BAB III

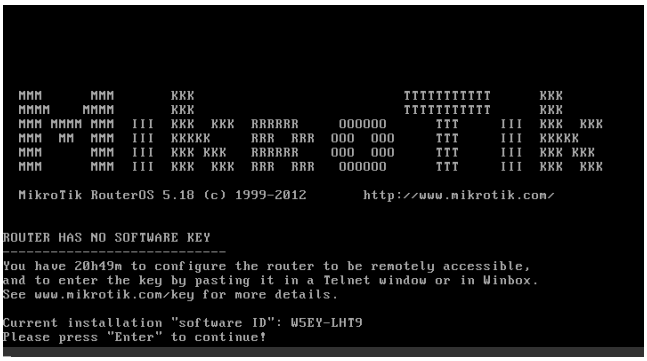
LANDASAN TEORI

3.1 Pengertian Mikrotik

MikroTikls atau yang lebih di kenal dengan Mikrotik didirikan tahun 1995 bertujuan mengembangkan sistem ISP dengan *wireless*. Mikrotik saat ini telah mendukung sistem ISP dengan *wireless* untuk jalur data internet di banyak negara, antara lain Iraq, Kosovo, Sri Lanka, Ghana dan banyak negara lainnya.. Berbagai pengembangan telah dilakukan hingga saat ini dengan tersedianya perangkat lunak sistem operasi *router* versi 2 yang menjamin kestabilan, kontrol, dan fleksibilitas pada berbagai media antar muka dan sistem *routing* dengan menggunakan komputer standart sebagai *hardware*. Perangkat lunak ini mendukung berbagai aplikasi ISP, mulai dari *RADIUS* modem pool, hingga sirkuit *backbone* dengan DS3. Mikrotik berlokasi di Riga, ibukota Latvia, dengan 50 orang karyawan. Mikrotik juga menjalankan sebuah *ISP* kecil, sebagai media percobaan untuk pengembangan *RouterOS software* (Moch. Linto Herlambang, 2008)

3.1.1 Jenis - Jenis Mikrotik

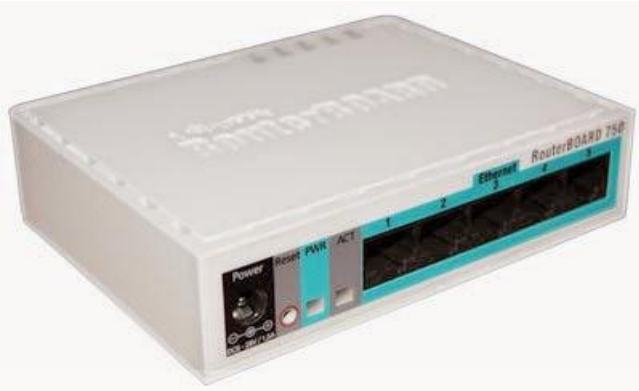
- **Mikrotik RouterOS**



Gambar 3.1 *RouterOS*

Gambar 3.1 adalah tampilan dari *RouterOS*. Sistem operasi yang dapat digunakan untuk menjadikan komputer menjadi *router network* yang handal, mencakup bebagai fitur yang dibuat untuk *IP network* dan jaringan *wireless*, cocok digunakan oleh *ISP* dan *provider hotspot*. Untuk instalasi Mikrotiik tidak dibutuhkan piranti lunak tambahan atau komponen tambahan lain. Mikrotik didesain untuk mudah digunakan dan sangat baik digunakan untuk keperluan administrasi jaringan komputer seperti merancang dan membangun sebuah sistem jaringan komputer.

- **Mikrotik RouterBoard**



Gambar 3.2 *RouterBoard*

Gambar 3.2 adalah bentuk dari *RouterBoard*. *RouterBoard* seperti sebuah *pesonal komputer* mini yang terintegrasi karena dalam satu board tertanam prosesor, *ram, rom,* dan memori *flash*. *RouterBoard* menggunakan *OS RouterOS* yang berguna sebagai *router* jaringan, *bandwidth management, proxy server, dhcp, dns server dan hotspot server.*

Fitur yang bisa didapatkan di mikrotik adalah :

- *Firewall dan NAT*
- *Routing – Static routing*
- *Data Rate Management*
- *Hotspot*
- *Point-to-Point tunneling protocols*
- *Simple tunnels*

- *IPsec*
- *Web proxy*
- *Caching DNS client*
- *DHCP*
- *Universal Client*
- *VRRP*
- *UPnP*
- *NTP*
- *Monitoring/Accounting*
- *SNMP*
- *M3P*
- *MNDP*
- *Tools*

3.2 Pengertian *Proxy*

Proxy adalah suatu *server* yang menyediakan layanan untuk meneruskan setiap permintaan kita kepada *server* lain di internet. Dengan *proxy*, maka identitas komputer anda berupa *IP* menjadi tersembunyi dikarenakan yang dikenali server yang direquest adalah *IP* dari *server proxy* anda. *Proxy* ini pada umumnya digunakan untuk kegiatan menyembunyikan identitas atau untuk menghindari pemblokiran akses ke suatu *server*.

P*roxy Server* adalah *server* yang diletakkan antara suatu aplikasi *client* dan aplikasi *server* yang dihubungi. Aplikasi *client* dapat berupa *browser web, client FTP*, dan sebagainya. Sedangkan aplikasi *server* dapat berupa *server web*, *server FTP* dan sebagainya. *Proxy Server* yang diletakkan di antara aplikasi *client* dan aplikasi *server* tersebut, dapat digunakan untuk mengendalikan maupun memonitor lalu-lintas paket data yang melewatinya (Wagito, 2007).

3.2.1 Manfaat Proxy Server

Secara umum manfaat *proxy server* ada dua macam, yaitu sebagai berikut:

- Meningkatkan kinerja jaringan

Dengan kemampuan *server proxy* untuk menyimpan data permintaan dari aplikasi client, permintaan yang sama dengan permintaan sebelumnya hanya akan diambilkan dari simpanan *server proxy*. Jika seorang pengguna internet sudah pernah membuka situs yang sama, tidak perlu dihubungkan langsung pada situs sumbernya, tetapi cukup diambilkan dari simpanan *server proxy*. Dengan cara demikian, koneksi langsung pada *server* sumbernya dapat dikurangi. Dengan demikian, penggunaan *bandwidth* internet untuk koneksi langsung menjadi berkurang.

- Filter permintaan

Server proxy juga dapat digunakan sebagai *filter* terhadap permintaan data dari suatu situs. Dalam hal ini, *server proxy* menjadi *filter* terhadap situs yang boleh atau

tidak boleh dikunjungi. Selain itu, *server proxy* juga dapat sebagai *filter* terhadap aplikasi client yang dapat menggunakan akses terhadap internet. Dalam hal ini *server proxy* berlaku sebagai *filter* terhadap gangguan internet.

3.2.2 Fungsi *Proxy Server*

Proxy Server merupakan pihak ketiga yang menjadi perantara antara kedua pihak yang saling berhubungan, dalam hal ini adalah jaringan lokal dan jaringan internet. Secara prinsip pihak pertama dan pihak kedua tidak langsung berhubungan, akan tetapi masing-masing berhubungan degan pihak ketiga yaitu *proxy*.

Tiga fungsi utama *proxy server* adalah:

- *Connection sharing*

Bertindak sebagai gateway yang menjadi batas antara jaringan lokal dan jaringan luar. *Gateway* juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya dan koneksi jaringan luar juga terhubung kepadanya. Dengan demikian koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh *gateway* secara bersama-sama *(connecion sharing).*

- *Filtering*

Layar aplikasi berfungsi sebagai *firewalll* paket *filtering* yang digunakan untuk melindungi jaringan lokal terhadap gangguan atau serangan dari jaringan luar dan untuk menolak situs web tertenu pada waktu-waktu tertentu.

- *Caching*

Proxy Server memiliki mekanisme penyimpanan obyek-obyek yang sudah diminta dari *server-server* di internet. Mekanisme *caching* akan menyimpan obyek-obyek yang merupakan permintaan dari para pengguna yang didapat dari internet.

3.2.3 Keuntungan *Proxy Server*

Keuntungan *proxy server* dalam suatu jaringan *TCP/IP* adalah:

1. Keamanan jaringan lebih terjaga, karena adanya *proxy* sebagai pembatas antara jaringan lokal dan jaringan luar (internet).
2. Pengaksesan kembali terhadap situs-situs yang telah diakses sebelumnya menjadi lebih cepat, karena pengaksesan tidak perlu ke jaringan luar (internet) melainkan ada pada direktori *cache proxy*.
3. Terdapat fasilitas *filtering*, baik *filtering* pengguna, *content* dan waktu akses.

3.2.4 Kekurangan *Proxy Server*

Kekurangan *proxy server* dalam suatu jaringan *TCP/IP* adalah:

1. Pengaksesan terhadap situs yang belum pernah dibuka sebelumnya akan menjadi lebih lambat, karena *client* harus meminta terlebih dahulu ke pada *proxy*, setelah itu baru *proxy* yang akan meminta *request* dari *client* tersebut ke pada penyedia layanan internet.
2. Bila *proxy server* terlambat melakukan *update cache*, maka *client* akan mendapatkan *content* yang belum *update* ketika melakukan *request content* tersebut.

3.3 Pengertian *ACL (ACCESS CONTROL LIST)*

Access Control List (ACL) adalah pengelompokan paket berdasarkan kategori. *ACL* bisa sangat membantu ketika membutuhkan pengontrolan dalam lalu lintas *network. ACL* menjadi *tool* pilihan untuk pengambilan keputusan pada situasi ini. Penggunaan *ACL* yang paling umum dan paling mudah untuk dimengerti adalah penyaringan paket yang tidak diinginkan ketika mengimplementasikan kebijakan keamanan. Sebagai contoh kita dapat mengatur ACL untuk membuat keputusan yang sangat spesifik tentang peraturan pola lalu lintas sehingga *ACL* hanya memperbolehkan *host* tertentu mengakses sumber daya *WWW* sementara yang lainnya ditolak. Dengan kombinasi *ACL* yang benar, *network* manajer mempunyai kekuasaan untuk memaksa hampir semua kebijakan keamanan yang bisa mereka ciptakan. *ACL* juga bisa digunakan pada situasi lain yang tidak harus meliputi

penolakan paket. Sebagai contoh ACL digunakan untuk mengontrol *network* mana yang akan atau tidak dinyatakan oleh protokol *dynamic* routing. Perbedaannya adalah bagaimana menerapkan ke protokol *routing* dan bukan ke *interface*. Kita juga bisa menggunakan *ACL* untuk mengkategorikan paket atau antrian / layanan *QOS*, dan mengontrol tipe lalu lintas data nama yang akan mengaktifkan *link* ISDN.

Membuat *ACL* sangat mirip dengan *statement* pada *programming* if – then jika sebuah kondisi terpenuhi maka aksi yang diberikan akan dijalankan tidak terpenuhi, tidak ada yang terjadi dan *statement* berikutnya akan dievaluasi. *Statement ACL* pada dasarnya adalah paket *filter* dimana paket dibandingkan, dimana paket dikategorikan dan dimana suatu tindakan terhadap paket dilakukan. Daftar yang telah dibuat bisa diterapakan baik kepada lalu lintas *inbound* maupun *outbound* pada *interface* mana saja. Menerapkan *ACL* menyebabkan *router* menganalisa setiap paket arah spesifik yang melalui *interface* tersebut dan mengambil tindakan. Ketika paket dibandingkan dengan ACL, terdapat beberapa peraturan penting yang diikuti:

- Paket selalu dibandingkan dengan setiap baris dari *ACL* secara berurutan, sebagai contoh paket dibandingkan dengan baris pertama dari *ACL*, kemudian baris kedua, ketiga, dan seterusnya.
- Paket hanya dibandingkan baris-baris *ACL* sampai terjadi kecocokan. Ketika paket cocok dengan kondisi pada baris *ACL*, paket akan ditindak lanjuti dan tidak ada lagi kelanjutan perbandingan.

- Terdapat *statement* "tolak" yang tersembunyi *(impilicit deny)* pada setiap akhir baris *ACL*, ini artinya bila suatu paket tidak cocok dengan semua baris kondisi pada *ACL*, paket tersebut akan ditolak

3.3.1 Jenis ACL

- ***Standard ACL***

Standard ACL hanya menggunakan alamat sumber *IP* di dalam paket *IP* sebagai kondisi yang ditest. Semua keputusan dibuat berdasarkan alamat *IP* sumber. Ini artinya, standard *ACL* pada dasarnya melewatkan atau menolak seluruh paket protokol. *ACL* ini tidak membedakan tipe dari lalu lintas *IP* seperti *WWW, telnet, UDP, DSP*.

- ***Extended ACL***

Extended ACL bisa mengevaluasi banyak *field* lain pada *header* layer 3 dan layer 4 pada paket *IP*. *ACL* ini bisa mengevaluasi alamat *IP* sumber dan tujuan, *field* protokol pada *header network layer* dan nomor *port* pada *header transport layer*. Ini memberikan *extended* ACL kemampuan untuk membuat keputusan-keputusan lebih spesifik ketika mengontrol lalu lintas.

3.3.2 Jenis Lalu Lintas ACL

- ***Inbound ACL***

Ketika sebuah *ACL* diterapkan pada paket *inbound* di sebuah interface, paket tersebut diproses melalui *ACL* sebelum di-*route* ke *outbound interface*. Setiap paket yang ditolak tidak bisa di-*route* karena paket ini diabaikan sebelum proses routing diabaikan.

- ***Outbond ACL***

Ketika sebuah *ACL* diterapkan pada paket *outbound* pada sebuah *interface*, paket tersebut di-*route* ke *outbound interface* dan diproses melalui *ACL* malalui antrian.