proxy*. Teknik ini akan membuat akses ke internet lebih cepat.

Pada dasarnya *web proxy* terdapat 2 tipe :

Nontransparent web proxy

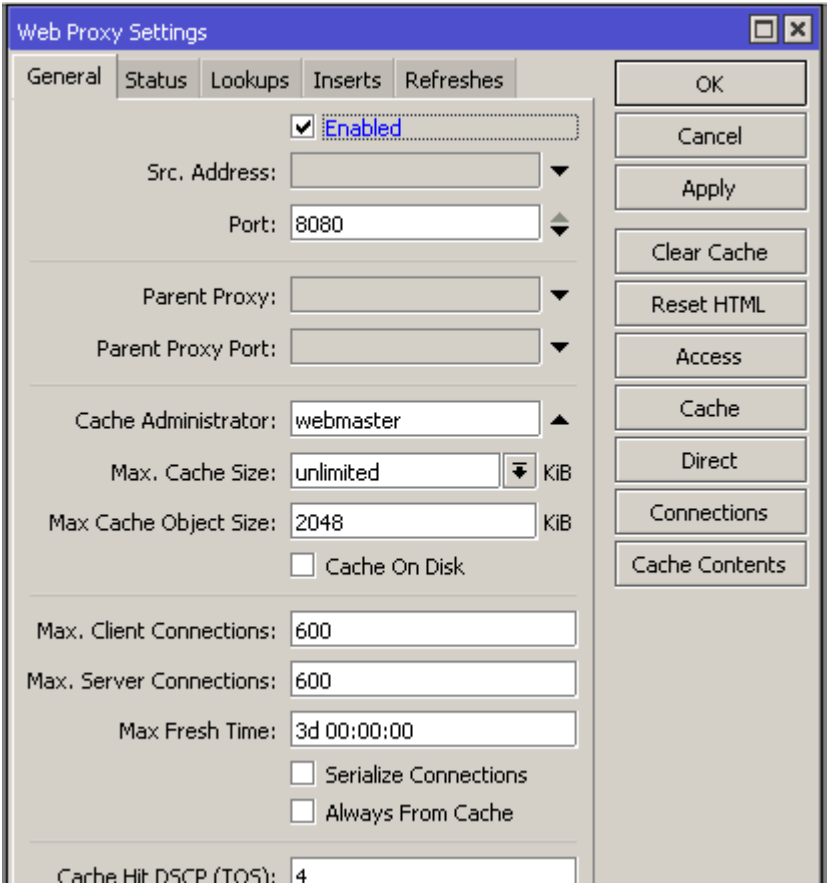
Transparent web proxy

Nontransparent Proxy



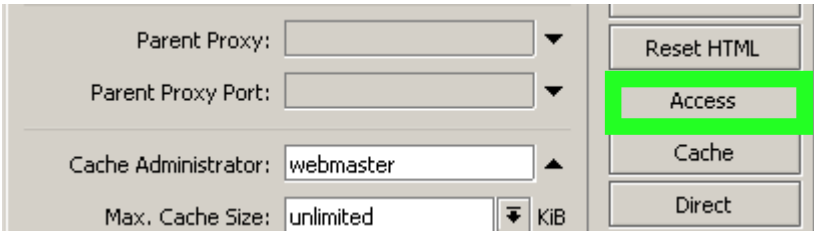
Gambar 3.1 Diagram blok proxy

Berikut adalah langkah reguler *proxy* Mikrotik. Dengan menggunakan Winbox, klik IP > *Web Proxy.* Lihat gambar 3.2



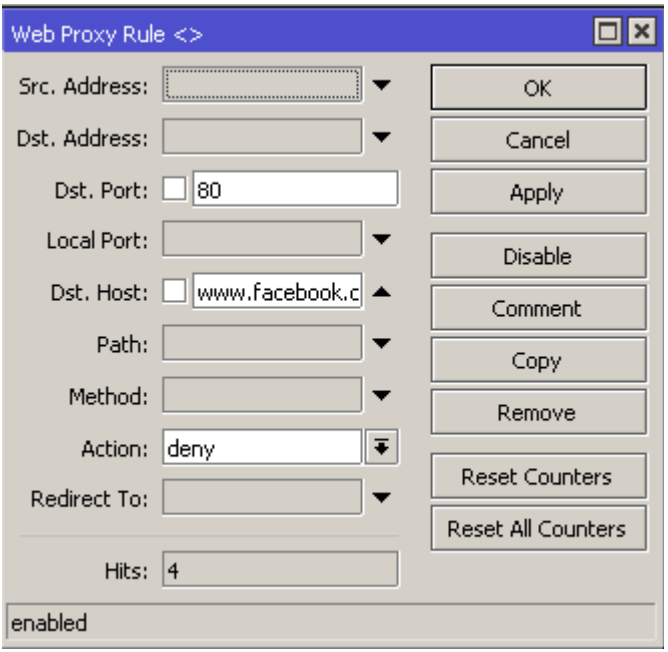
Gambar 3.2 IP *web proxy*

Misalkan kita ingin menetapkan beberapa akses *website* ditolak oleh Mikrotik menggunakan *web proxy* kita dapat klik pada tombol *Access.* Lihat gambar 3.3



Gambar 3.3 *Access web proxy*

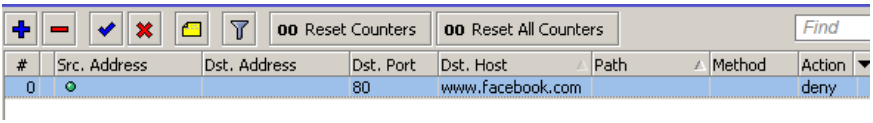
Kemudian tambahkan rule *access* dengan klik *add* atau tanda '+'. Sebagai contoh kita akan menghentikan akses menuju *facebook* menggunakan *web proxy*. Maka untuk aturannya adalah seperti pada gambar 3.4 di bawah ini.



Gambar 3.4 *Web proxy rule*

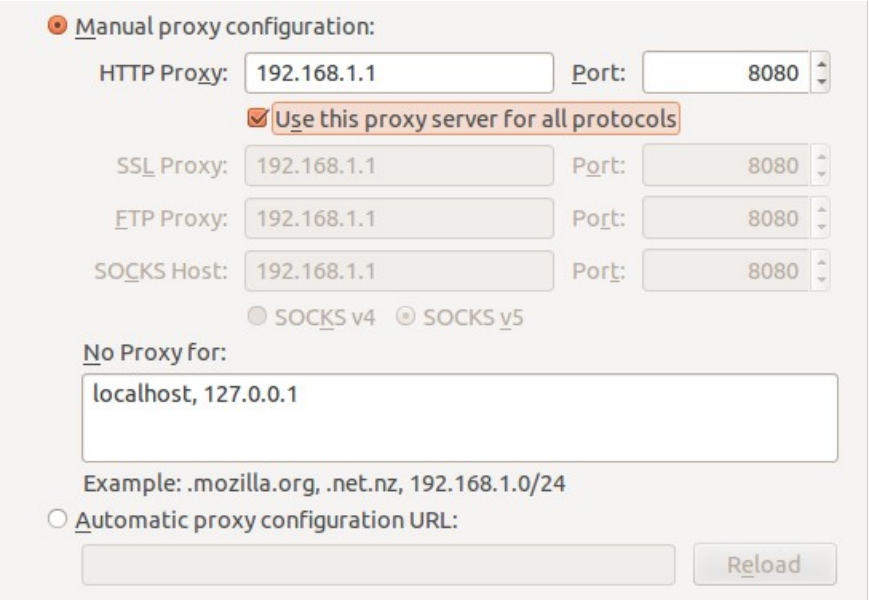
Maksud dari *rule* seperti pada gambar di atas adalah dengan tujuan port web (http) yaitu 80 pada www.facebook.com akan dilakukan *action* oleh *proxy* dengan aksi *deny*.

Lihat hasil list akan muncul seperti gambar 3.5 di bawah ini.



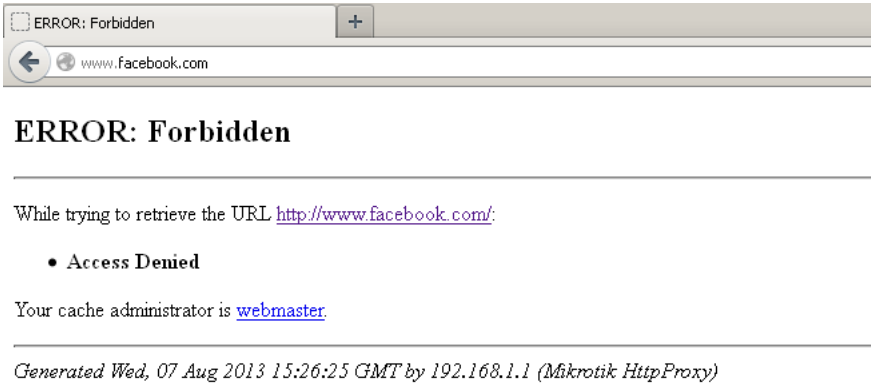
Gambar 3.5 *Web proxy list*

Keluar jendela kemudian klik OK pada halaman *web proxy.* Karena dalam contoh ini menggunakan *nontransparent web proxy* dimana *client* harus di *setting* terlebih dahulu, contoh pada gambar 3.6 di bawah ini adalah menggunakan *browser* Firefox.



Gambar 3.6 *Proxy client*

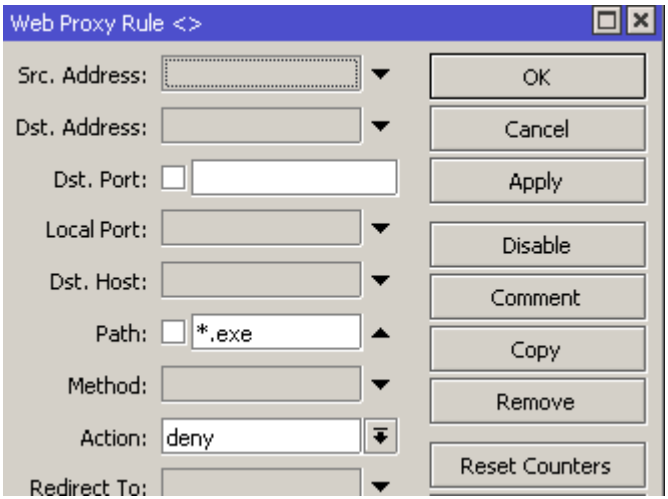
Untuk melakukan *check* konfigurasi cobalah akses *facebook* dari *client* pastikan akan mendapatkan pesan error pada halaman browser seperti gambar 3.7 di bawah ini.



Gambar 3.7 *Error proxy*

Blokir Download dengan MikroTik

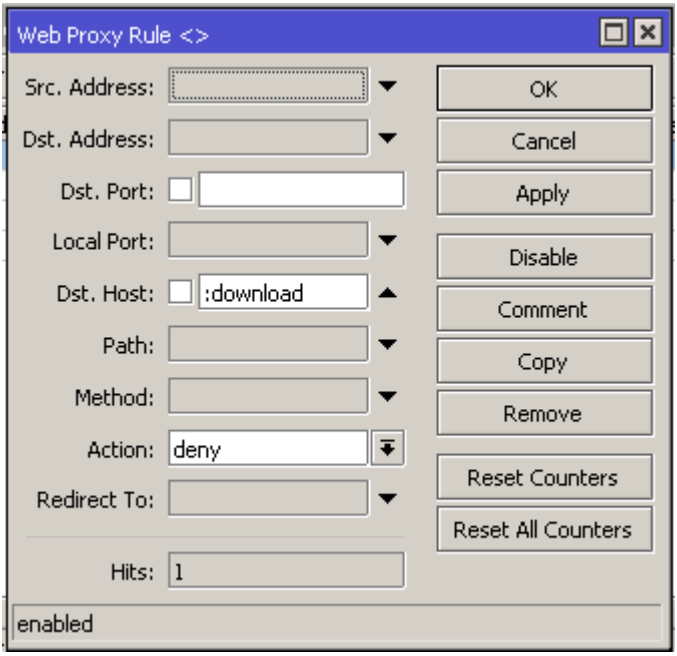
Dengan menggunakan *proxy* Mirkrotik selain memblok suatu halaman *website* kita juga dapat memblok apabila ada *client* yang akan melakukan *download* suatu file dengan format tertentu. Contoh kali ini kita kan memblok untuk melakukan *download* file dengan format .exe. Hal itu dapat kita konfigurasi dengan cara klik pada menu Winbox IP > *Web Proxy* > *klik button Access.*



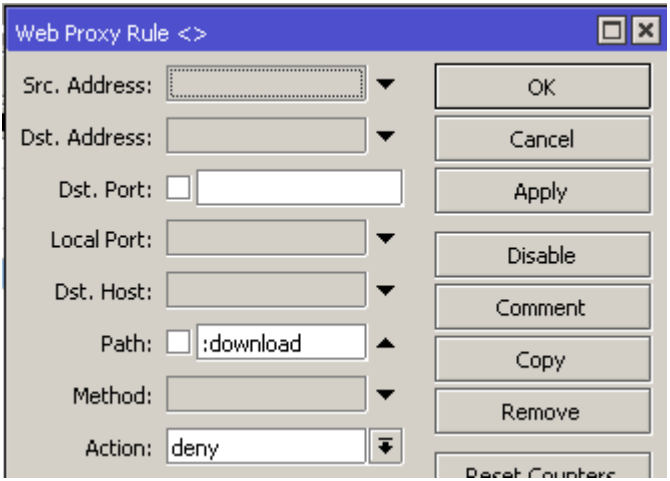
Block download

Block by Word

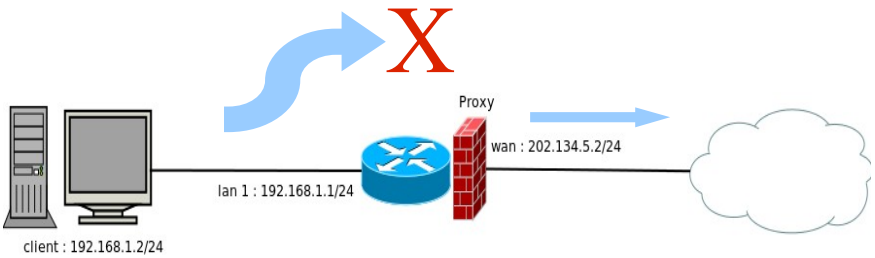
Block by Word istilah ini menjelaskan suatu teknik membloking menggunakan *proxy* berdasarkan kata, misalkan dalam contoh kali ini kita akan memblok suatu akses internet dengan alamat *website* yang mengandung kata **download**. Hal itu dapat kita lakukan dengan cara klik menu Winbox IP > *Web Proxy* > *klik button Access.*



Dan apabila kita ingin memblok segala akses internet yang menggunakan kata download dapat kita gunakan konfigurasi seperti berikut.

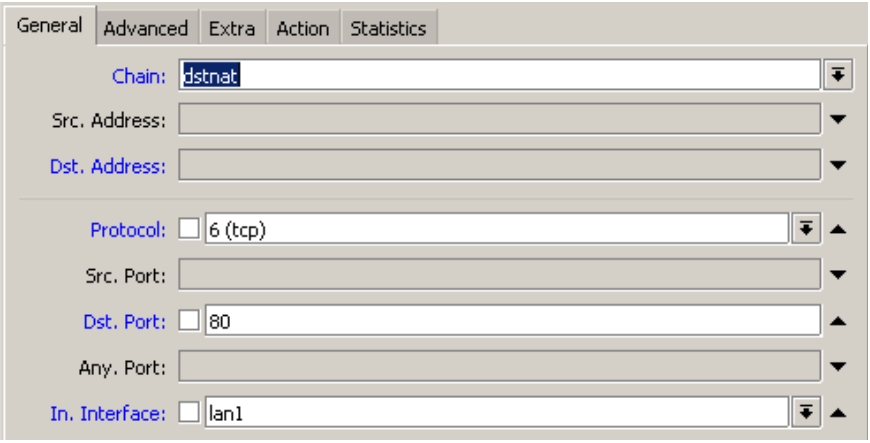


Transparent Proxy

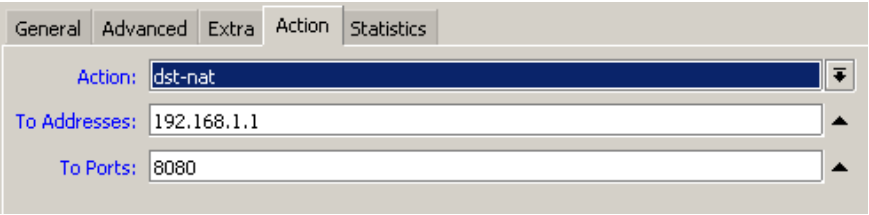


Gambar 3.8 *Transparent proxy*

Transparent proxy digunakan untuk memaksa *client* menggunakan *rule proxy* yang telah ditetapkan di Mikrotik. Sehingga tidak perlu setting pada browser pada *client* karena secara otomatis akan diarahkan oleh router. Dengan menggunakan konfigurasi *proxy* pada sub bab sebelumnya kemudian dapat ditambahkan dengan membuat rule pada *Firewall* NAT seperti pada gambar 3.9 di bawah ini.



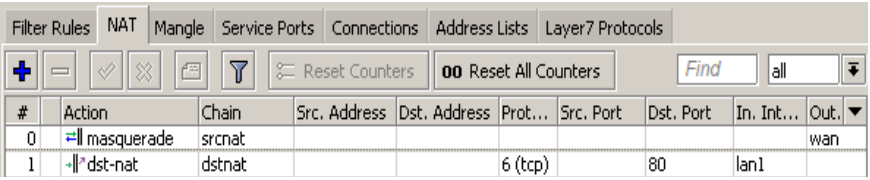
(a)



(b)

Gambar 3.9 (a) *General firewall* NAT *transparent proxy* (b) *Action firewall* NAT *transparent proxy*

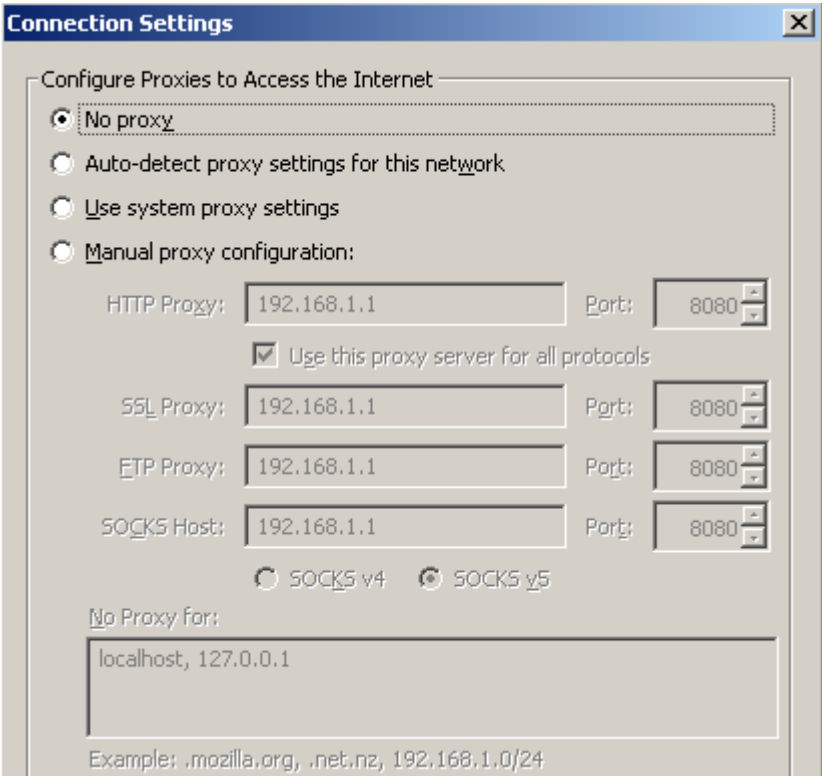
Sehingga pada list *firewall* NAT akan muncul satu buah rule seperti pada gambar 3.10 dibawah ini.



Gambar 3.10 *List firewall* NAT

Setelah konfigurasi di atas dilakukan cobalah sekarang setting kembalikan pada *proxy client* ke posisi *no proxy*. Kemudian akses

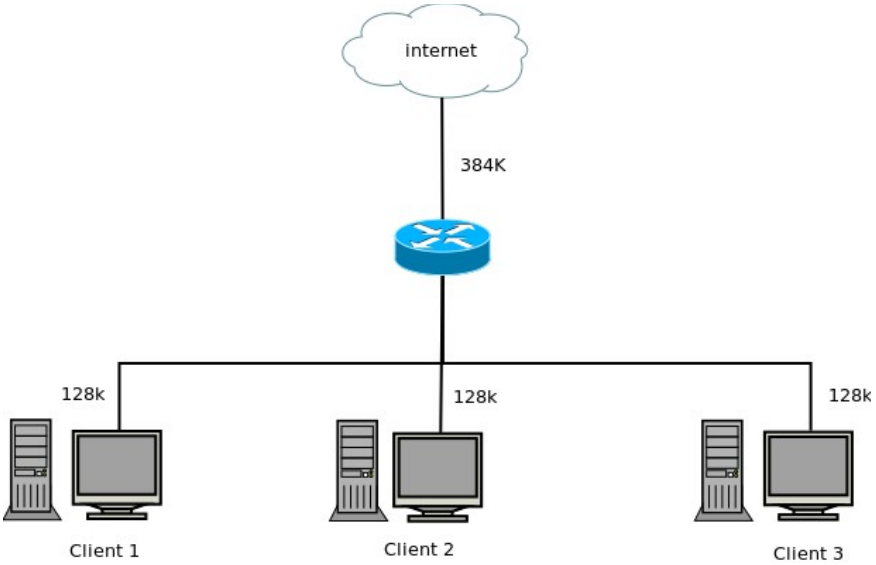
facebook kembali. Seharusnya koneksi akan tetap ditolak oleh proxy meskipun pada *proxy client* tidak diarahkan ke proxy Mikrotik.



Gambar 3.11 *No proxy*

BAB V Quality of Service (QoS)

Bandwidth Management



Gambar 5.1 *Bandwidth management*

Bandwidth management merupakan teknik QoS dari Mikrotik sebagai internet *gateway.* Salah satu tekniknya adalah dengan menggunakan *queue* yaitu sistem antrian yang berfungsi manajemen *bandwidth* yang digunakan oleh jaringan lokal. Dalam bab ini akan di bahas 3 teknik *queue*, yaitu :

Simple Queue
Queue Tree
Burst
PCQ *(Per Connection Queue)*

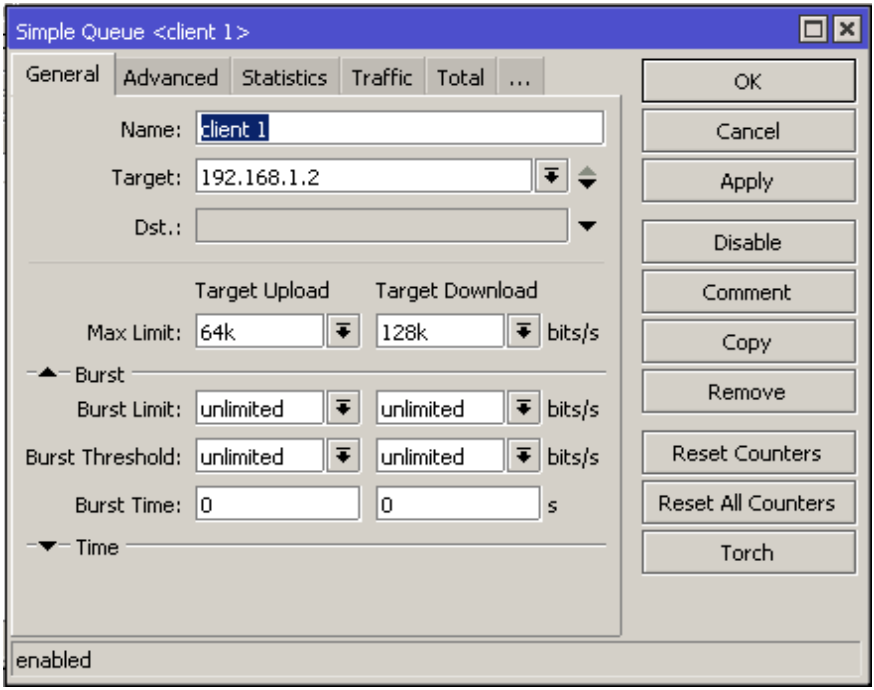
Simple Queue

Cara membagi *bandwidth* secara sederhana adalah menggunakan *simple queue.* Pada pembahasan kali ini kita ambil contoh ada suatu jaringan lokal dengan 3 PC, kemudian jaringan tersebut memiliki *bandwidth* 384Kbps *download* dan 192Kbps *upload* sebagai koneksi ke internet yang akan diakses oleh ke 3 PC lokal. Untuk itu sebagai seorang *administrator* jaringan kita akan membagi *bandwidth* secara rata. Untuk itu dapat kita perhitungkan untuk masing-masing komputer mendapatkan *bandwidth* untuk *download* = 128Kbps dan *upload* = 64Kbps. Bila disajikan dalam tabel maka seperti berikut.

Tabel 5.1 *Data* IP dan *bandwidth*

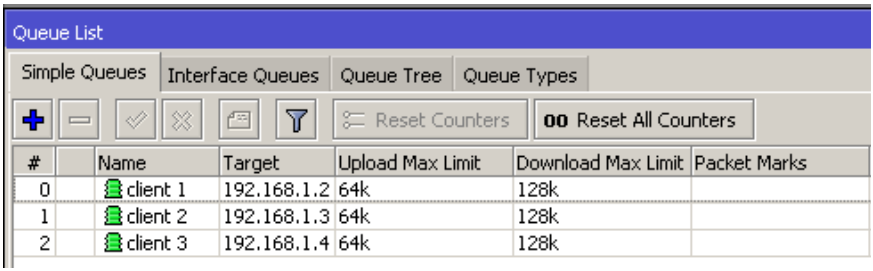
No	Nama	IP	*Up*	*Down*
1.	Client 1	192.168.1.2	64 Kbps = 8 KBps	128 Kbps = 16 KBps
2.	Client 2	192.168.1.3	64 Kbps = 8 KBps	128 Kbps = 16 KBps
3.	Client 3	192.168.1.4	64 Kbps = 8 KBps	128 Kbps = 16 KBps

Untuk melakukan setting menggunakan Winbox klik pada menu *Queue.* Kemudian pada tab *simple queues* tambahkan rule seperti pada gambar 5.2 di bawah ini



Gambar 5.2 Seting *simple queue*

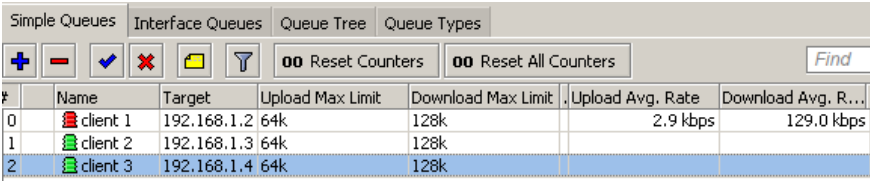
Tambahkan *simple queue* untuk komputer client 1 dan client 2, sehingga di lihat dari hasil konfigurasinya adalah seperti gambar 5.3 berikut.



Gambar 5.3 *Simple queue*

Setelah selesai konfigurasi untuk melakukan uji coba bisa kita lakukan *download* sebuah file untuk menguji berapa kecepatan yang didapat.

Gambar 5.4 *Download simple queue (client)*



Gambar 5.5 *Monitoring simple queue*

Dari gambar diatas dilihat bahwa rata-rata kecepatan download pada sisi client adalah 15.5 KB/s diperoleh dari kecepatan yang termonitoring di mikrotik (129 kbps) dibagi dengan 8.

Queue Tree

Cara membagi *bandwidth* yang lebih kompleks adalah dengan menggunakan *queue tree*. Cara ini digunakan untuk membagi *bandwidth* berdasarkan *protocol* dan *port*.

Untuk konfigurasi *queue tree* pertama yang harus dilakukan adalah buat *mark* terlebih dahulu untuk menandai paket data yang melalui suatu *queue*. Untuk itu lakukan langkah-langkah sebagai berikut :

Mark connection berdasarkan IP address.

Mark packet berdasarkan *connection* di atas.

Kali ini kita kembali menggunakan contoh kasus pada pembahasan *simple queue* sebelumnya.

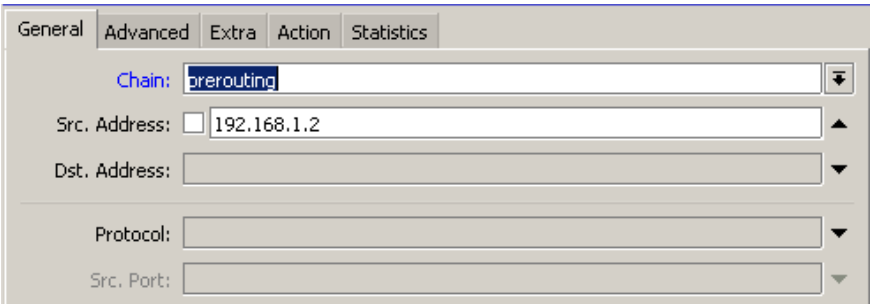
Lakukan pengaturan *mangle* seperti berikut

Chain : *prerouting*

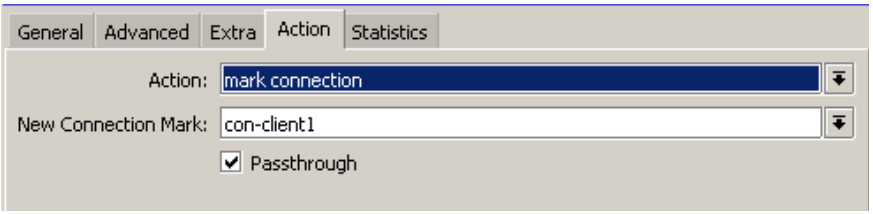
Src. Address : 192.168.1.2

Action : *mark connection*

New Connection Mark : con-client1



(a)



(b)

Gambar 5.6 (a) *Mangle rule tab general (mark connection)* (b) Mangle rule tab action (*mark connection*)

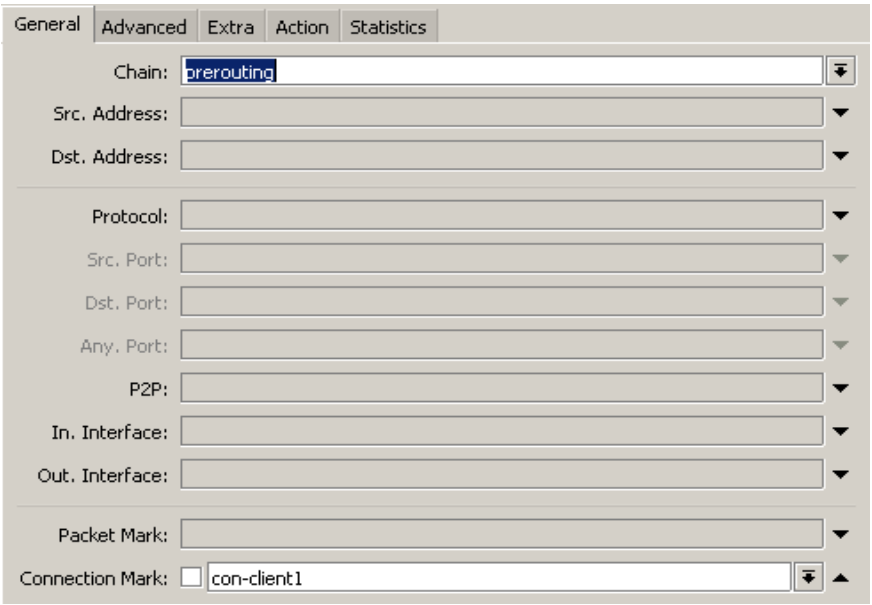
Lakukan juga pada komputer client2 dan client3. Langkah berikutnya adalah melakukan *marking* terhadapat paket.

Chain : *prerouting*

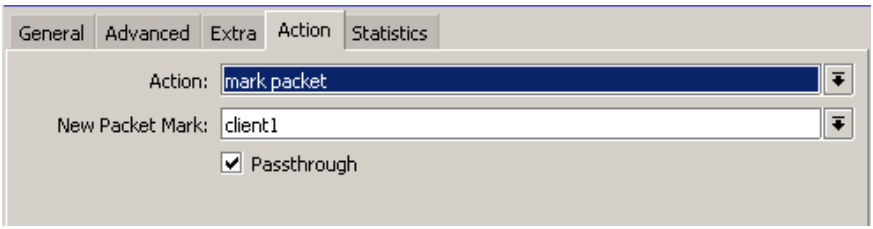
Connection Mark : con-client

Action : *mark packet*

New Packet Mark : client1



(a)



(b)

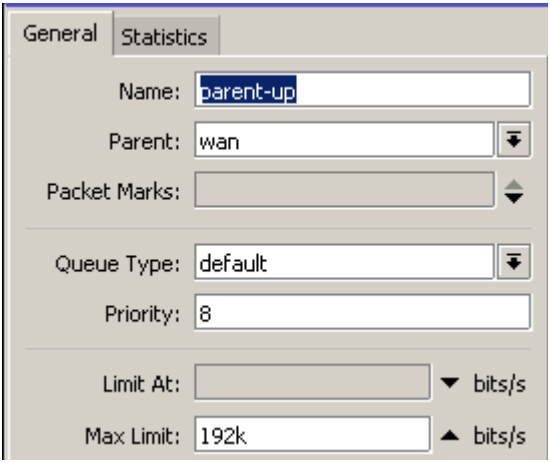
Gambar 5.7 (a) *Mangle rule tab general (mark packet)* (b) *Mangle rule tab action (mark packet)*

Lakukan juga *mark packet* pada komputer client1 dan client2. Berikutnya adalah memasukkan paket yang sebelumnya telah dilakukan *marking* ke *queue tree*.

Buat *parent* terlebih dahulu sebagai induk dari semua cabang *queue* yang akan dibuat, yaitu *parent-up* dan *parent-down*.

Klik pada menu *Queue* pada Winbox, kemudian tambahkan *rule* dengan klik pada tanda '+' di *tab queue trees*. Berikut adalah pengaturannya.

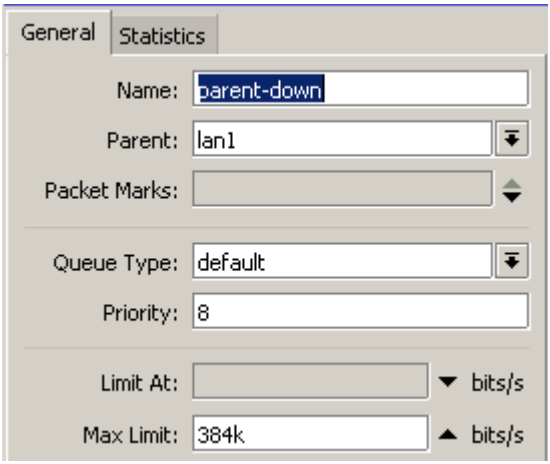
Name : parent-up
Parent : *wan*
Max limit : 192k



Gambar 5.8 *Add parent-up*

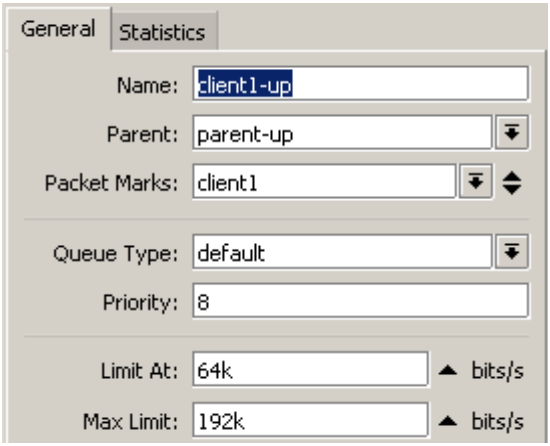
pada gambar diatas kolom parent diisikan dengan *out interface* yaitu *interface wan*. Karena trafik *upload* menuju jaringan luar. Sedangkan pada *parent-down* menggunakan *in interface* yaitu *interface .lan1* Konfigurasi *parent-down* adalah sebagai berikut.

Name : parent-down
Parent : *wan*
Max limit : 384k

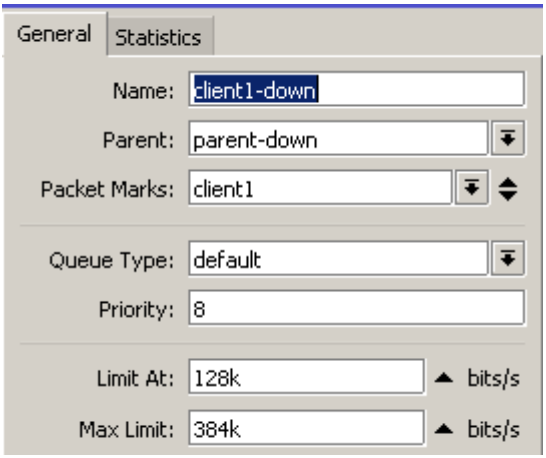


Gambar 5.9 *Add parent-down*

Langkah berikutnya adalah membuat *queue* untuk masing-masing komputer lokal. Klik pada menu *queue* di Winbox kemudian tambahkan *rule* dengan klik pada tanda '+'. Sebagai contoh di bawah ini adalah untuk *up* dan *down* pada komputer *client1*.

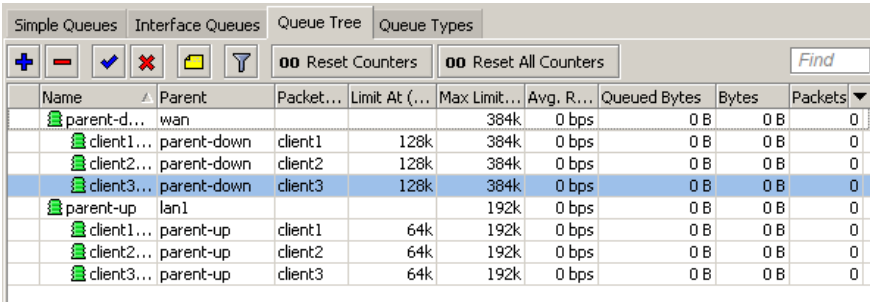


Gambar 5.10 *Up client* 1



Gambar 5.11 *Down client* 2

Tambahkan konfigurasi pada komputer client 2 dan client 3. Sehingga bila sudah ditambahkan semua konfigurasi akan menghasilkan tabel *queue* sepert berikut.



Name	Parent	Packet...	Limit At (...	Max Limit...	Avg. R...	Queued Bytes	Bytes	Packets
parent-d...	wan			384k	0 bps	0 B	0 B	0
client1...	parent-down	client1	128k	384k	0 bps	0 B	0 B	0
client2...	parent-down	client2	128k	384k	0 bps	0 B	0 B	0
client3...	parent-down	client3	128k	384k	0 bps	0 B	0 B	0
parent-up	lan1			192k	0 bps	0 B	0 B	0
client1...	parent-up	client1	64k	192k	0 bps	0 B	0 B	0
client2...	parent-up	client2	64k	192k	0 bps	0 B	0 B	0
client3...	parent-up	client3	64k	192k	0 bps	0 B	0 B	0

Gambar 5.12 Tabel *queue*

Sebagai uji coba kali ini lakukan kembali *download* dari klien dan perhatikan kecepatan *download*.

Gambar 5.13 *Download rate (client)*

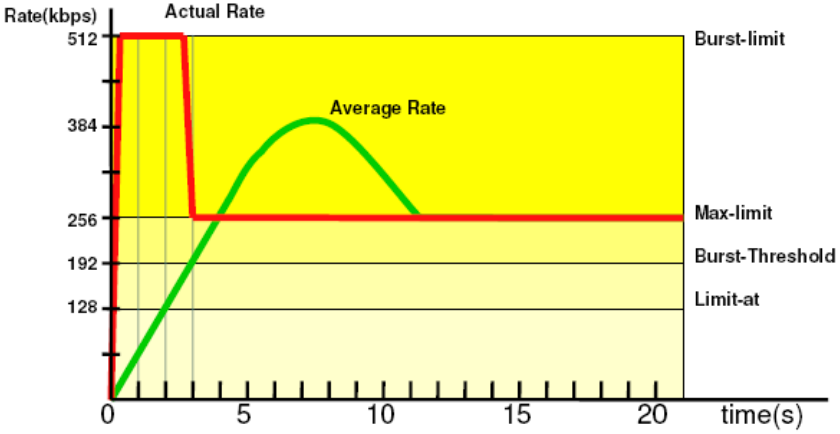
Name	Parent	Packet...	Limit At (...	Max Limit...	Avg. Rate
parent-down	lan1			384k	386.0 kbps
client1-down	parent-down	client1	128k	384k	419.3 kbps
client2-down	parent-down	client2	128k	384k	0 bps
client3-down	parent-down	client3	128k	384k	0 bps
parent-up	wan			192k	10.8 kbps
client1-up	parent-up	client1	64k	192k	10.8 kbps
client2-up	parent-up	client2	64k	192k	0 bps
client3-up	parent-up	client3	64k	192k	0 bps

Gambar 5.14 *Monitoring queue tree*

Dari gambar di atas dilihat bahwa kecepatan rata-rata yang diperoleh pada saat *download* di sisi client adalah sebesar 45.9 KB/s yang diperoleh dari kecepetan yang termonitoring pada Mikrotik (419 kbps) dibagi dengan 8. Kecepatan yang diperoleh adalah mencapai *Max limit* karena koneksi hanya dipakai satu client yang memungkinkan *bandwidth* yang dipakai secara maksimal.

Burst Simple Queue

Burst adalah salah satu teknik untuk melakukan QoS. Dengan menggunakan *burst* memungkinkan *client* dapat mencapai *data-rate* melebihi *max-limit* untuk periode waktu tertentu. Jika *data-rate* rata-rata lebih kecil dari *burst-threshold, burst* dapat dilakukan hingga *data-rate* mencapai *burs-limit*.



Gambar 5.15 Diagram *burst*

Dalam contoh kali ini akan dibuat sebuah aturan dimana client dapat memperoleh *burst-limit* 512 kbps selama 5 detik sedangkan limit yang sebenarnya adalah 64 kbps kita tentukan waktu *burst* adalah 20 detik, dengan ketentuan tersebut kita dapat melakukan perhitungan seperti berikut.

Tabel 5.1 Perhitungan *burst*

Detik	Data rate rata-rata (kbps)	Hasil perhitungan (kbps)	Status
1	$\frac{1x512}{20}$	25.6	Burst dapat dilakukan

2	$\frac{2x512}{20}$	51.2	Burst dapat dilakukan
3	$\frac{3x512}{20}$	76.8	Burst dapat dilakukan
4	$\frac{4x512}{20}$	102.4	Burst dapat dilakukan
5	$\frac{5x512}{20}$	128	Burst dapat dilakukan
6	$\frac{6x512}{20}$	153.6	Dari detik 6-20 Burst tidak dapat dilakukan

Dari tabel diatas adalah suatu konsep bahwa *burts* dapat dilakukan selama 5 detik di setiap 20 detik. Maka dapat kita temukan bahwa *burst-threshold* adalah 128 kbps. Konfigurasi pada mikrotik adalah sebagai berikut sebagai contoh untuk satu *client* dengan ketentuan *bandwidth upload* dan *download* disamakan.

Name : client 1

Target : 192.168.1.2

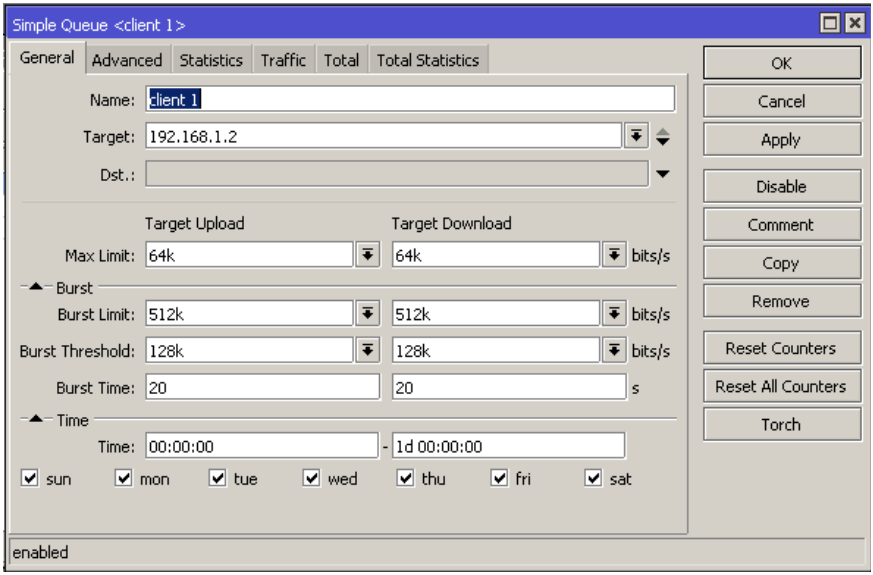
Limit At : 64 kbps

Max Limit : 64 kbps

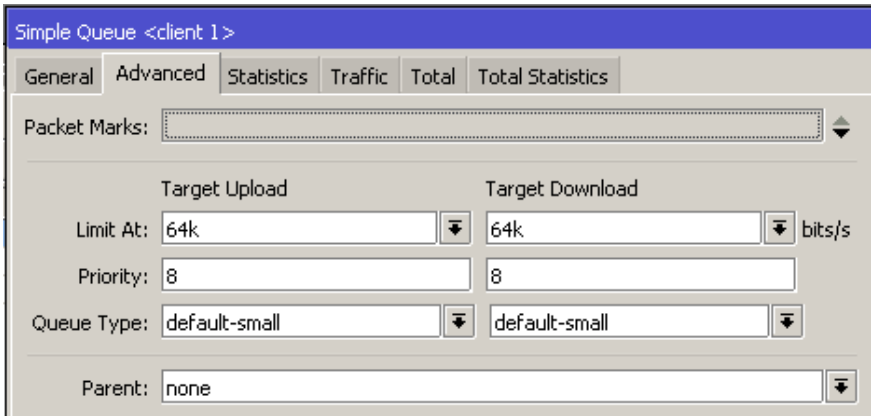
Burst-limit : 512 kbps

Burst-threshold : 128 kbps

Burst-time : 20 s

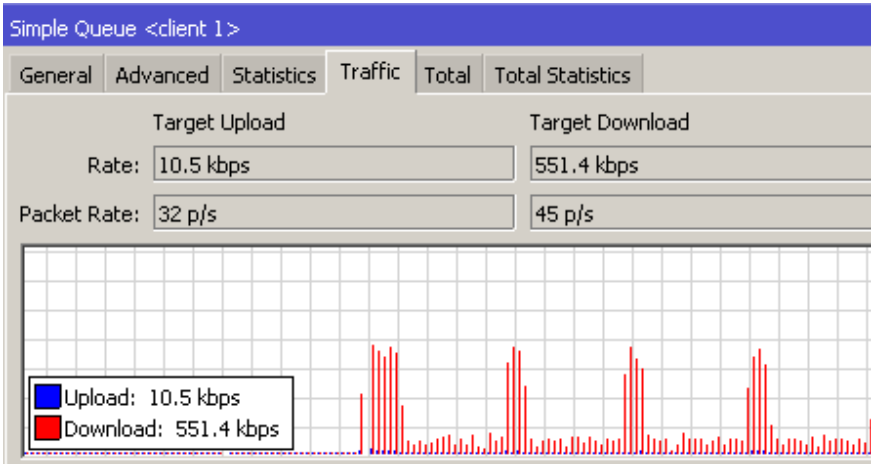


Gambar 5.16 *Burst simple queue (tab general)*



Gambar 5.17 *Burst simple queue (tab advanced)*

Kemudian untuk melakukan pengujian cobalah akses browsing atau download dari *client* amati kecepatan *download* dan monitoring *traffic bandwidth* di Mikrotik.



Gambar 5.18 *Traffic bandwidth burst simple queue*

Dari gambar di atas terlihat pada setiap 20 detik terjadi burst pada rata-rata data rate 512 kbps selama 5 detik dan 15 detik sisanya adalah *limit bandwidth*.

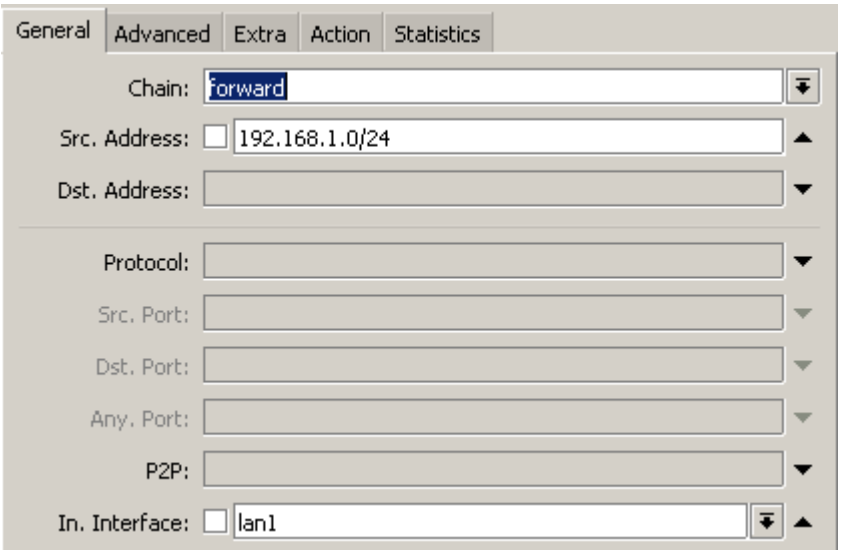
Per Connection Queue (PCQ)

Cara yang lebih mudah adalah membagi *bandwidth* dengan sama rata tanpa mendefinisikan komputer A harus diberik sekian kbps. Dengan cara ini, kita cukup mendefinisikan angka nominal *up* dan *down* yang didapat dari ISP, kemudian Mikrotik akan membagi ke seluruh *client*.

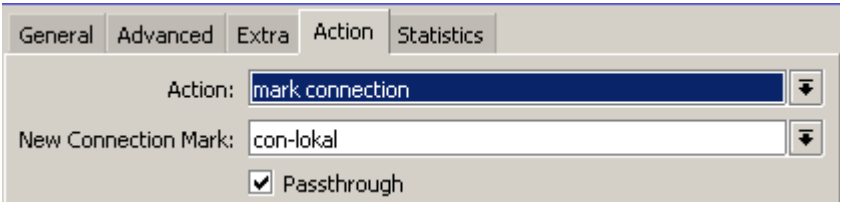
Dalam hal ini, kita akan menggunakan *queue* dengan jenis PCQ. Langkah pertama adalah melakukan *marking packet* terhadap semua paket yang datang dari *network client*. *Network client* di sini adalah 192.168.1.0/24 dengan ketentuan limitasi *bandwidth* adalah 64 kbps dan *max-limit bandwidth* 128 kbps.

Lakukan *marking connection* terlebih dahulu menggunakan Winbox menu IP > *Firewall* > *tab Mangle* dengan *rule* seperti berikut :

Chain : *Forward*
Src. Address : 192.168.1.0/24
In. Interface : lan1
Action : *mark connection*
New Connection Mark : con-lokal



Gambar 5.19 PCQ *mangle mark connection (tab general)*



Gambar 5.20 PCQ *mangle mark connection (tab action)*

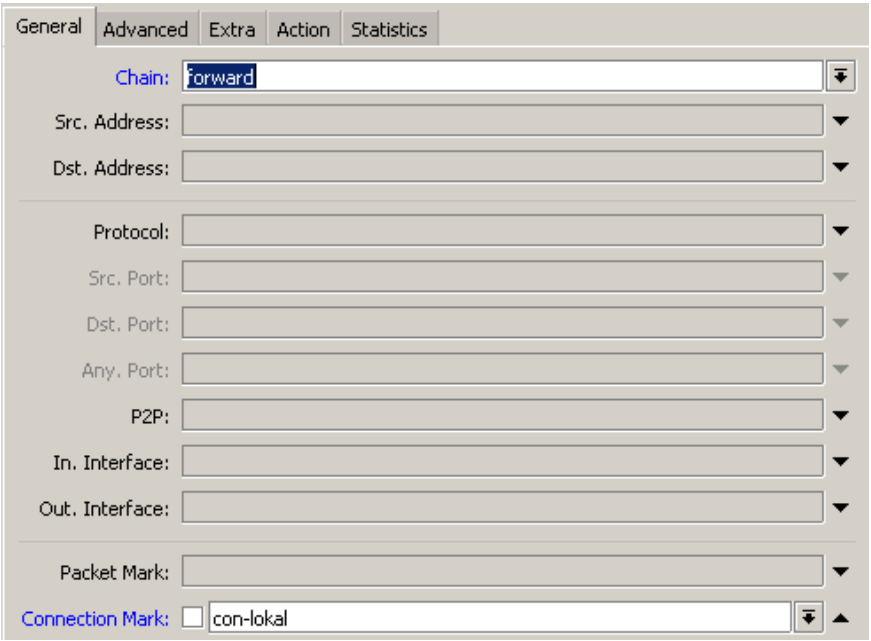
Setelah membuat *mangle* untuk *marking connection* langkah berikutnya adalah membuat *mangle* untuk melakukan *marking packet* dengan konfigurasi seperti berikut :

Chain : *Forward*

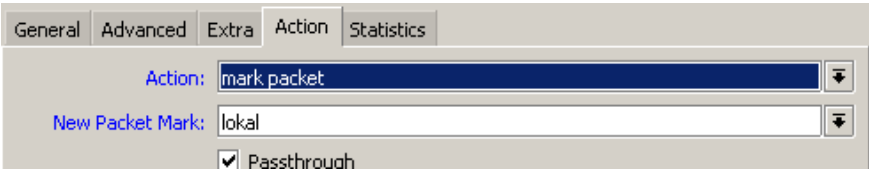
Connection Mark : con-lokal

Action : *mark packet*

New Packet Mark : lokal



Gambar 5.21 PCQ *mangle mark packet (tab general)*



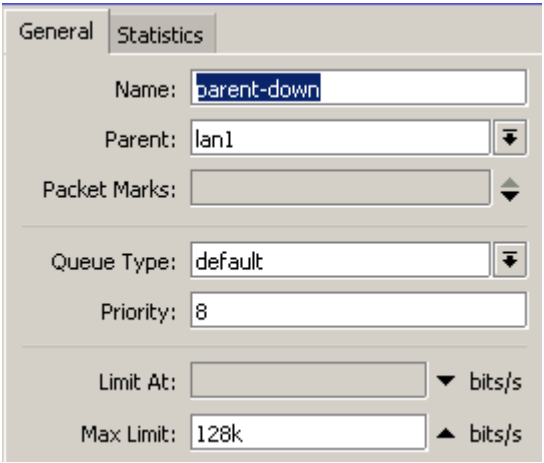
Gambar 5.22 PCQ *mangle mark packet (tab action)*

Langkah berikutnya adalah menambahkan queue *parent-down* dan *parent-up* dengan cara klik menu *Queue* > *tab Queue Tree.*

Name : *parent-down*

Parent : lan1

Max Limit : 128k

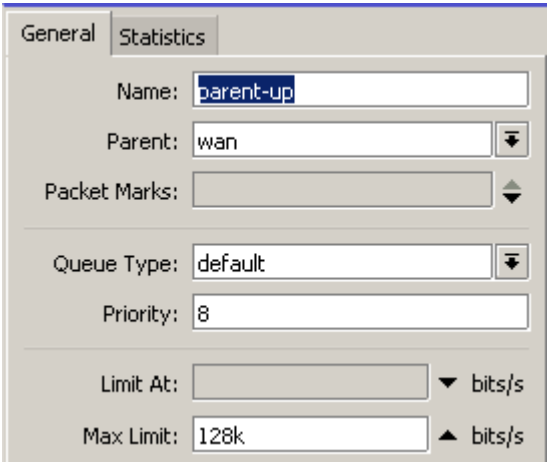


Gambar 5.23 *Parent-down* PCQ

Name : *parent-up*

Parent : wan

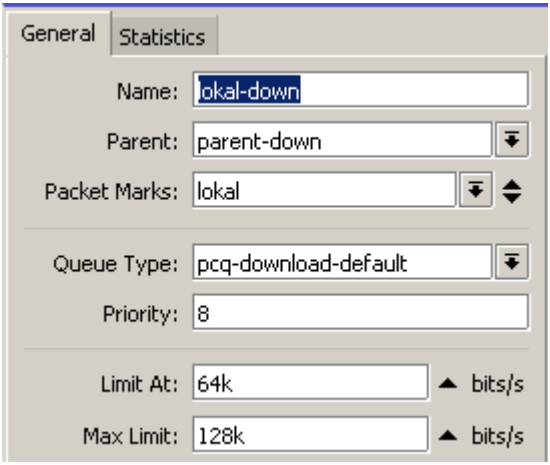
Max-limit : 128k



Gambar 5.24 *Parent-up* PCQ

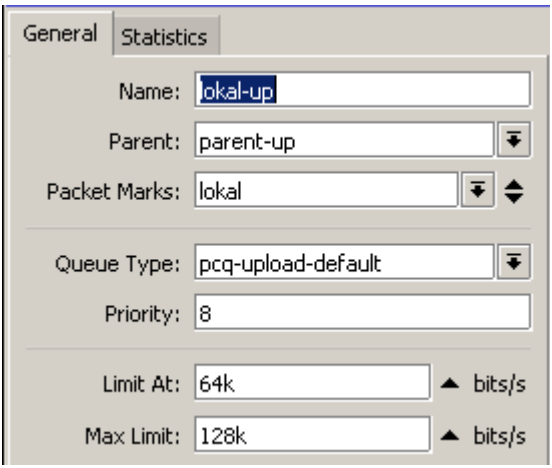
Setelah *parent* dibuat maka selanjutnya adalah menambahkan *queue tree* untuk network lokal baik untuk koneksi *up* maupun *down.* Lakukan dengan cara klik menu *Queue* > *tab Queue Tree.*

Name	: *lokal-down*
Parent	: *parent-down*
Packet Marks	: *lokal*
Queue Type	: *pcq-download-default*
Limit At	: 64k
Max Limit	: 128k



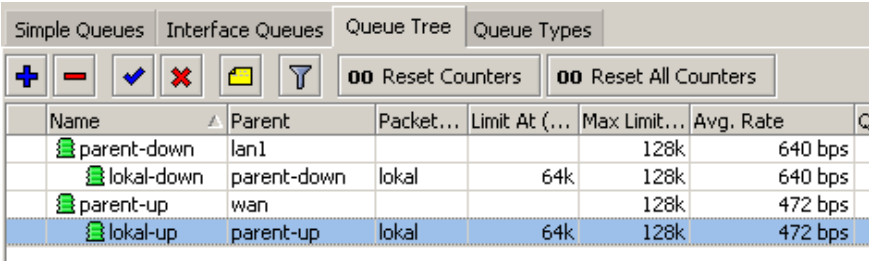
Gambar 5.24 *Lokal down* PCQ

Name : *lokal-up*
Parent : *parent-up*
Packet Marks : *lokal*
Queue Type : *pcq-upload-default*
Limit At : 64k
Max Limit : 128k



Gambar 5.25 *Lokal up* PCQ

Hasil tabel *queue* setelah melakukan konfigurasi adalah seperti gambar di bawah ini.



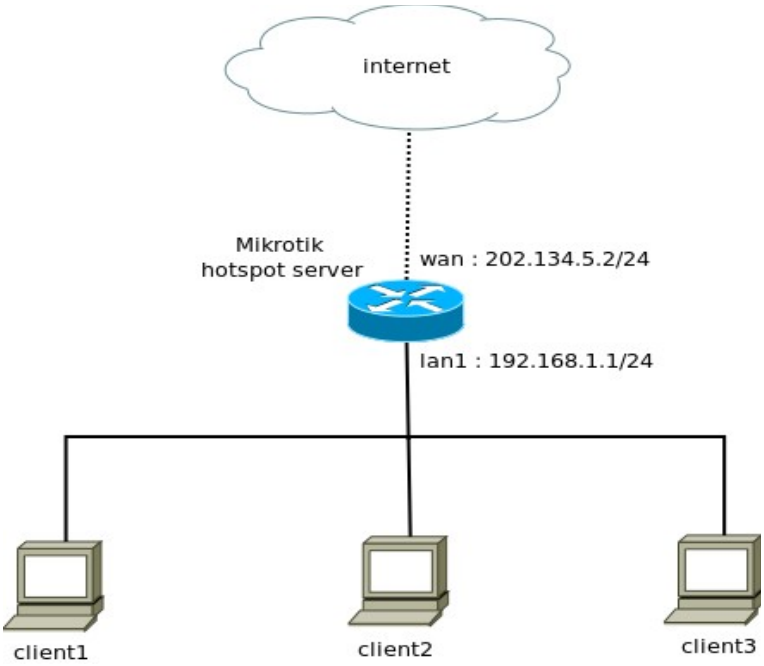
Gambar 5.26 Tabel *queue*

Selanjutnya lakukan uji koneksi dengan cara cobalah download sebuah file dari jaringan lokal komputer 1 dan komputer 2.

BAB 6 Hotspot Mikrotik

Hotspot

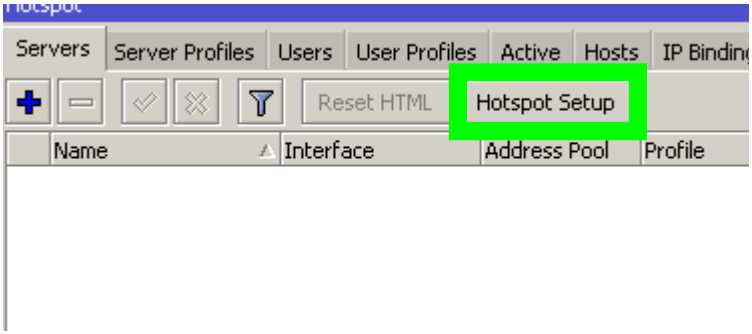
Hotspot digunkan untuk memberikan layanan akses internet di area publik dengan media kabel maupun *wireless*. Ketika *user client* membuka halman web maka *router* akan memeriksa apakah *user* tersebut terotentikasi atau tidak. Jika tidak melakukan otentikasi, pengguna akan dialihkan ke halaman *login* hotspot yang memerlukan *username* dan *password*. Jika informasi login yang dimasukkan adalah benar, maka router akan mengijinkan *user* untuk mengakses internet. Pengguna akses internet dalam jaringan *hotspot* dapat dihitung/dibatasi berdasarkan waktu (*time-based*) dan data *download/upload* (*volume-based*).



Gambar 6.1 *Hotspot* Mikrotik

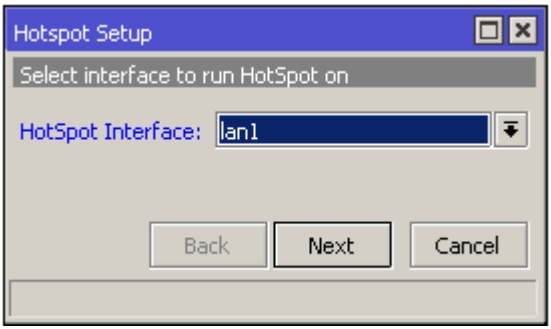
Konfigurasi Hotspot

Untuk konfigurasi *hotspot* di Mikrotik menggunakan Winbox sangatlah mudah dan tidak terlalu lama untuk membangun *hotspot* Mikrotik. Untuk melakukannya klik pada menu IP > *Hotspot* > *Hotspot setup*.



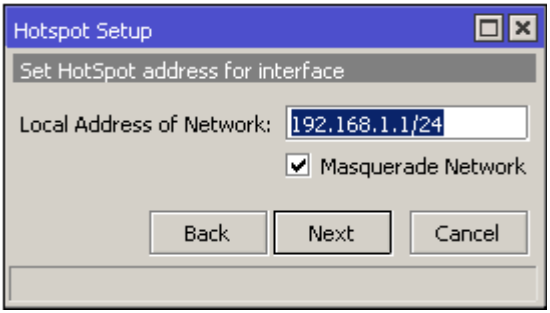
Gambar 6.2 *Hotspot setup*

Pertama, akan muncul form isian yang meminta *interface* yang digunakan dalam jaringan *hotspot*. Pada contoh kali ini adalah *interface* lan1



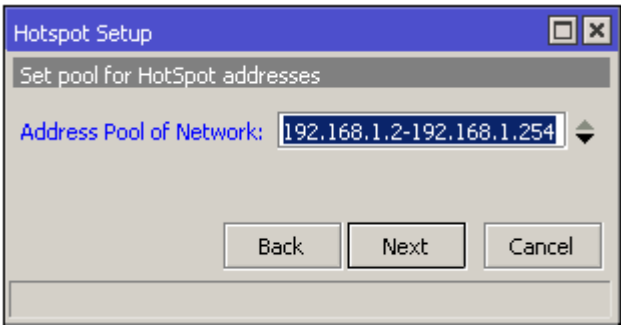
Gambar 6.3 *Hotspot interface*

Kedua, menentukan IP *address* di *interface* lan1 yang akan menjadi *gateway* dari jaringan *hotspot*. IP *address* tersebut adalah 192.168.1.1/24.



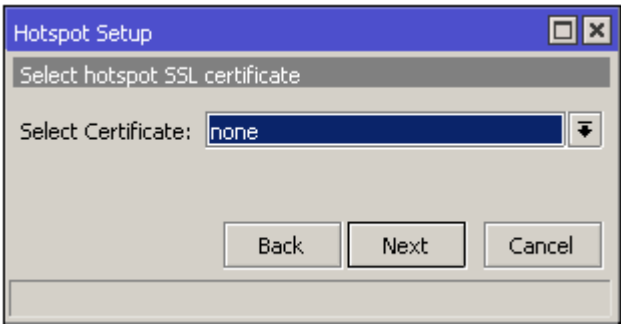
Gambar 6.4 *Local address hotspot*

Ketiga, muncul form yang harus diisi dengan alamat IP yang akan digunakan oleh *client* pada jaringan *hotspot*. Contoh kali ini adalah 192.168.1.2-192.168.1.254



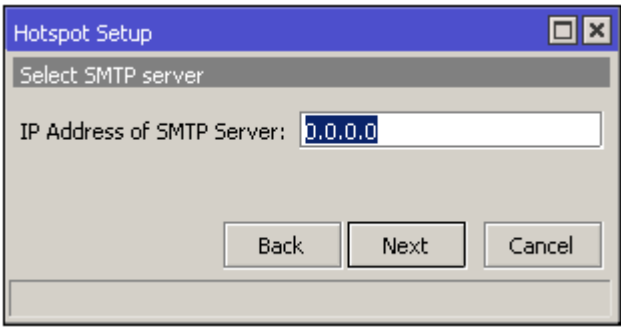
Gambar 6.5 *Address Pool of network*

Keempat, kita diminta untuk memilih *certificate,* pilih *none.*



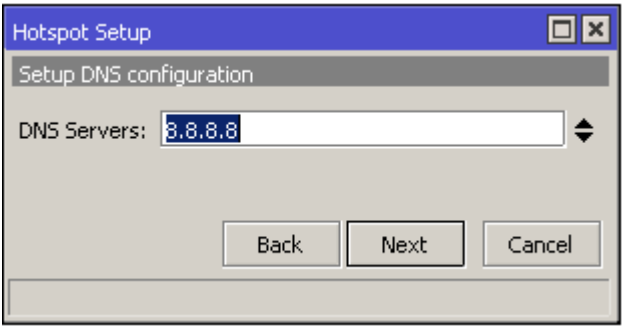
Gambar 6.6 *Select Certificate*

Kelima, kita akan diminta untuk mengisikan kolom IP *address* untuk SMTP server. Biarkan untuk IP SMTP server 0.0.0.0 atau isikan dengan SMTP jika ada.



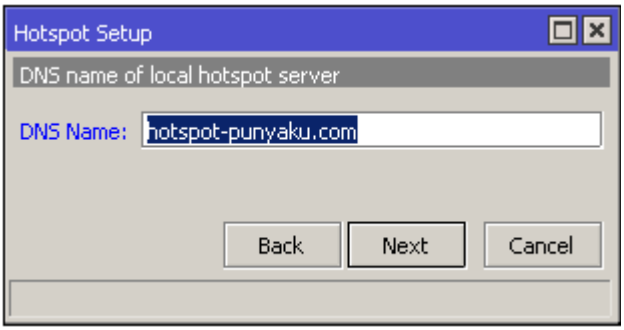
Gambar 6.7 IP *Address of* SMTP *Server*

Keenam, selanjutnya akan diminta untuk mengisi DNS server isikan dengan DNS server dari Mikrotik biasanya otomatis tersisikan.



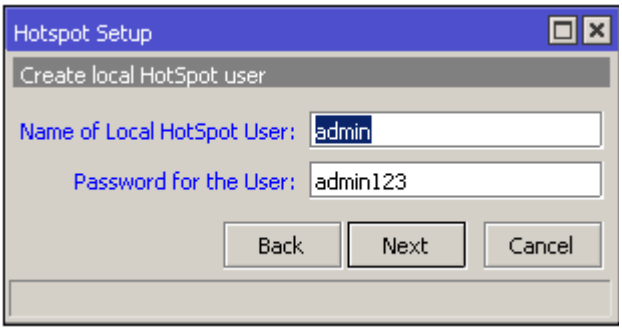
Gambar 6.8 DNS *Servers*

Ketujuh, kita akan diminta untuk mengisi kolom DNS *name*. Kita dapa mengisikan domain untuk *hotspot* sesuai keinginan kita, contoh kali ini adalah *hotspot-punyaku.com*



Gambar 6.9 DNS *name*

Terakhir adalah membuat satu *user* yang dapat digunakan pada otentikasi jaringan *hotspot*.

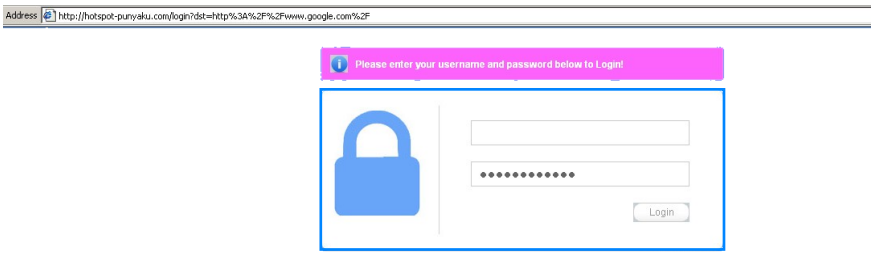


Gambar 6.10 *Hotspot user*

Selanjutnya adalah mengkonfigurasi profil dari server di menu IP > *Hotspot* > *tab Server Profiles*. Klik dua kali pada profil **hsprof1** makan muncul jendela *Hotspot Server Profile*. Klik pada tab *login* kemudian hilangkan tanda centang pada *Cookie*. Hal ini berguna untuk setiap kali *user* membuka browser akan dilakukan *login* setelah *user* tersebut melakukan *login hotspot* sebelumnya.

Gambar 6.11 *Hotspot Server Profile*

Kemudian cobalah akses internet dengan membuka browser Mozilla Firefox atau Internet Explorer, amati halaman apa yang akan pertama muncul ketika kita mencoba mengakses salah satu situs internet.



Gambar 6.12 *Login hotspot*

Bila dilihat gambar 6.12 diatas kita harus memasukkan *username* dan *password* yang telah ditambahkan pada Mikrotik. Setelah itu kita dapat mengakses situs yang kita inginkan. Untuk melihat status *user* dapat kita lihat di alamat domain sesuai dengan konfigurasi, contoh kali ini adalah *http://hotspot-punyaku.com/status*

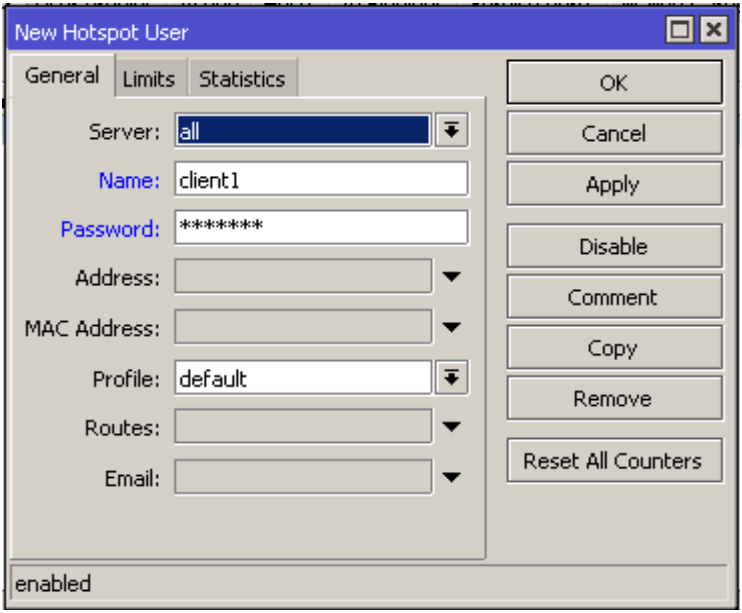
Gambar 6.13 *User status*

Kita dapat klik *log off* untuk keluar dari *user hotspot*.

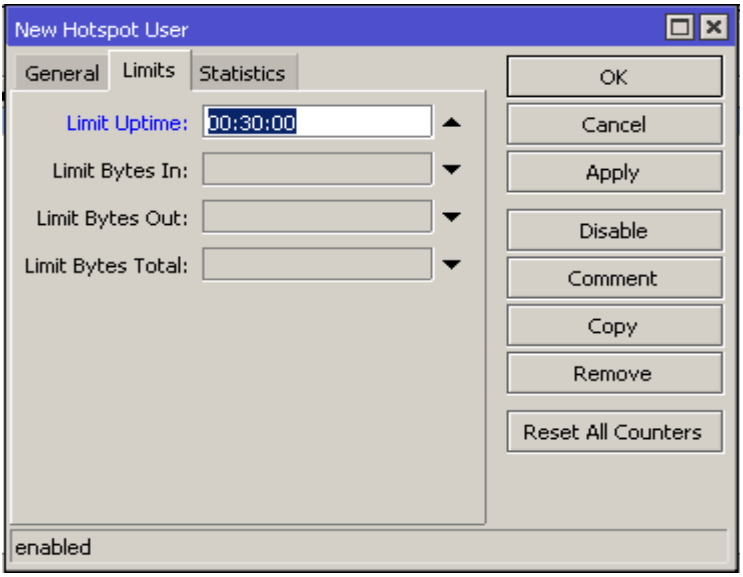
Time Based Charging

Dengan *hotspot* Mikrotik kita dapat melakukan pembatasan waktu akses untuk setiap *user*. Hal ini biasanya digunakan untuk pengelolaan Wi-Fi berbayar. Pada dasarnya ada dua pilihan pembatasan, yaitu pembatasan berdasarkan waktu atau pembatasan berdasarkan kuota. Pada pembahasan kali ini akan dibahas pembatasan berdasarkan waktu terlebih dahulu. Lakukan dengan cara klik menu Winbox IP > *Hotspot* > *tab User*. Pada tab *user* kita dapat menambahkan *user* baru seperti berikut untuk pembatasan waktu selama 30 menit..

Server : *all*
Name : client1
Password : client1
Profile : *default*
Limit Uptime : 00:30:00

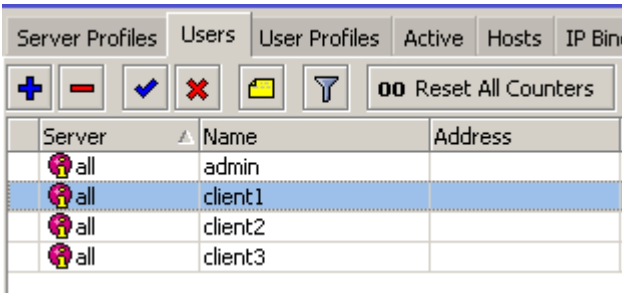


Gambar 6.13 *New Hotspot User (general)*



Gambar 6.14 *New Hotspot User (limits)*

Tambahkan untuk client2 dan client 3, sehingga pada tabel *user hotspot* akan seperti gambar 6.15 berikut.

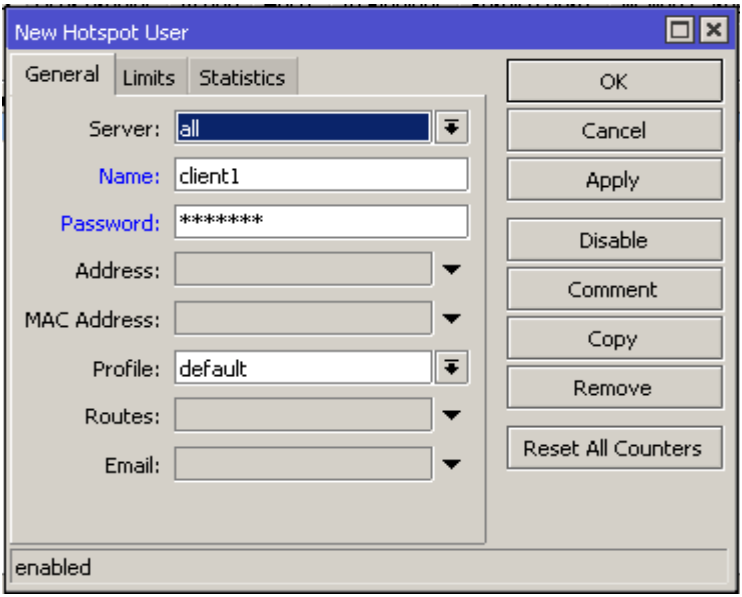


Gambar 6.15 Tabel *user hotspot*

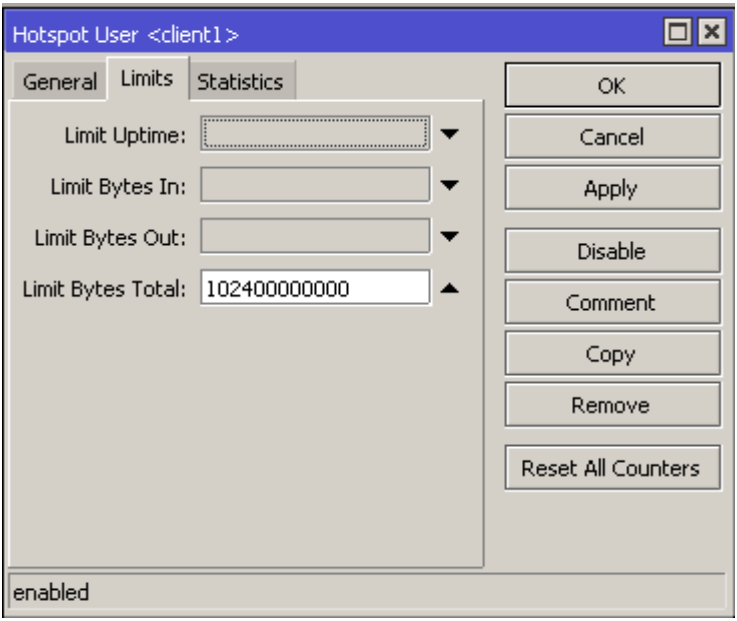
Volume Based Charging

Cara berikutnya mengenai pembatasan akses *hotspot* adalah melalui kuota (*volume*) apabila mencapai kuota tertentu maka secara otomatis koneksi tertutup. Hal itu dapat kita lakuka dengan cara seperti berikut dengan pembatasan total kuota sebesar 100 MB (102400000000 Bytes).

Server : *all*
Name : client1
Password : client1
Profile : *default*
Limit Bytes Total : 102400000000 (100 MB)



Gambar 6.16 Setting *volume based charging*

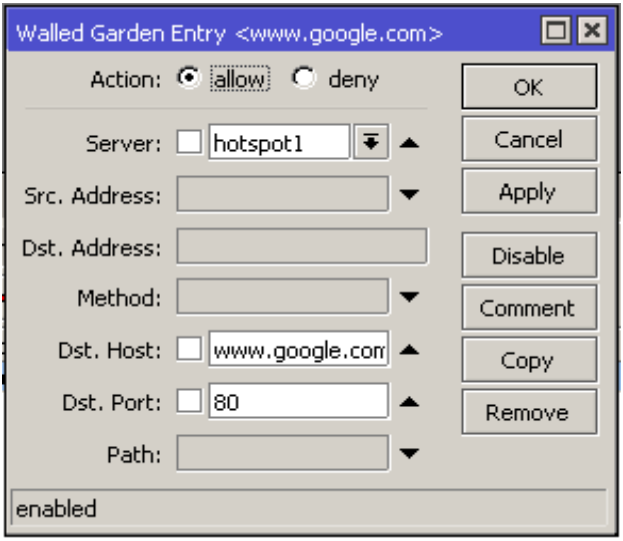


Gambar 6.17 *Limit Bytes Total* 100 MB

Walled Garden

Walled Garden adalah suatu istilah dimana kita dapat mengijinkan akses *hotspot* tanpa harus melewati proses otentikasi, dimana secara normal untuk mengakses *hotspot* kita harus *login* terlebih dahulu. Misalkan kita ingin membebaskan akses ke *http://www.google.com* dari jaringan *hotspot.* Maka bisa kita lakukan dengan klik menu di Winbox IP > *Hotspot* > *tab Walled Garden*. Kemudian tambahkan seperti berikut.

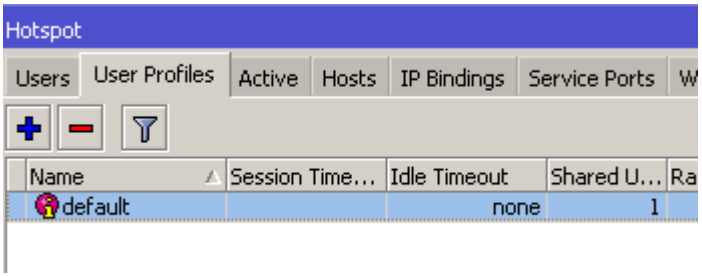
Action : *allow*
Server : *hotspot1*
Dst. Host : *www.google.com*
Dst. Port : *80*



Gambar 6.18 *Walled Garden*

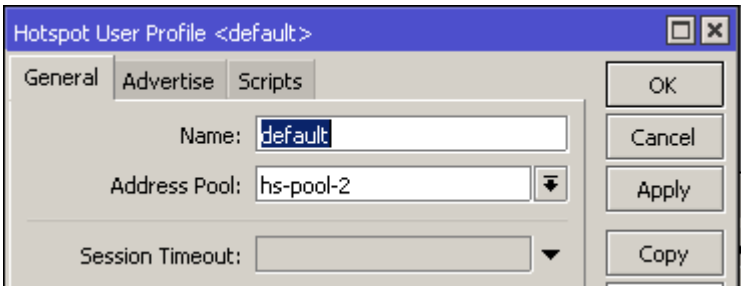
Hotspot Trial

Hotspot trial adalah salah satu *feature* pada *hotspot* Mikrotik yang berfungsi mem-*bypass* koneksi pada waktu tertentu tanpa harus melakukan otentikasi. Cara ini dapat dilakukan dengan cara klik menu Winbox IP > *Hotspot* > *tab User Profiles*. Klik pada *user profile* **default**.



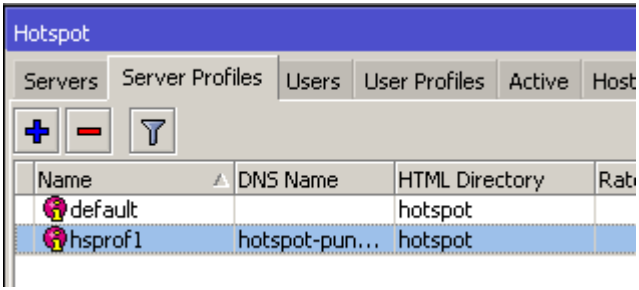
Gambar 6.19 *User Profiles*

Isikan pada kolom *Address Pool* sesuai dengan konfigurasi *Setup Hotspot* sebelumnya.



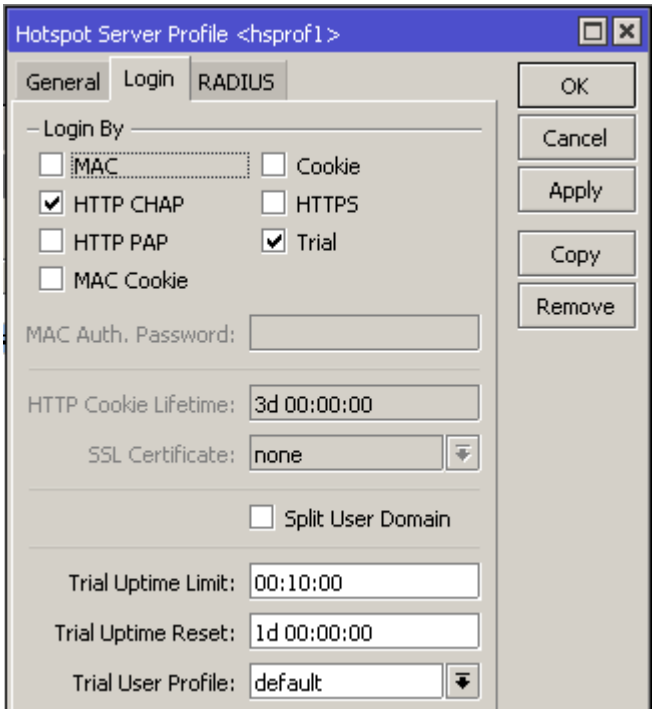
Gambar 6.20 *Address Pool*

Selanjutnya masuk ke tab *Server Profiles,* klik pada *server profile* yang dipakai pada *hotspot server*. Dalam contoh kali ini adalah **hsprof1.**



Gambar 6.21 *Server Profiles*

Centang pada *Trial* kemudian isikan berapa lama waktu *trial* dengan mengisikikn pada kolom Trial *Uptime Limit*. Contoh kali ini adalah selama 10 menit.



Gambar 6.22 *Trial Hotspot*

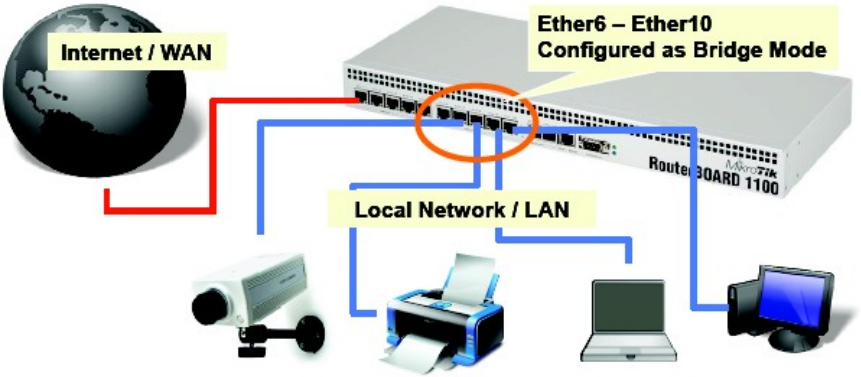
Cobalah buka browser pada klien kemudian akan muncul *link* yang mengijinkan kita menggunakan koneksi *trial*.

Gambar 6.23 *Free trial login*

BAB 7 Bridge

Bridge

Bridge merupakan teknik yang digunakan untuk menggabungkan 2 atau lebih *interface* yang seolah-olah berada dalam 1 segmen *network* yang sama. Bila dilihat dari layer OSI proses penggabungan ini terjadi pada layer *data link*. Mengaktifkan bridge pada 2 buah atau lebih *interface* akan menonaktifkan fungsi *routing* diantara *interface* tersebut. *Bridge* sama saja seperti mengemulasi mode *switch* secara *logic* pada dua atau lebih *interface*.



Gambar 7.1 *System bridge*

Meskipun demikian menggunakan *system bridge* memiliki beberapa konsukensi atau kekurangan, antara lain

Sulit untuk mengatur trafik *broadcast* (misalnya akibat virus).

Permasalahan pada satu *segment* akan membuat masalah pada semua *segment* pada *bridge* yang sama.

Sulit untuk melihat kualitas link pada tiap *segment.*

Beban trafik pada setiap perangkat yang dilalui akan berat, karena terjadi akumulasi traffic.

Berikut adalah jenis-jenis *interface* yang dapat kita jadikan *Bridge Port*.

Ethernet

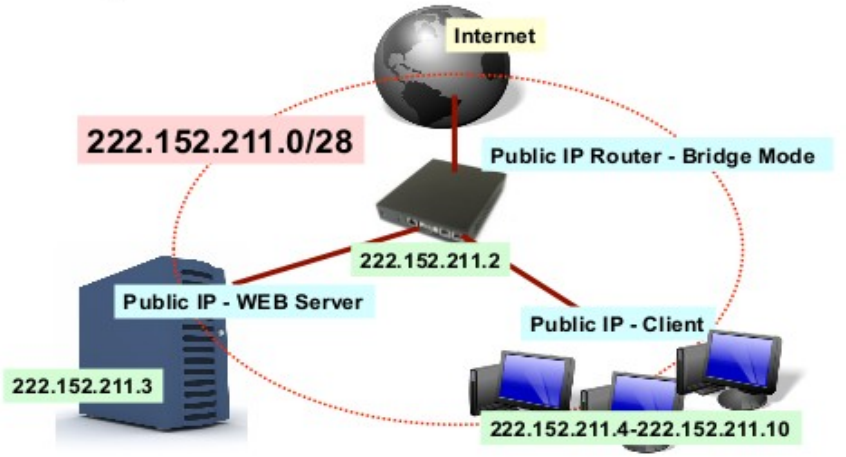
VLAN, merupakan bagian dari *ethernet* atau *wireless interface* dan perlu dicatat jangan melakukan *bridge* ke sebuah VLAN dengan *interface* induknya.

***Wireless* AP, WDS**

EoIP

PPTP

Gambar 7.2 dibawah ini adalah salah satu contoh bentuk implementasi dari jaringan *bridge*.

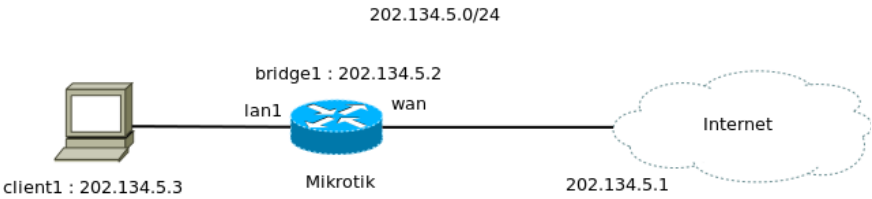


Gambar 7.2 Implementasi *bridge*

Dari gambar diatas bisa kita lihat bahwa antara jaringan internet dan jaringan di belakang router memiliki segment (*nework*) yang sama, jadi seperti itulah konsep dasasr dari *bridge*.

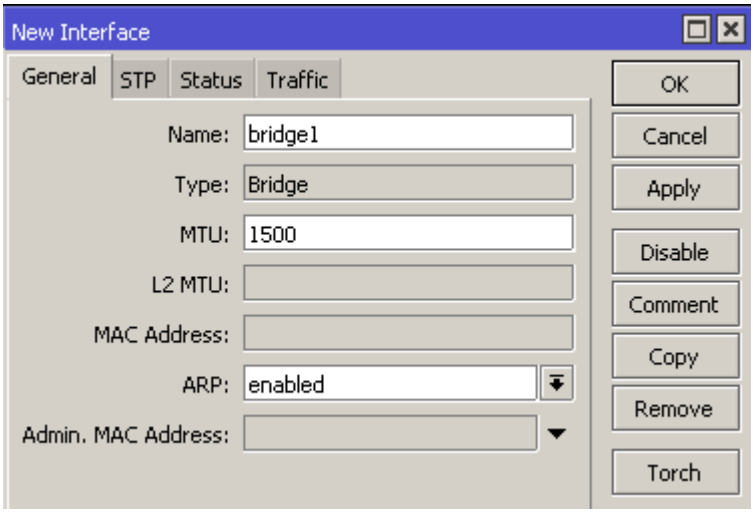
Konfigurasi Bridge

Berikut akan kita konfigurasi sebagai contoh penggunaan sederhana untuk *bridge* dengan melihat topologi jaringan seperti berikut :



Gambar 7.3 Lab *bridge*

Klik menu pada Winbox *Bidge > tab Bridge.* Tambahkan *interface bridge* dengan nama **bridge1.**

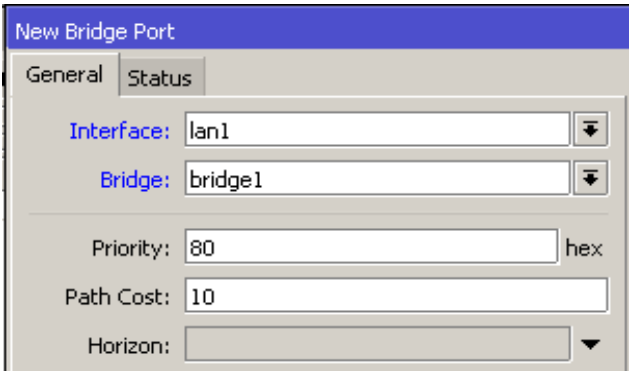


Gambar 7.4 *Add interface bridge*

Langkah berikutnya adalah menambahkan *port* yang akan kita masukan dalam member *interface* bridge1. Melalui menu Winbox *Bridge* > *tab Port*. Dalam kasus kali ini kita akan menambahkan *port* **wan** dan **lan1**.

Interface : lan1

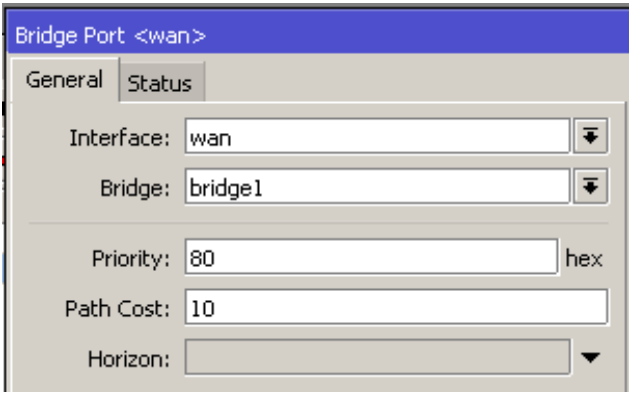
Bridge : bridge1



Gambar 7.5 *Bridge* lan1

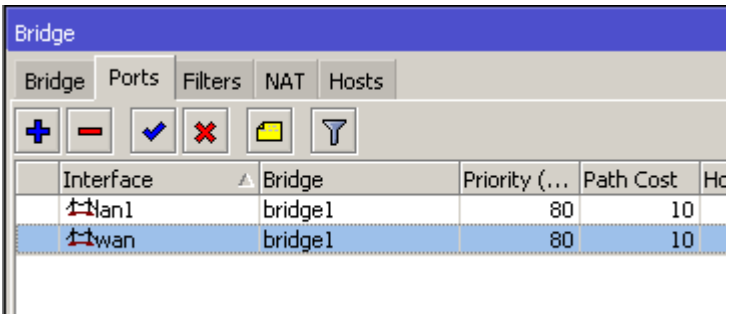
Interface : lan1

Bridge : bridge1



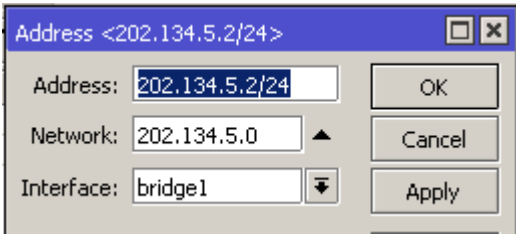
Gambar 7.6 *Bridge* wan

Bila kita lihat list *port*nya adalah seperi gambar dibawah ini.



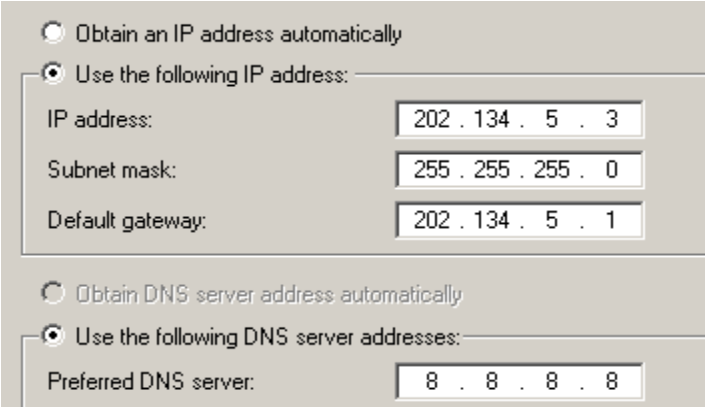
Gambar 7.7 List *port bridge*

Kemudian tambahkan IP untuk *interface* bridge1 klik menu IP > *addresses.*



Gambar 7.8 IP *address* bridge1

Untuk langkah pengujian dari klien atur IP klien sesuai dengan gambar topologi diatas, yaitu 202.134.5.3 masukkan IP *geteway* dengan IP ISP (202.134.5.1)



Gambar 7.9 IP klien

Lakukan uji ping dari klien ke IP *interface* bridge1, IP gateway dan internet (www.google.com)

```
C:\Documents and Settings\Akrom Musajid>ping www.google.com

Pinging www.google.com [118.98.30.39] with 32 bytes of data:

Reply from 118.98.30.39: bytes=32 time=568ms TTL=59
Reply from 118.98.30.39: bytes=32 time=691ms TTL=59
Reply from 118.98.30.39: bytes=32 time=755ms TTL=59
Reply from 118.98.30.39: bytes=32 time=1329ms TTL=59

Ping statistics for 118.98.30.39:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 568ms, Maximum = 1329ms, Average = 835ms
```

Gambar 7.10 Ping klien

BAB 8 Tunneling

Tunnel

Tunnel merupakan suatu cara untuk meng-enkapsulasi atau membungkus paket data yang biasa digunakan untuk membuat jalur *private*/pribadi pada jaringn *public* (internet).

Tunnel memiliki beberapa macam jenis antara lain yang terdapat pada Mikrotik adalah EoIP, Ipsec, IPIP, L2TP, PPPoE, PPTP, VLAN, MPLS dan OpenVPN.

EoIP Tunnel

Ethernet over IP (EoIP) *Tunnel* adalah salah satu teknik *tunnel* yang hanya dimiliki oleh Mikrotik, oleh karena itu EoIP hanya dapat dilakukan antara sesama Mikrotik saja. Namun EoIP tidak menggunakan enkripsi untuk melindungi jalannya data, jadi tidak disarankan bila digunakan untuk transmisi data yang membutuhkan tingkat keamanan tinggi. Mikrotik mampu membuat *tunnel* menggunakan EoIP maksimum sebanyak 65535.

Kali ini kita akan membuat contoh konfigurasi untuk penggunaan EoIP tunnel dengan gambaran topologi jaringan seperti gambar 8.1 di bawah ini.

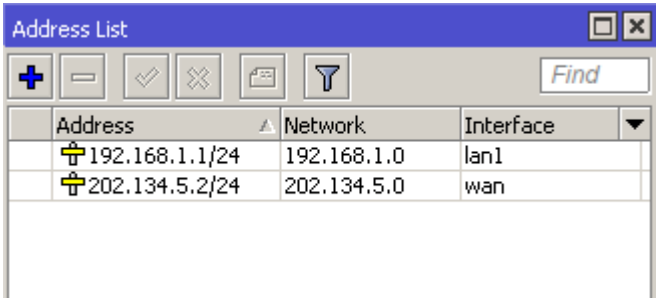
Gambar 8.1 EoIP *tunnel*

Untuk membangun *tunnel* EoIP lakukan konfigurasi seperti berikut :

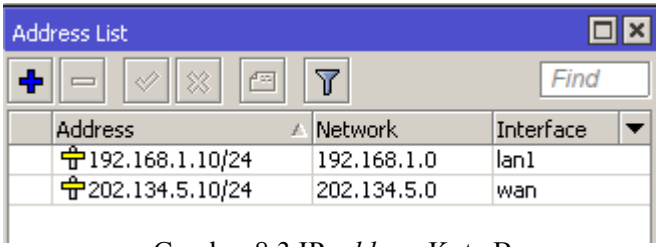
Tambahkan IP address pada masing-masing *router,* sesuai dengan gambar topologi diatas *router* pada kota A dan kota B masing-masing memiliki 2 interface dan 2 IP.

Router kota A : wan, 202.134.5.2/24
lan1, 192.168.1.1/24

Router kota B : wan, 202.134.5.10/24
lan1, 192.168.1.10/24



Gambar 8.2 IP *address* Kota A



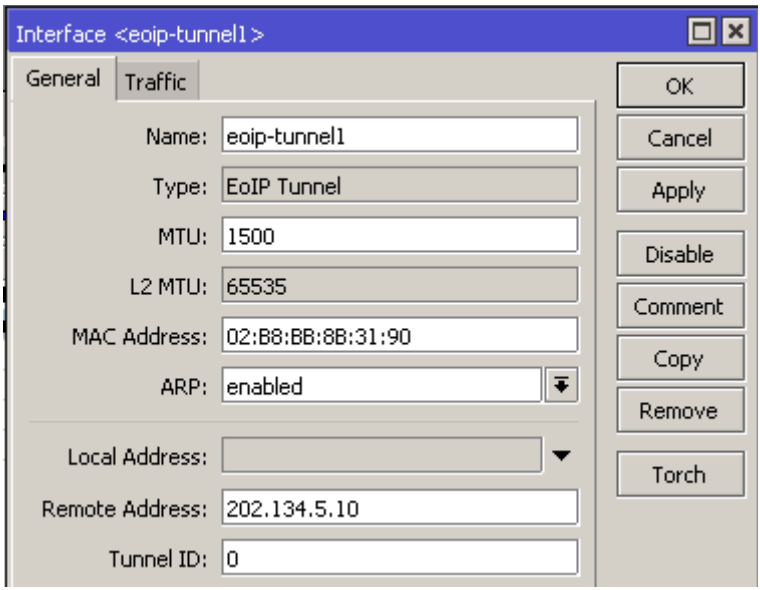
Gambar 8.3 IP *address* Kota B

Tambahkan sebuah *interface* baru dengan tipe EoIP *Tunnel* dengan cara klik menu *Interface* > *tab* EoIP *Tunnel* menggunakan Winbox. Kemudian klik '+'. Perlu diingat bawha **Tunnel ID** pada sebuah EoIP *tunnel* harus sama antar kedua EoIP *tunnel* dan **MAC *address*** antar EoIP harus saling berbeda.

Name : eoip-tunnel1

Remote Address : 202.134.5.10

Tunnel ID : 0

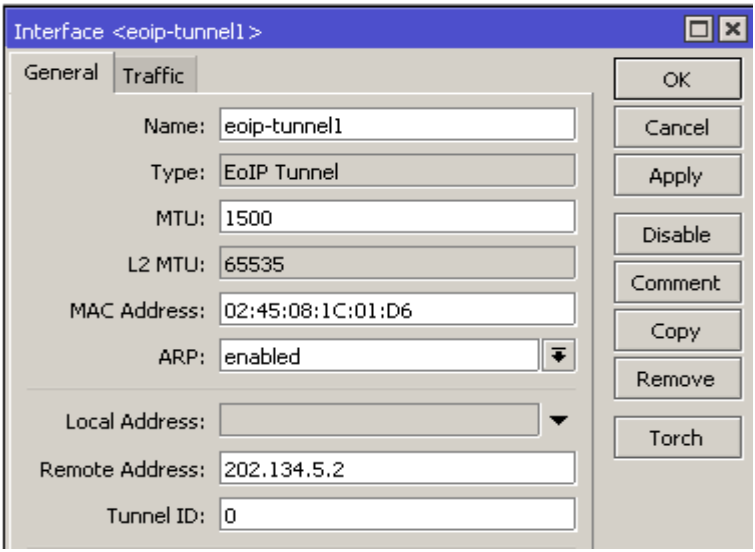


Gambar 8.4 EoIP *Interface* kota A

Name : eoip-tunnel1

Remote Address : 202.134.5.2

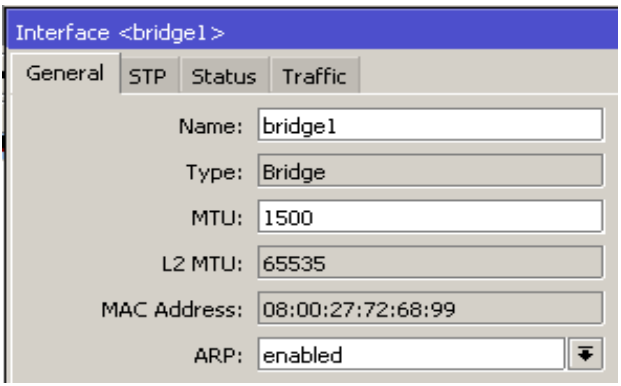
Tunnel ID : 0



Gambar 8.5 EoIP *Interface* kota B

Langkah berikutnya adalah masukkan *interface* lokal dan *interface* EoIP ke dalam *interface bridge* pada masing-masing *router* dengan menambahkan satu *interface bridge* terlebih dahulu dengan cara klik menu *Bridge* > *tab Bridge,* klik tanda '+'.

Name : bridge1

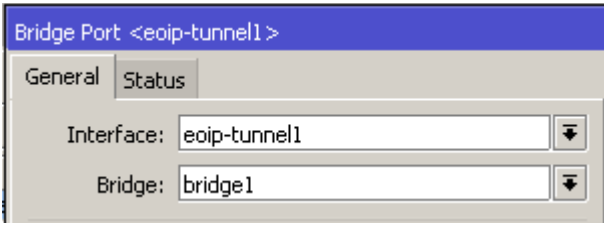


Gambar 8.6 *Interface brige*\

Klik pada menu *Bridge* > *tab Ports* dan masukkan *Interface* lan1 dan *interface* eoip-tunnel1 ke dalam *interface bridge1* pada masing-masing *router*.

Interface : eoip-tunnel1

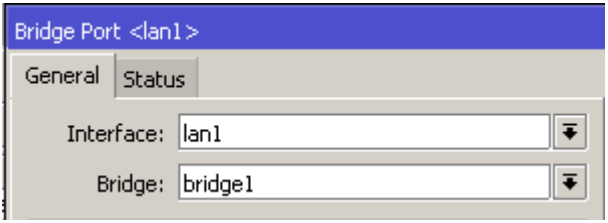
Bridge : bridge1



Gambar 8.7 eoip-tunnel1 *bridge*

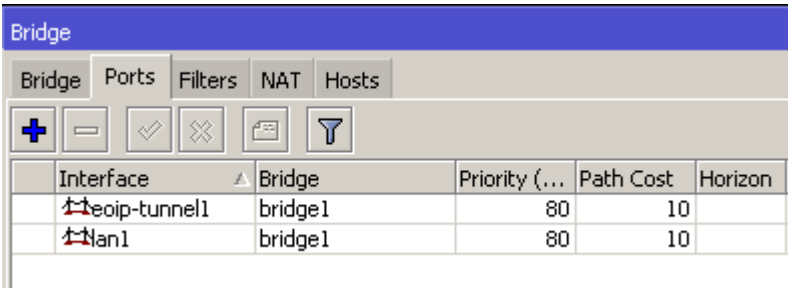
Interface : eoip-tunnel1

Bridge : bridge1



Gambar 8.8 lan1 *bridge*

Bila dilihat tabel *port bridge* akan terlihat seperti gambar di bawah ini.



Gambar 8.9 Tabel *port bridge*

Set IP *address* komputer lokal pada kota A dan kota B dalam satu *network* yang sama (komputer A 192.168.1.2 dan komputer B 192.168.1.10). Lakukan komunikasi antar kedua komputer tersebut seperti test ping untuk komunikasi sederhananya.

PPTP Tunnel

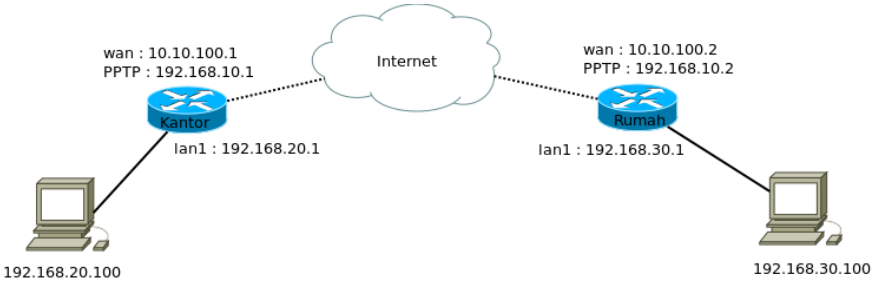
Point to Point Tunneling Protocol (PPTP) merupakan salah satu jenis *protocol tunnel* yang digunakan pada Mikrotik yang berjalan pada layer 3 pada layer OSI yaitu layer *network* dan menggunakan *port* TCP 1723. Beberapa alasan kenapa menggunakan PPTP antara lain :

Koneksi antar *router over* internet yang bersifat *secure* (aman).

Untuk menghubungkan jaringan lokal via WAN.

Untuk digunakan sebagai *mobile client* atau *remote client* yang ingin melakukan akses ke *network local.*

Kita akan membuat koneksi PPTP *tunnel* dari kantor ke rumah untuk memungkinkan koneksi remote yang aman, dimana jaringan lokal antara kantor dan rumah memiliki *network* yang berbeda dan akan di*route* ke PPTP *tunnel*.



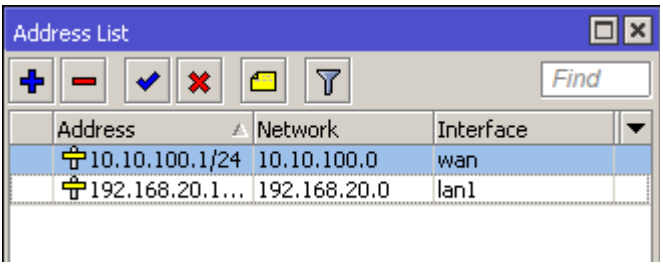
Gambar 8.10 PPTP *Tunnel*

Sesuai pada gambar topologi jaringan PPTP di atas konfigurasi yang harus pertama dilakukan adalah atur IP pada masing-masing *router* untuk *interface* lan1 dan wan.

Kantor

lan1 : 192.168.20.1/24

wan : 10.10.100.1/24

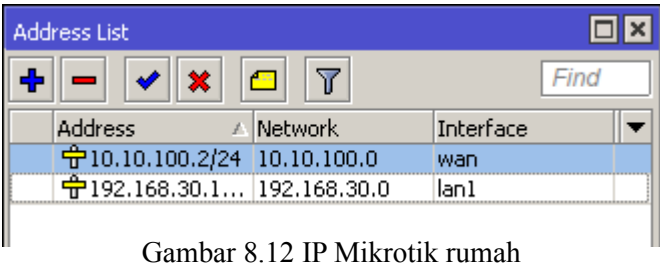


Gambar 8.11 IP Mikrotik kantor

Rumah

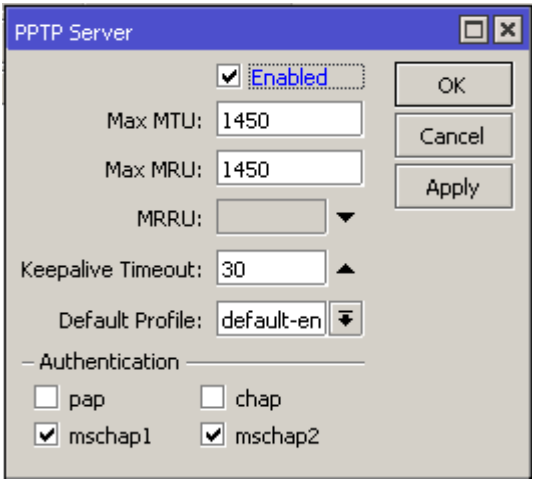
lan1 : 192.168.30.1

wan : 10.10.100.2



Gambar 8.12 IP Mikrotik rumah

Langkah berikutnya adalah mengaktifkan PPTP *tunnel* dengan cara klik menu Winbox PPP > tab *interface* > PPTP server, centang pada *enable*.



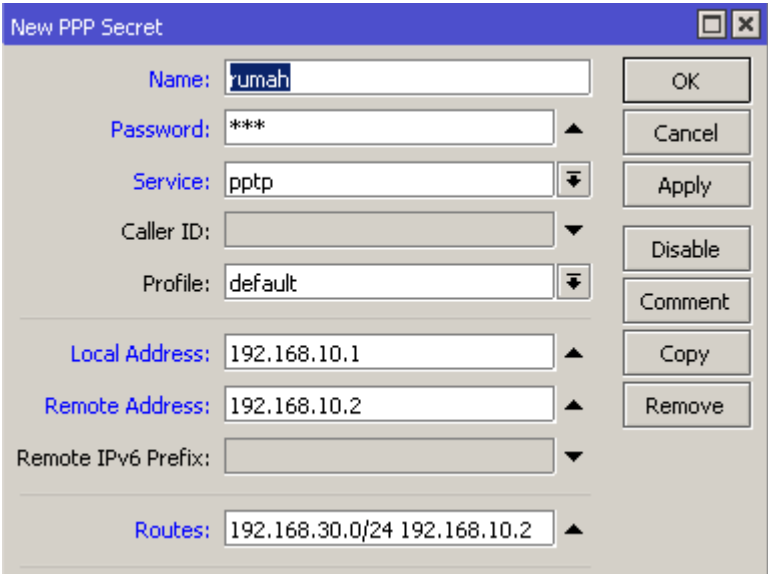
Gambar 8.13 *Enable* PPTP

Setelah mengaktifkan PPTP, kemudian buatlah PPP *secret* yang akan digunakan untuk otentikasi *remote* dari Mikrotik rumah ke Mikrotik kantor. Lakukan dengan cara klik menu Winbox PPP > *tab Secrets*. Tambahkan *secret* dengan klik tanda '+'. Sesuai dengan topologi jaringan diatas untuk konfigurasi *secret* adalah seperti berikut.

Name : Rumah

Password : 123

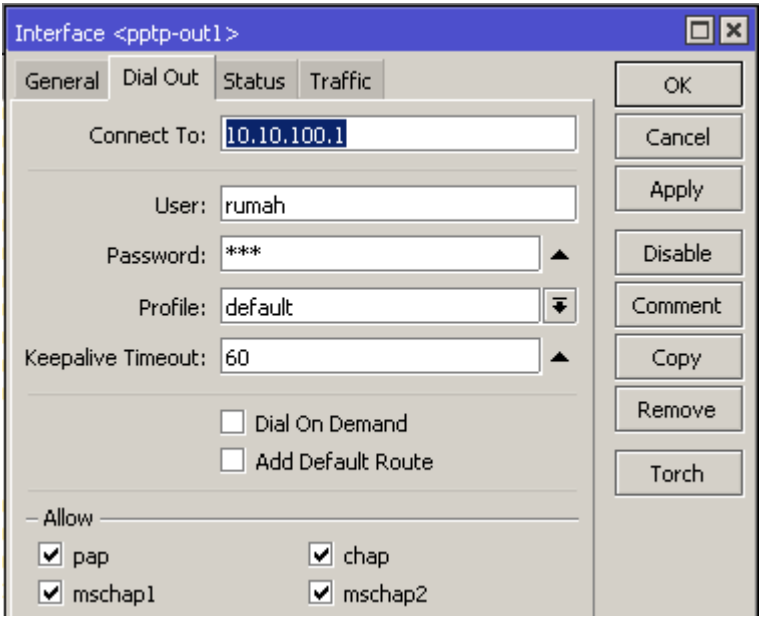
Service : pptp
Profile : default
Local Address : 192.168.10.1
Remote Address: 192.168.10.2
Routes : 192.168.30.0/24 192.168.10.2



Gambar 8.14 PPP *secret*

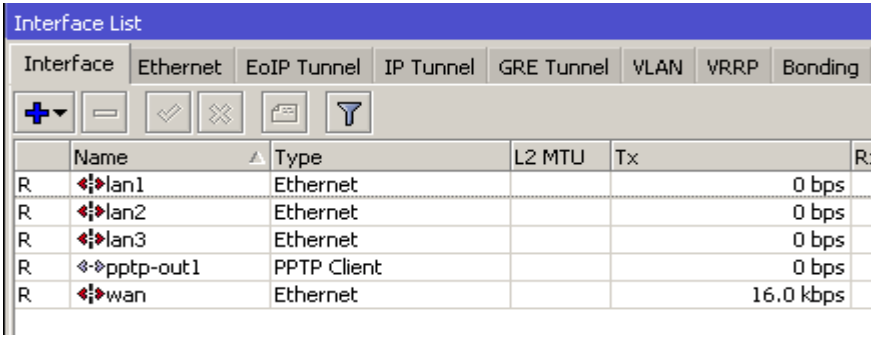
Setelah konfigurasi PPP *secret* dilakukan selanjutnya kita dapat lakukan koneksi *tunnel* dari Mikrotik rumah menuju Mikrotik kantor dengan cara klik menu Winbox PPP > *tab interface.* Tambahkan *interface* PPTP *client* dengan klik pada tanda '+'.

Connect To : 10.10.100.1
User : rumah
Password : 123
Profile : default

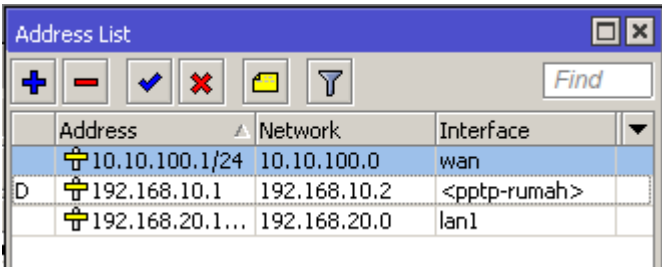


Gambar 8.15 PPTP *client*

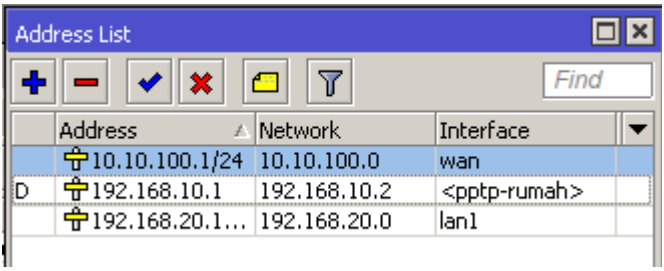
Apabila koneksi berhasil maka akan kita lihat muncul *interface* baru dan IP *address* baru pada kedua router Mikrotik yang dapat kita lihat di menu *interface* Mikrotik.



Gambar 8.16 *Interface* PPTP



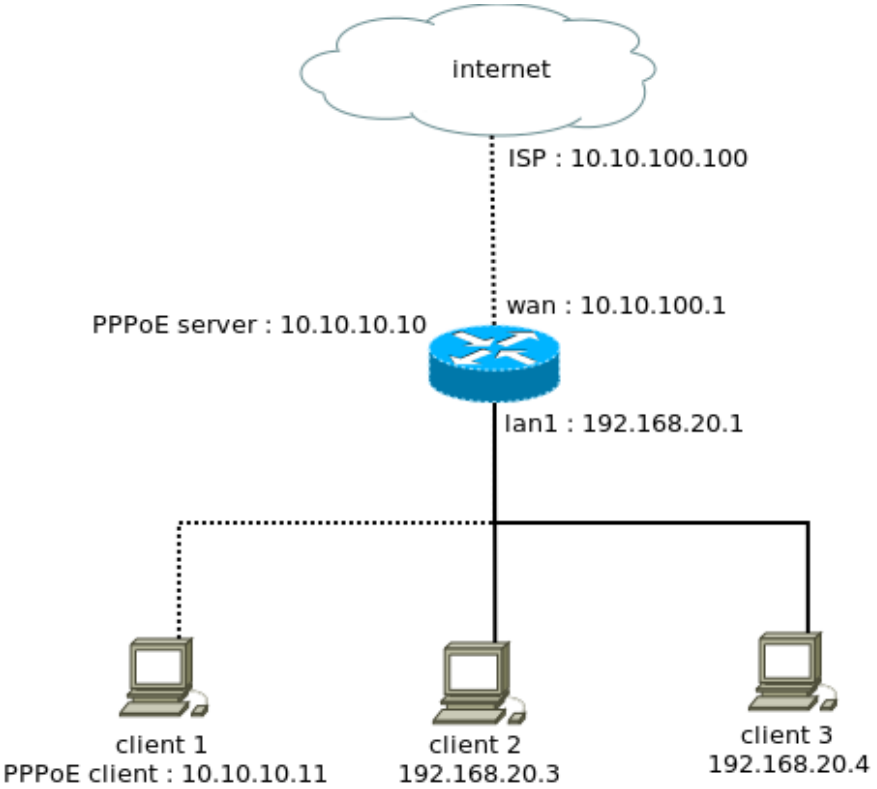
Gambar 8.17 IP *address* PPTP kantor



Gambar 8.17 IP *address* PPTP kantor

PPPoE Tunnel

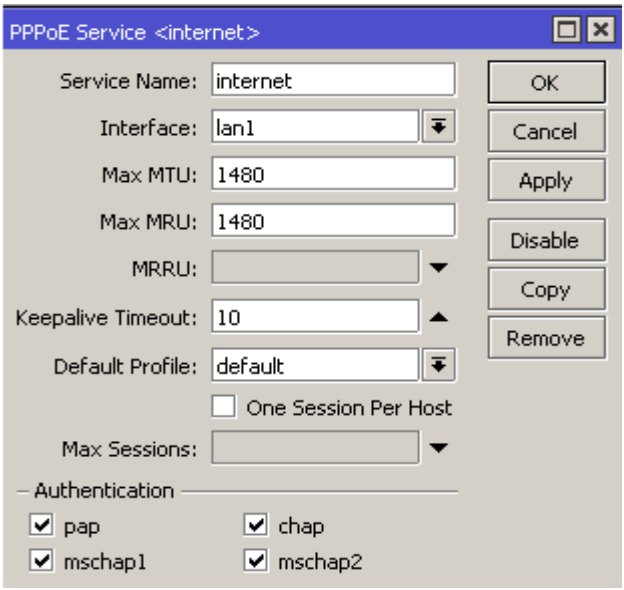
PPPoE *tunnel* adalah koneksi antar *client* dan *router* yang bersifat *secure*. Untuk digunakan sebagai koneksi internet bersifat *secure* di jaringan local (LAN). Sebuah koneksi PPPoE terdiri dari sever dan *client*. Mikrotik bisa difungsikan sebagai PPPoE server maupun PPPoE *client* bahkan gabungan dari keduanya. Koneksi PPPoE menggunakan *ethernet frame* sebagai *protocol* transportnya. Sebagian besar sistem operasi sudah memiliki fungsi untuk koneksi PPPoE *client.*



Gambar 8.18 Jaringan PPPoE

Untuk konfigurasinya langkah yang harus dilakukan pertama kali adalah menambahkan PPPoE server dari Mikrotik dengan cara klik Menu PPP > *tab* PPPoE Servers, tambahkan dengan klik tanda '+'.

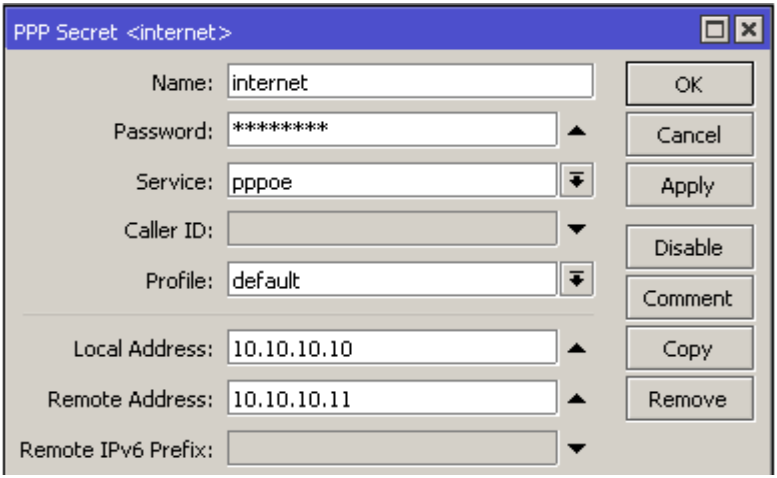
Service Nama : internet
Interface : lan1
Default Profile : default



Gambar 8.19 *Add* PPPoE

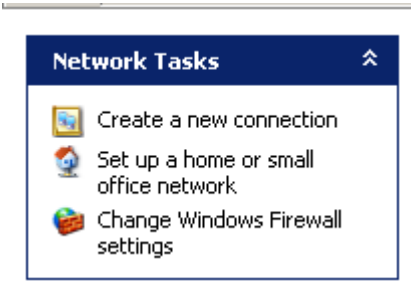
Kemudian buatlah sebuah *secret* yang digunakan untuk keperluan otentikasi pada *client* yang akan terhubung. Klik Menu PPP > *tab Secrets.* Tambahkan dengan klik tanda '+'.

Name : internet
Password : internet
Service : pppoe
Profile : default
Local Address : 10.10.10.10
Remote Address : 10.10.10.11



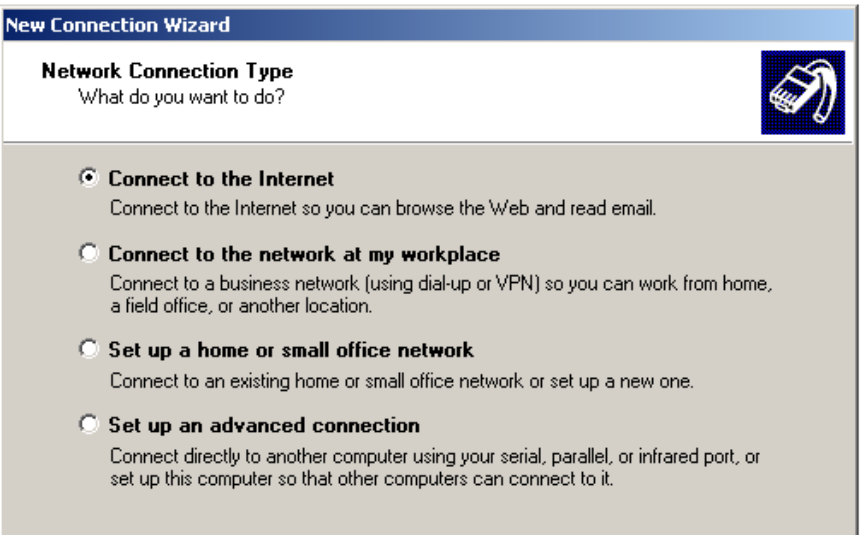
Gambar 8.20 *Secret* PPPoE

Langkah terakhir adalah melakukan koneksi dari client, koneksi ini dapat dilakukan di berbagai sistem operasi, Windows XP, Windows 7, Windows 8, maupun Linux. Contoh kali ini akan dilakukan koneksi PPPoE client dari Windows XP. Pertama yang harus dilakukan adalah buat koneksi baru dengan membuka *control panel* > *Network and Internet Connection* > *Network Connection* > *Create a new connection*



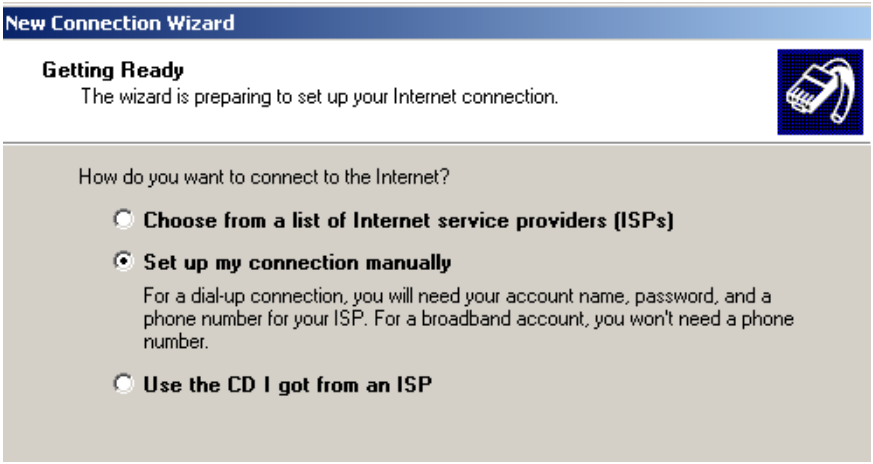
Gambar 8.21 *Create a new connection*

Kemdian akan muncul halaman *Network Connection Type,* Pilih pada



pilihan *Connect to the Internet.*Gambar 8.22 *Network Connection Type*

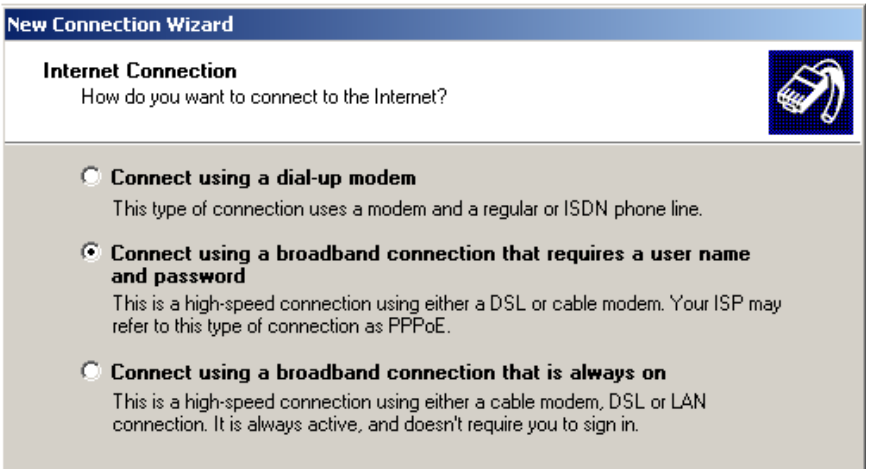
Berikutnya adalah halaman untuk mempersiapkan koneksi internet. Pilih pada pilihan *Set up my connection manually.*



Gambar 8.23 *Getting Ready*

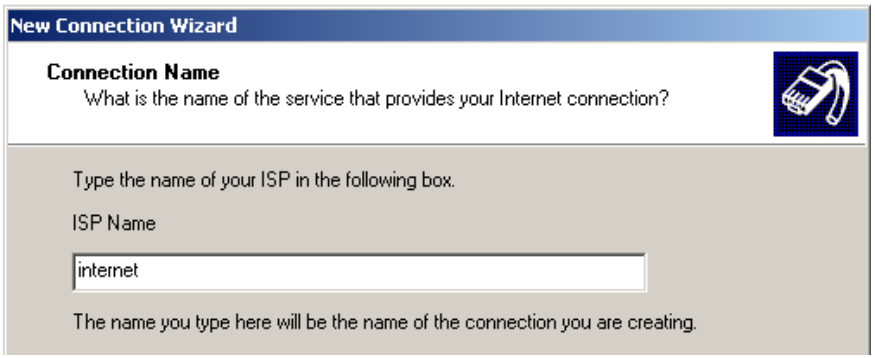
Kemudian pada halaman *Internet Connection* Pilih *Connect using a*

broadband connection that requires a user name and passsword.



Gambar 8.24 *Internet Connection*

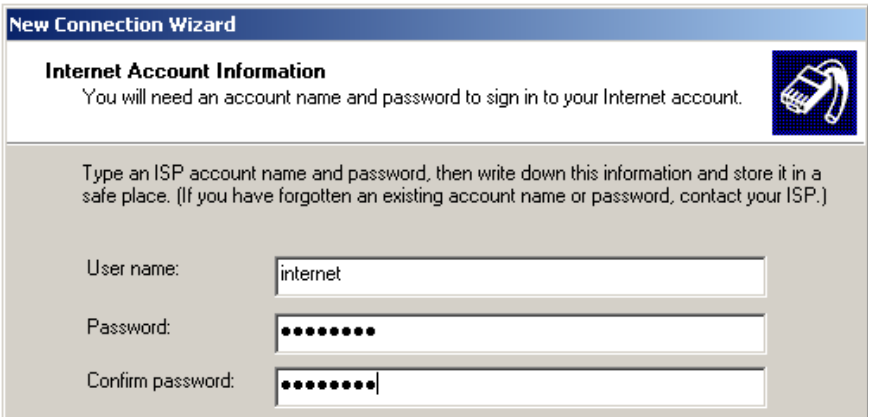
Pada *Connection Name* isikan nama pada koneksi ini. Contoh dibawah ini berisi nama koneksi **internet**.



Gambar 8.25 *Connection Name*

Di halaman *Internet Account Information* isikan *user name* dan *password* seperti *secret* yang dibuat pada Mikrotik

User name : internet
Password : internet
Confirm Password : internet



Gambar 8.26 *Internet Account Information*

Setelah konfigurasi diatas lakukan *connect internet* seperti pada gambar dibawah ini.

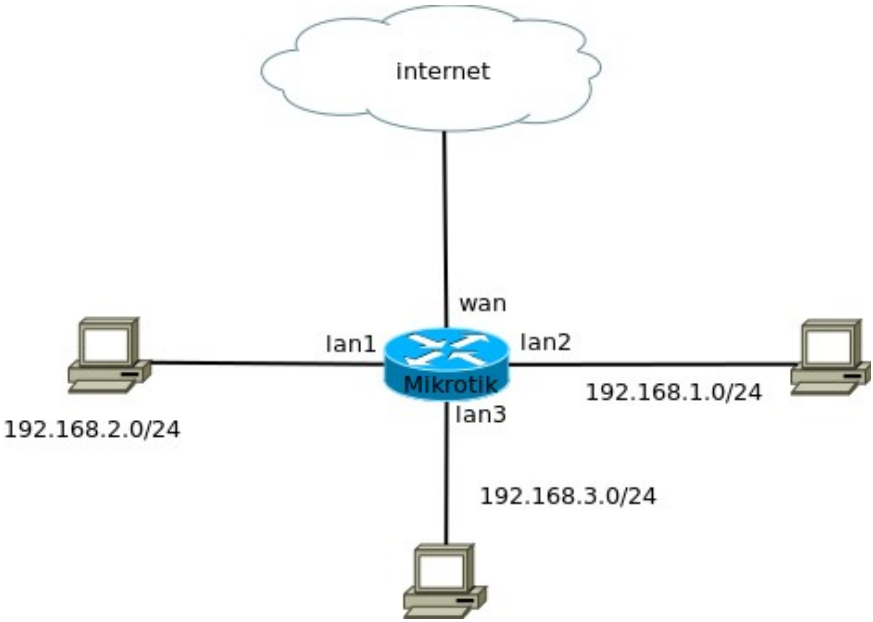
Gambar 8.27 *Connect internet*

BAB 9 Routing

Routing

Routing merupakan pengaturan jalur antar *segment network* yang berbeda berdasarkan IP *address* tujuan maupun asal. *Routing* bekerja pada layer 3 (*network*) bila berdasarkan OSI layer. Untuk menghubungkan *network* yang berbeda *segment* memerlukan sebuah perangkat yang mampu melakukan proses *routing* yang disebut *router*.

Mikrotik yang berfungsi sebagai *router* akan menjembatani komunikasi antar *network* yang berbeda.



Gambar 9.1 *Routing*

Beberapa keuntungan yang didapat dari proses *routing* ini antara lain :

Memungkinkan kita melakukan pemantauan dan pengelolaan jaringan yang lebih baik.

Lebih aman (*firewall filtering* lebih mudah).

Trafik *broadcast* (virus) hanya terkonsentrasi di *local network* dengan segmen yang sama.

Untuk *network* skala besar, *routing* bisa diimplementasikan menggunakan *Dynamic Routing protocol* (RIP/OSPF/BGP).

Tipe Routing

Secara umum *routing* dibedakan dengan 2 jenis, antara lain :

Static Routing

Static routing adalah informasi *routing* yang dapat dibuat secara manual oleh seorang *administrator* jaringan untuk mengatur ke arah mana saja trafik tertentu akan disalurkan. *Default route* adalah salah satu contoh *static routing*.

Dynamic Routing

Berbeda dengan *static routing, dynamic routing* adalah infomasi *routing* yang dibuat secara otomatis oleh *router* sendiri. Informasi *routing* yang didapat dari *protocol dynamic routing* seperti RIP, OSPF, dan BGP.