TEKNIK KEAMANAN JARINGAN DAN DATA DENGAN LINUX DEMILITARIZED ZONE

Jufriadif Na'am

Dosen Tetap Universitas Putra Indonesia YPTK Padang
Padang – Sumatera Barat – Indonesia

Abstrac*t*

In a computer network, must have its own security. It acts to protect the data that is considered imPortant, because to be closed and confidential. Topology, a good placement strategy and a clear calculation can provide added value to data security and server. Linux is touted to have all the above criteria. In addition to the level of stability and the ease and consider things in Linux because it is opensource. In a foreign country. It's not taboo anymore to use Linux as their data centers. Because in addition to safety, service is also easy. Therefore Linux precedence over the priority to build a server.

Keywords: Linux, Windows, Server, Kernel, Opensource, MikroTik, PORT

1. PENDAHULUAN

Zona demiliterisasi (DMZ) dapat digunakan untuk mengamankan jaringan *internal* dari akses *eksternal*. Kita juga dapat menggunakan *firewallLinux* untuk membuat DMZ. Ada banyak cara yang berbeda untuk merancang sebuah jaringan dengan DMZ. Metodedasar adalah dengan menggunakan *firewallLinux* tunggal.
Seorang *Network Engineer*, diharapkan mampu untuk mengamankan jaringannya dari serangan dari luar maupun dari dalam jaringan. Banyak cara atau metode yang dapat di implementasikan, tergantung dari kemampuan seorang *Network Engineer*. Salah satu cara yang paling baik adalah dengan mengamankan*server* pada jaringan dengan teknik *Demilitarized Zone (DMZ)*. Ada banyak cara pula untuk mengimplementasikan DMZ ini ke dalam jaringan kita. Seperti yang kitaketahui ada 2 (dua) jenis *firewall* yaitu secara *hardware* maupun *software*. Untuk versi *hardware* ada beberapa*firewall* papan atas seperti contoh *Cisco PIXFirewall*.Sedangkan untuk *firewallsoftware* seperti *Black Ice*, memang sangat *powerfull*. Tapi membeli *softwareBlack ICE* cukup memberatkan juga. Karena itu kita tawarkan *firewall* murah meriah dengan menggunakan *Linux*. Dan ternyata *Linux* ini cukup *capable* untuk di jadikan *firewall* DMZ.

2. KONSEP DASAR JARINGAN KOMPUTER

Network atau jaringan, dalam bidang komputer dapat diartikan sebagai dua buah komputer atau lebih yang dihubungkan melalui media penghubung sehingga dapat saling berkomunikasi, sehingga akan menimbulkan suatu effisiensi, sentralisasi dan

optimasi kerja. Pada jaringan komputer, yang dikomunikasikan adalah data dalam bit-bit biner. Komputer yang mengirimkan data disebut transmitter sedangkan komputer penerima disebut receiver.

Jaringan komputer tentunya memiliki beberapa manfaat dibandingkan dengan komputer yang berdiri sendiri. Manfaat-manfaat tersebut antara lain adalah sebagai berikut:

1. Berbagi sumber daya (sharing resources)
   Berbagi sumber daya bertujuan agar seluruh data, aplikasi, atau peripheral lainnya dapat dimanfaatkan oleh setiap orang yang tergabung pada jaringan tanpa terpengaruh lokasi maupun pengaruh dari pemakai.
2. Media komunikasi
   Jaringan komputer memungkinkan terjadinya komunikasi antara pengguna, dengan demikian orang-orang yang jaraknya berjauhan akan lebih mudah bekerja sama.
3. Integrasi data
   Pembangunan jaringan komputer dapat mencegah ketergantungan pada komputer pusat, karena setiap proses data tidak harus dilakukan pada satu komputer saja, melainkan dapat didistribusikan ke tempat lainnya. Oleh sebab inilah maka dapat terbentuk integrasi data yang memudahkan pemakai untuk memperoleh dan mengolah informasi setiap saat.
4. Keamanan data
   Sistem Jaringan Komputer dapat memberikan perlindungan terhadap data. Karena pemberian dan pengaturan hak akses kepada para pemakai, serta teknik perlindungan terhadap hard disk sehingga data mendapatkan perlindungan yang efektif.
5. Sumber daya lebih efisien dan informasi terkini (*up-to-date*)
   Dengan pemakaian sumber daya secara bersama-sama, akan mendapatkan hasil yang maksimal dan kualitas yang tinggi. Selain itu data atau informasi yang diakses selalu terbaru, karena setiap ada perubahan yang terjadi dapat segera langsung diketahui oleh setiap pemakai.

2.1. Network Address

Network Address digunakan untuk mengenali suatu *network* pada jaringan*Internet*. Misalkan sebuah *host* memiliki *IP Address* kelas B 165.72.1.2. *NetworkAddress* dari *host* tersebut adalah 165.72.0.0. *Address* ini didapat dengan membuat seluruh bit *host* pada 2 segmen terakhir menjadi 0. Tujuannya adalah untuk menyederhanakan informasi *routing* pada *Internet*. *Router* cukup melihat *networkAddress* (165.72) untuk menentukan ke *Router* mana data harus dikirimkan. Analoginya mirip dengan dalam proses pengantaran surat, petugas penyortir pada kantor pos cukup melihat kota tujuan pada alamat surat dan tidak perlu membaca seluruh alamat untuk menentukan jalur mana yang harus ditempuh surat tersebut.

2.2. Broadcast Address

Broadcast *Address* merupakan sebuah *IP Address* dalam sebuah jaringan yang digunakan untuk mengirim atau menerima informasi yangsama yang harus diketahui oleh seluruh *host* yang ada pada suatu jaringan. Setiap paket IP memiliki

header alamat tujuan berupa *IP Address* dari *host*yang akan dituju oleh paket tersebut. Dengan adanya alamat ini, maka hanya *host* tujuan saja yang memproses paket tersebut, sedangkan *host* lain akan mengabaikannya. Bagaimana jika suatu *host* ingin mengirim paket kepada seluruh *host* yang ada pada jaringannya? Tidak efisien jika ia harus membuat replikasi paket sebanyak jumlah *host* tujuan. Pemakaian *bandwidth* akan meningkat dan beban kerja *host* pengirim bertambah, padahal isi paket-paket tersebut sama.

Oleh karena itu dibuat konsep broadcast *Address* di mana sebuah *host* cukup mengirim ke alamat broadcast, maka seluruh *host* yang ada pada *network* akan menerima paket tersebut. Konsekuensinya, seluruh *host* pada jaringan yang sama harus memiliki broadcast *Address* yang sama dan alamat tersebut tidak boleh digunakan sebagai nomor IP untuk *host* tertentu. Jadi, sebenarnya setiap *host* memiliki dua alamat untuk menerima paket: pertama adalah nomor IP yang bersifat unik dan kedua adalah broadcast *Address* pada jaringan tempat *host* tersebut berada.

2.3. Gateway Address

Secara fisik setiap *host* dalam suatu jaringan terhubung melalui sebuah media penghantar, baik melalui kabel (wired) ataupun tanpa kabel (wireless). Jika salah satu *host* dalam jaringan A ingin mengirim data pada salah satu *host* di jaringan B, maka dibutuhkanlah sebuah *IP Address* yang dapat menghubungkan antara jaringan A dan jaringan B. Sebuah *host* pada jaringan A dapat berkomunikasi dengan *host* yang terdapat pada jaringan B, jika tersedia gateway yang dapat berkomunikasi dengan *host* pada jaringan B.

2.4. Subnetting

Jumlah *IP Address* sangat terbatas, apalagi jika harus memberikan alamat semua *host* di *internet*. Oleh karena itu perlu dilakukan efisiensi dalam penggunaan *IP Address* agar dapat mengalamati semaksimal mungkin *host* yang ada dalam satu jaringan. Konsep *subnetting* dan *IP Address* merupakan teknik yang umum digunakan di *internet* untuk mengefisienkan alokasi *IP Address* dalam sebuah jaringan supaya dapat mengoptimalkan penggunaan *IP Address*. *Routing* dan konsekuensi logis lainnya akan terjadi dengan lebih efisien dengan metoda *subnetting* yang baik. Untuk beberapa alasan yang menyakut efisiensi *IP Address*, mengatasi masalah topologi *network* dan organisasi, *networkadmnistrator* biasanya melakukan *subnetting*.

Esensi dari *subnetting* adalah memindahkan garis pemisah antara bagian *network* dan bagian *host* dari suatu *IP Address*. Beberapa bit dari bagian *host* dialokasikan menjadi bit tambahan pada bagian *network*. *Address* satu *network* menurut struktur baku dipecah menjadi beberapa sub*network*. Cara ini menciptakan sejumlah *network* tambahan dengan mengurangi jumlah maksimal *host* yang ada dalam setiap *network* tersebut.

Subnet diciptakan untuk membatasi cakupan lalu lintas broadcast (broadcast traffic), untuk menerapkan langkah-langkah keamanan jaringan, untuk memisahkan segmen jaringan berdasarkan fungsi, dan untuk membantu dalam menyelesaikan masalah kemacetan jaringan (*networkcongestion*). Untuk lebih memahami masalah *subnetting* dapat mengikuti contoh kasus berikut ini.

2.5. Routing

Routing adalah proses dimana suatu item dapat sampai ke tujuan dari satu lokasi ke lokasi lain. Beberapa contoh item yang dapat di-*routing* : mail, telepon call, dan data. Di dalam jaringan, *Router* adalah perangkat yang digunakan untuk melakukan *routing* trafik. *Router* atau perangkat-perangkat lain yang dapat melakukan fungsi *routing*, membutuhkan informasi sebagai berikut :

1 Alamat *Tujuan/DestinationAddress* - Tujuan atau alamat item yang akan di*routing*
2 Mengenal sumber informasi - Dari mana sumber (*Router* lain) yang dapat dipelajari oleh *Router* dan memberikan jalur sampai ke tujuan.
3 Menemukan rute - Rute atau jalur mana yang mungkin diambil sampai ke tujuan.
4 Pemilihan rute - Rute yang terbaik yang diambil untuk sampai ke tujuan.
5 Menjaga informasi *routing* - Suatu cara untuk menjaga jalur sampai ke tujuan yang sudah diketahui dan paling sering dilalui.

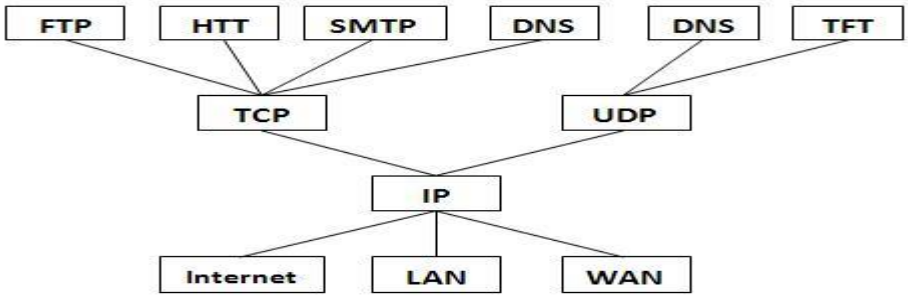
Jika jaringan tujuan tidak terhubung langsung di badan *Router*, *Router* harus mempelajari rute terbaik yang akan digunakan untuk meneruskan paket. Informasi ini dapat dipelajari dengan cara :

1. Manual oleh "*networkAdministrator*"
2. Pengumpulan informasi melalui proses dinamik dalam jaringan.

Ada dua cara untuk memberitahu *Router* bagaimana cara meneruskan paket ke jaringan yang tidak terhubung langsung *(not directly connected)* di badan *Router*. Dua metode untuk mempelajari rute melalui jaringan adalah :

1. *Rute Statik* - *Rute* yang dipelajari oleh *Router* ketika seorang *Administrator* membentuk rute secara manual. *Administrator* harus memperbarui atau meng"*update*" rute statik ini secara manual ketika terjadi perubahan topologi antar jaringan (*internetwork*).
2. Rute Dinamik - Rute secara Dinamik dipelajari oleh *Router* setelah seorang *Administrator* mengkonfigurasi sebuah protokol *routing* yang membantu menentukan rute. Tidak seperti rute Statik, pada rute Dinamik, sekali seorang *Administrator* jaringan mengaktifkan rute Dinamik, maka rute akan diketahui dan di*update* secara otomatis oleh sebuah proses *routing* ketika terjadi perubahan topologi jaringan yang diterima dari "*internetwork*".

2.6. Manajemen Port



Gambar 1– Manajement *Port*

Dalam hal akses, baik ke komputer lokal maupun ke komputer *server*, kita mengenal istilah *Port*, dan berikut adalah fungsi-fungsi dari *Port* tersebut :

1. *Port* 80, *WebServerPort* ini biasanya digunakan untuk web *server*, jadi ketika *user* mengetikan alamat IP atau *host*name di *web broeser* maka *web browser* akan melihat IP tsb pada *Port* 80,
2. *Port* 81, Web *ServerAlternatif* ketika *Port* 80 diblok maka *Port* 81 akan digunakan sebagai *Port altenatifhost*ing website
3. *Port* 21, FTP *Server* Ketika seseorang mengakses FTP *server*, maka ftp client secara default akan melakukan koneksi melalui *Port* 21 dengan ftp *server*
4. *Port* 22, SSH *Secure Shell Port* ini digunakan untuk *Port SSH*
5. *Port* 23, Telnet Jika anda menjalankan *server* telnet maka *Port* ini digunakan client telnet untuk hubungan dengan *server* telnet
6. *Port* 25, SMTP(Simple Mail Trans*Port* Protokol) Ketika seseorang mengirim email ke *server* SMTP anda, maka *Port* yg digunakan adalah *Port* 25
7. *Port* 2525 SMTP Alternate *ServerPort* 2525 adalah *Port* alternatifi aktif dari TZO untuk men*serviceforwarding* email. *Port* ini bukan standard *Port*, namun dapat diguunakan apabila *Port* smtp terkena blok.
8. *Port* 110, POP *Server* Jika anda menggunakan Mail *server*, *user* jika log ke dalam mesin tersebut via POP3 (Post Office Protokol) atau IMAP4 (*Internet* Message Access Protocol) untuk menerima emailnya, POP3 merupakan protokol untuk mengakses *mail box*
9. *Port* 119, News (NNTP) *Server*
10. *Port* 3389, *Remote* Desktop *Port* ini adalah untuk *remote* desktop di WinXP
11. *Port 389, LDAP Server LDAP or Lightweight Directory Access Protocol is becoming popular for Directory access, or Name, Telephone, Address directories. For Example*LDAP://LDAP.Bigfoot.Com *is a LDAP directory server.*
12. *Port 143, IMAP4 Server IMAP4 or Internet Message Access Protocol is becoming more popular and is used to retrieve Internet Mail from a remoteserver. It is more disk intensive, since all messages are stored on the server, but it allows for easy online, offline and disconnected use.*
13. *Port 443, Secure Sockets Layer (SSL) Server When you run a secure server, SSL Clients wanting to connect to your Secure server will connect on Port*

14. *443. This Port needs to be open to run your own Secure Transaction server. Port 445, SMB over IP, File Sharing Kelemahan windows yg membuka Port ini. biasanya Port ini digunakan sebagai Port file sharing termasuk printer sharing, Port inin mudah dimasukin virus atau Worm dan sebangsanya*
15. *Ports 1503 and 1720 Microsoft NetMeeting and VOIP MS NetMeeting and other VOIP allows you to host an Internet call or VideoConference with other 16. NetMeeting or VOIP users.*
16. *Port 5631, PCAnywhere When a PCAnywhere server is set up to receive remote requests, it listens on TCP Port 5631. This allow you to run a PCAnywhere host and use the Internet to connect back and remotely control your PC.*
17. *Port 5900, Virtual Network Computing (VNC) When you run an VNC server to remotely control your PC, it uses Port 5900. VNC is useful if you wish to remotely control your server.*
18. *Port 111, Portmap*
19. *Port 3306, Mysql*

3. KONSEP DASAR DMZ

De-Militarised Zone(DMZ) merupakan mekanisme untuk melindungi sistem internal dari serangan *hacker* atau pihak-pihak lain yang ingin memasuki sistem tanpa mempunyai hak akses. Sehingga karena DMZ dapat diakses oleh pengguna yang tidak mempunyai hak, maka DMZ tidak mengandung rule. Secara esensial, DMZ melakukan perpindahan semua layanan suatu jaringan ke jaringan lain yang berbeda. DMZ terdiri dari semua *Port* terbuka, yang dapat dilihat oleh pihak luar. Sehingga jika *hacker* menyerang dan melakukan *cracking* pada *server* yang mempunyai DMZ, maka *hacker* tersebut hanya dapat mengakses *host* yang berada pada DMZ, tidak pada jaringan internal.

Misalnya jika seorang pengguna bekerja di atas *server* FTP pada jaringan terbuka untuk melakukan akses publik seperti akses *internet*, maka *hacker* dapat melakukan *cracking* pada *server* FTP dengan memanfaatkan layanan *Network* Interconnection *System* (NIS), dan *Network* File *System*(NFS). Sehingga *hacker* tersebut dapat mengakses seluruh sumber daya jaringan, atau jika tidak, akses jaringan dapat dilakukan dengan sedikit upaya, yaitu dengan menangkap paket yang beredar di jaringan, atau dengan metoda yang lain. Namun dengan menggunakan lokasi *server* FTP yang berbeda, maka *hacker* hanya dapat mengakses DMZ tanpa mempengaruhi sumber daya jaringan yang lain. Selain itu dengan melakukan pemotongan jalur komunikasi pada jaringan internal, *trojan* dan sejenisnya tidak dapat lagi memasuki jaringan. Makalah ini akan membahas bagaimana memberi hak pada pengguna baik internal maupun eksternal, pada semua layanan jaringan yang diperlukan.

Konsep NAT, PAT, dan Daftar Akses. *NetworkAddress Translation(NAT)* berfungsi untuk mengarahkan alamat riil, seperti alamat *internet*, ke bentuk alamat internal. Misalnya alamat riil 203.8.90.100 dapat diarahkan ke bentuk alamat jaringan internal 192.168.0.1 secara otomatis dengan menggunakan NAT. Namun jika semua informasi secara otomatis ditranslasi ke bentuk alamat internal, maka tidak ada lagi kendali terhadap informasi yang masuk. Oleh karena itu maka muncullah PAT.

PortAddressTranslation(PAT) berfungsi untuk mengarahkan data yang masuk melalui *Port*, sekumpulan *Port* dan protokol, serta alamat IP pada *Port* atau sekumpulan post. Sehingga dapat dilakukan kendali ketat pada setiap data yang mengalir dari dan ke jaringan. Daftar Akses melakukan layanan pada pengguna agar dapat mengendalikan data jaringan. Daftar Akses dapat menolak atau menerima akses dengan berdasar pada alamat IP, alamat IP tujuan, dan tipe protokol.

4. KONSEP DASAR KEAMANAN

Sekuriti adalah Segala sesuatu yang mengenai keamanan, sedangkan Komputer adalah Suatu sistem yang meliputi *CPU (Prosesor),* Memori, I/O Device, dan sebagainya. Jadi Sekuriti Komputer adalah Segala sesuatu yang mengenai keamanan bagi Sistem Komputer Filosofi (dasar pemikiran) Keamanan Komputer Agar dapat mengamankan sistem komputer dengan benar, maka kita harus tahu karakteristik penganggu yang akan mendatangi sisten komputer kita.

a. Komponen sistem Komputer :Perangkat Keras (*Hardware*)
   Misalnya : dari pencurian, dll
b. Perangkat Lunak (*Software*)
   Misalnya : dari serangan *virus*, harckres, dll
c. Perangkat Manusia (Brainware)
   Misalnya : Pembajakan tenaga kerja

Ada 2 pihak yang terkait

Tabel 1 – Pihak Dalam Koputer

Pihak yang diganggu (Sistem Komputer)		Pihak yang mengganggu	
a.	Perangkat lunak	1.	Lingkungan
b.	Perangkat keras	2.	Fisika
c.	Perangkat manusia	3.	Kimia
d.	Manajemen	4.	Manajemen
e.	Basis Data	5.	Organisasi
f.	Operasional	6.	Perangkat Lunak & Keras
g.	Fisik	7.	Sistem Operasi
		8.	Telekomunikasi

4.1. Attacks (Serangan)

Kejadian yang sering menyebabkan data disadap atau tercuri bahkan terkena *virus* adalah akibat *password* terbuka karena adanya pencurian, catatan yang tercecer, pengamatan (cara mengetik, mengintip paket). Oleh karena itu cara yang dapat dilakukan adalah dengan membelokkan akses yaitu dengan mengganti ip, dns, atau route membelokkan akses ke *server* palsu untuk menjebak *password*.

Serangan yang terjadi kadangkala juga disebabkan karena kesalahan program. Seperti pepatah yang mengatakan tak ada gading yang tak retak.Oleh karena itu diberikan kebijakan keamanan jaringan yaitu dilarang menjalankan program yang tak

diketahui karena penyebaran *virus* dapat dengan mudah melalui *email, java script, vb script* akibatnya membebani *server* dengan akses yang besar.

Cara penyerangan yang juga perlu diwaspadai adalah batu loncatan. Biasanya akses dari komputer yang terletak di *intranet* kurang dibatasi. Apabila akses ke komputer di *intranet* terbuka, maka pemakai *internet* dapat masuk ke dalam komputer di *intranet*, kemudian menggunakan komputer tersebut sebagai batu loncatan. Oleh karena itu pembatasan pemakaian komputer terhadap orang yang tidak berwenang dilakukan dengan mengunci pintu ruangan setiap keluar dan masuk ruangan

4.2. Monitoring

Untuk mengamankan jaringan maka penting sekali dilakukan monitoring. Karena monitoring merupakan cara untuk mengetahui apa yang terjadi sebagai tindakan preventif dengan membaca catatan *system*. Karena *server* menggunakan *LINUX* maka dapat dilihat catatan *system*nya yang biasanya disimpan dalam directory /var/log.

/var/log/messages	Pesan-pesan dari sistem
/var/log/maillog	Transaksi email (SMTP)

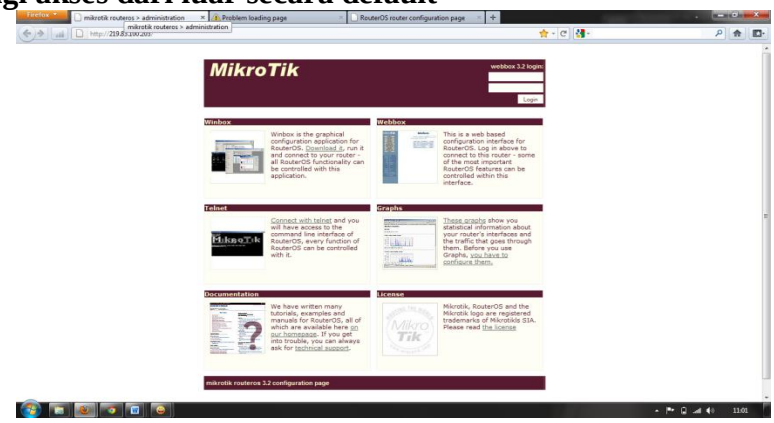
4.3. Komunikasi Terenkripsi

Komunikasi melalui jaringan publik memungkinkan adanya penyadap ikut mendengarkan percakapan oleh karena itu jalur komunikasi harus dienkripsi. Namun ada konsekuensi akibat enkripsi yaitu data yang dipertukarkan lebih besar. Dibawah ini merupakan beberapa *software* yang menjadi pilihan agar komunikasi dapat terenkripsi sehingga dapat diyakinkan keamanan jaringannya yaitu:

1.	Secure Shell	: pengganti telnet dengan enkripsi
2.	HTTPS	: secure HTTP

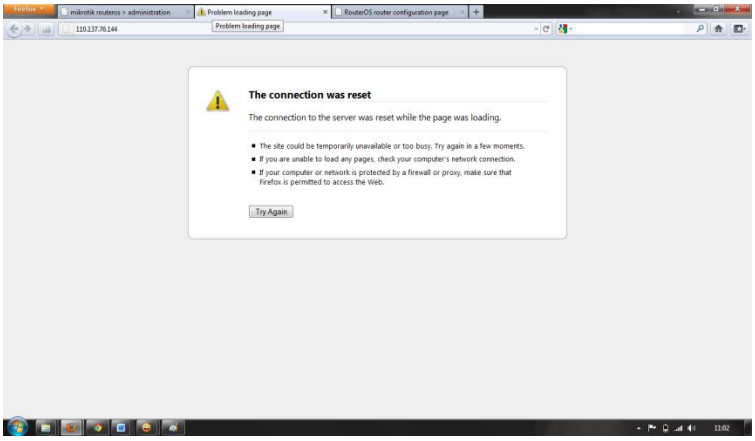
5. IMPLEMENTASI LINUX DEMILITARIZED ZONE (DMZ)

5.1. Lindungi akses dari luar secara default



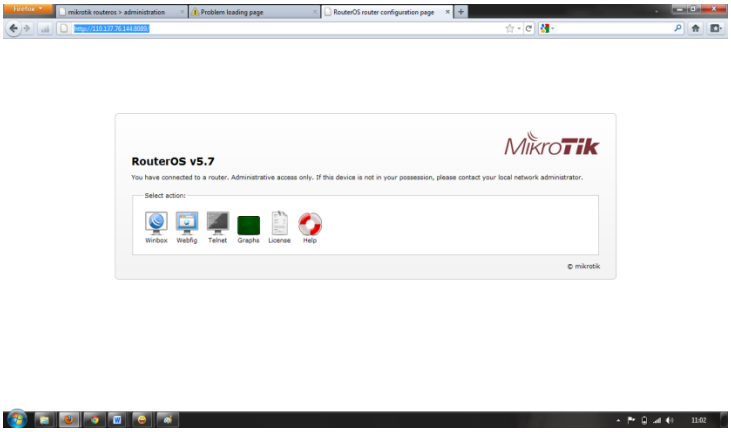
Gambar 2 *Router* Default

Ini memposisikan *Router* yang tidak aman, karna masih default.



Gambar 3 *Router* DI Hidden

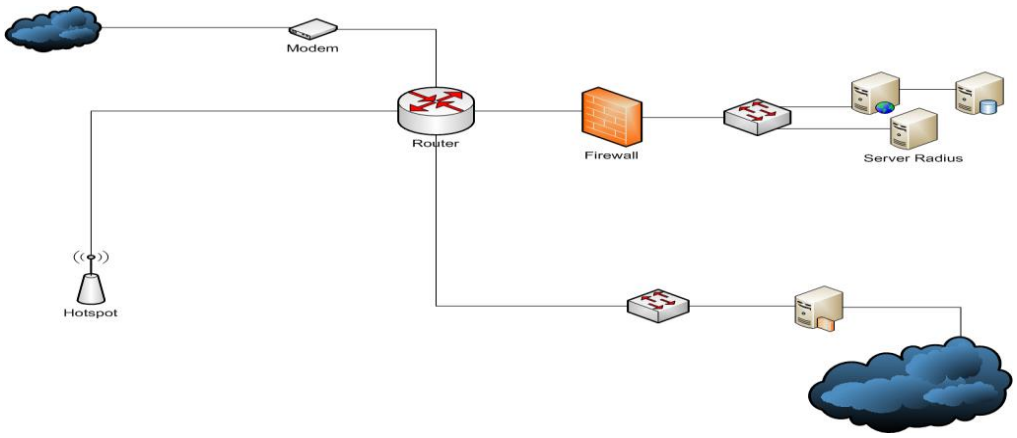
Pada halaman ini, telah melakukan perubahan pada *Port* default, seolah-olah *server* tersebut tidak tersedia. Tetapi di balik itu ada mesin yang menunggu dan sedang berjalan.



Gambar 4 Login *Router* Yang Aman

Ini adalah sistem yang telh di tingkatkan keamanan nya dari luar agar tidak dapat di akses sembarangan dari luar. Pada keterangan di atas penulis menggunakan *RouterMikroTik* RB 1100AH-1U

5.2. Atur Konsep Sesuai Topologi



Gambar 5 Topologi

Sesuai dengan konsep di atas seluruh data yang akan masuk / keluar dari *server* akan melewati *firewall*. Yang di belakang *firewall* terdapat aplikasi utama.

5.3.Firewall Linux Demilitarized Zone

Di sini menggunakan 3 buah kartu jaringan. Yang ber identitas eth0, eth1, dan eth2.

1. Eth0, di gunakan untuk jaringan *internal*
2. Eth1, di gunakan untuk *Ekstranet*
3. Eth2, di gunakan untuk *Server* yang di lindungi

Maka dengan skenario yang di tentukan di atas di perolelah *syntax* sebagai berikut

```
# forward traffic between DMZ and LAN

iptables -A FORWARD -i eth0 -o eth2 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i eth2 -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

# forward traffic between DMZ and WAN servers SMTP, Mail etc

iptables -A FORWARD -i eth2 -o eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i eth1 -o eth2 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Route incoming SMTP (Port 25 ) traffic to DMZ server 192.168.2.2

iptables -t nat -A PREROUTING -p tcp -i eth1 -d 202.54.1.1 --dPort 25 -j DNAT --to-destination
192.168.2.2

# Route incoming HTTP (Port 80 ) traffic to DMZ server load balancer IP 192.168.2.3

iptables -t nat -A PREROUTING -p tcp -i eth1 -d 202.54.1.1 --dPort 80 -j DNAT --to-destination
192.168.2.3

# Route incoming HTTPS (Port 443 ) traffic to DMZ server reverse load balancer IP 192.168.2.4

iptables -t nat -A PREROUTING -p tcp -i eth1 -d 202.54.1.1 --dPort 443 -j DNAT --to-destination
192.168.2.4
```

Jika menggunakan *MikroTik*, maka cara akan lebih mudah lagi. Contohnya :

```
/ip firewall nat
add action=netmap chain=dstnat comment="web" disabled=no dst-Address=\
[ip] dst-Port=9999 protocol=tcp to-Addresses=10.10.10.2 \
to-Ports=80
add action=netmap chain=dstnat comment="radius" disabled=no dst-Address=\
[ip] dst-Port=9999 protocol=tcp to-Addresses=10.10.10.3 \
to-Ports=80
add action=netmap chain=dstnat comment="proxy" disabled=no dst-Address=\
[ip] dst-Port=9999 protocol=tcp to-Addresses=10.10.10.4 \
to-Ports=80
```

5.4.Setting Firewall

Dengan *firewall* kita bisa melindungi akses yang akan keluar dan masuk pada *network* Rumah Sakit ini. *Firewall* bisa di bangun dengan *Linux* ataupun dapat di configurasi langsung pada *Router*. Configurasi *firewall* dengan *Linux*

```
#!/bin/sh
IPTABLES=/sbin/iptables
# Definisi komponen sistem untuk mempermudah perawatan.
# ----------------------------------------------------------
LOOPBACK_INTERFACE="lo"             # Interface Loopback
CLASS_D_MULTICAST="224.0.0.0/4?     # Class D multicast addr
CLASS_E_RESERVED_NET="240.0.0.0/5?  # Class E reserved addr
OSPF_MCAST="224.0.0.5?              # OSPF
OSPFD_MCAST="224.0.0.6?             # OSPFD
BROADCAST_src="0.0.0.0? mce_src="0.0.0.0?          # Broadcast source
addr
BROADCAST_DEST="255.255.255.255?    # Broadcast destination addr
PRIVPORTS="0:1023?                  # Privileged Port range
UNPRIVPORTS="1024:"                 # Unprivileged Port range
SSH_LOCAL_PORTS="1022:65535?        # Port range for local clients
SSH_REMOTE_PORTS="513:65535?        # Port range for remote clients
TRACEROUTE_SRC_PORTS="32769:65535?  # Port range sources for
traceroute
TRACEROUTE_DEST_PORTS="33434:33523? # Port range destination for
traceroute
# ----------------------------------------------------------

# Firewalls…. begins here!

# Kosongin semua aturan
$IPTABLES -P INPUT ACCEPT
$IPTABLES -P FORWARD ACCEPT
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -F

# Buat aturan firewall (DROP semua)
```

```
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT DROP

# Spesifik Rule Firewall
# Furtive Port scanner
$IPTABLES -A INPUT -p tcp –tcp-flags SYN,ACK,FIN,RST RST -m limit –limit 1/s -j
ACCEPT
$IPTABLES -A FORWARD -p tcp –tcp-flags SYN,ACK,FIN,RST RST -m limit –limit
1/s -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –tcp-flags SYN,ACK,FIN,RST RST -m limit –limit
1/s -j ACCEPT

# Batasi Paket Flooding
$IPTABLES -A INPUT -p tcp –syn -m limit –limit 1/s -j ACCEPT
$IPTABLES -A FORWARD -p tcp –syn -m limit –limit 1/s -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –syn -m limit –limit 1/s -j ACCEPT

# Batasi Ping of Death
$IPTABLES -A INPUT -p icmp -m length –length 512: -j DROP
$IPTABLES -A FORWARD -p icmp -m length –length 512: -j DROP
$IPTABLES -A OUTPUT -p icmp -m length –length 512: -j DROP

$IPTABLES -A INPUT -p icmp –icmp-type echo-request -m limit –limit 1/s -j
ACCEPT
$IPTABLES -A FORWARD -p icmp –icmp-type echo-request -m limit –limit 1/s -j
ACCEPT
$IPTABLES -A OUTPUT -p icmp –icmp-type echo-request -m limit –limit 1/s -j
ACCEPT

# Unlimited traffic on the loopback interface.
$IPTABLES -A INPUT -i $LOOPBACK_INTERFACE -j ACCEPT
$IPTABLES -A OUTPUT -o $LOOPBACK_INTERFACE -j ACCEPT

# OSPF
$IPTABLES -A INPUT -p ospf -j ACCEPT
$IPTABLES -A FORWARD -p ospf -j ACCEPT
$IPTABLES -A OUTPUT -p ospf -j ACCEPT

# GRE Tunneling
#$IPTABLES -A INPUT -p GRE -j ACCEPT
#$IPTABLES -A FORWARD -p GRE -j ACCEPT
#$IPTABLES -A OUTPUT -p GRE -j ACCEPT

# ICMP
$IPTABLES -A INPUT -p icmp -j ACCEPT
```

```
$IPTABLES -A FORWARD -p icmp -j ACCEPT
$IPTABLES -A OUTPUT -p icmp -j ACCEPT

# TRACEROUTE (-S 32769:65535 -D 33434:33523)
$IPTABLES -A INPUT -p udp –sPort $TRACEROUTE_SRC_PORTS –dPort
$TRACEROUTE_DEST_PORTS -j ACCEPT
$IPTABLES -A FORWARD -p udp –sPort $TRACEROUTE_SRC_PORTS –dPort
$TRACEROUTE_DEST_PORTS -j ACCEPT
$IPTABLES -A OUTPUT -p udp –sPort $TRACEROUTE_SRC_PORTS –dPort
$TRACEROUTE_DEST_PORTS -j ACCEPT

# Dynamic Routing (2600-2605)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 2600:2605 -j
ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 2600:2605 -j
ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 2600:2605 –dPort $UNPRIVPORTS -j
ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 2600:2605 –dPort $UNPRIVPORTS -j
ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 2600:2605 –dPort $UNPRIVPORTS -j
ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 2600:2605 -j
ACCEPT

# HTTP (80)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 80 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 80 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 80 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 80 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 80 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 80 -j ACCEPT

# WebCache (8080)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 8080 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 8080 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 8080 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 8080 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 8080 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 8080 -j ACCEPT

# DNS: full server (53)
$IPTABLES -A INPUT -p udp –sPort $UNPRIVPORTS –dPort 53 -j ACCEPT
```

```
$IPTABLES -A FORWARD -p udp –sPort $UNPRIVPORTS –dPort 53 -j ACCEPT
$IPTABLES -A OUTPUT -p udp –sPort 53 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p udp –sPort 53 –dPort 53 -j ACCEPT
$IPTABLES -A FORWARD -p udp –sPort 53 –dPort 53 -j ACCEPT
$IPTABLES -A OUTPUT -p udp –sPort 53 –dPort 53 -j ACCEPT

# DNS client (53)
$IPTABLES -A INPUT -p udp –sPort 53 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p udp –sPort 53 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p udp –sPort $UNPRIVPORTS –dPort 53 -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 53 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 53 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 53 -j ACCEPT

# DNS Zone Transfers (53)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 53 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 53 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 53 –dPort $UNPRIVPORTS -j ACCEPT

# HTTPS (443)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 443 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 443 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 443 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 443 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 443 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 443 -j ACCEPT

# MikroTik (3987)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 3987 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 3987 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 3987 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 3987 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 3987 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 3987 -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 8291 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 8291 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 8291 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 8291 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 8291 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 8291 -j ACCEPT
```

```
# SSH (22)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 22 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 22 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 22 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 22 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 22 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 22 -j ACCEPT

# FTP (20-21)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 20:1024 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 20:1024 -j
ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 20:1024 –dPort $UNPRIVPORTS -j
ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 20:21 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 20:21 –dPort $UNPRIVPORTS -j
ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 20:21 -j ACCEPT

# POP3 (110)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 110 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 110 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 110 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 110 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 110 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 110 -j ACCEPT

# Instant Messanger (5050)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 5050 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 5050 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 5050 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 5050 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 5050 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 5050 -j ACCEPT

# VoIP (5060)
$IPTABLES -A INPUT -p udp –sPort $UNPRIVPORTS –dPort 5060 -j ACCEPT
$IPTABLES -A FORWARD -p udp –sPort $UNPRIVPORTS –dPort 5060 -j ACCEPT
$IPTABLES -A OUTPUT -p udp –sPort 5060 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A INPUT -p udp –sPort 5060 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p udp –sPort 5060 –dPort $UNPRIVPORTS -j ACCEPT
```

```
$IPTABLES -A OUTPUT -p udp –sPort $UNPRIVPORTS –dPort 5060 -j ACCEPT

# SNMP (161)
$IPTABLES -A INPUT -p udp –sPort $UNPRIVPORTS –dPort 161 -j ACCEPT
$IPTABLES -A FORWARD -p udp –sPort $UNPRIVPORTS –dPort 161 -j ACCEPT
$IPTABLES -A OUTPUT -p udp –sPort 161 –dPort $UNPRIVPORTS -j ACCEPT
#$IPTABLES -A INPUT -p udp –sPort 161 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p udp –sPort 161 –dPort $UNPRIVPORTS -j ACCEPT
# IMAP over SSL (993)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 993 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 993 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 993 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 993 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 993 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 993 -j ACCEPT

# IMAP (143)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 143 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 143 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 143 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 143 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 143 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 143 -j ACCEPT

# QMQP (628)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 628 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 628 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 628 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 628 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 628 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 628 -j ACCEPT

# SMTP (25)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 25 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 25 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 25 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 25 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 25 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 25 -j ACCEPT

# IMAP (143)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 143 -j ACCEPT
```

```
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 143 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 143 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 143 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 143 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 143 -j ACCEPT

# QMQP (628)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 628 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 628 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 628 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 628 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 628 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 628 -j ACCEPT

# SMTP (25)
$IPTABLES -A INPUT -p tcp –sPort $UNPRIVPORTS –dPort 25 -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort $UNPRIVPORTS –dPort 25 -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort 25 –dPort $UNPRIVPORTS -j ACCEPT

$IPTABLES -A INPUT -p tcp –sPort 25 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A FORWARD -p tcp –sPort 25 –dPort $UNPRIVPORTS -j ACCEPT
$IPTABLES -A OUTPUT -p tcp –sPort $UNPRIVPORTS –dPort 25 -j ACCEPT
```

Jika menggunakan *MikroTik* maka *syntax* tersebut akan berbeda, namun hal dan rule serta aturannya lebih sederhana.

```
/ip firewall filter
add action=accept chain=input comment="Connection State" connection-state=established disabled=no
add action=accept chain=input comment="" connection-state=related disabled=no
add action=DROPchain=input comment="" connection-state=invalid disabled=no
add action=DROPchain=forward comment="Block Bogus IP Address" disabled=no src-Address=0.0.0.0/8
add action=DROPchain=forward comment="" disabled=no dst-Address=0.0.0.0/8
add action=DROPchain=forward comment="" disabled=no src-Address=127.0.0.0/8
add action=DROPchain=forward comment="" disabled=no dst-Address=127.0.0.0/8
add action=DROPchain=forward comment="" disabled=no src-Address=224.0.0.0/3
add action=DROPchain=forward comment="" disabled=no dst-Address=224.0.0.0/3
add action=DROPchain=input comment="DROP SSH brute forcers" disabled=no dst-Port=22 protocol=tcp src-Address-list=ssh_blacklist
add action=add-src-to-Address-list Address-list=ssh_blacklist Address-list-timeout=1w3d chain=input comment="" connection-state=new disabled=no dst-Port=22 protocol=tcp src-Address-list=ssh_stage3
```

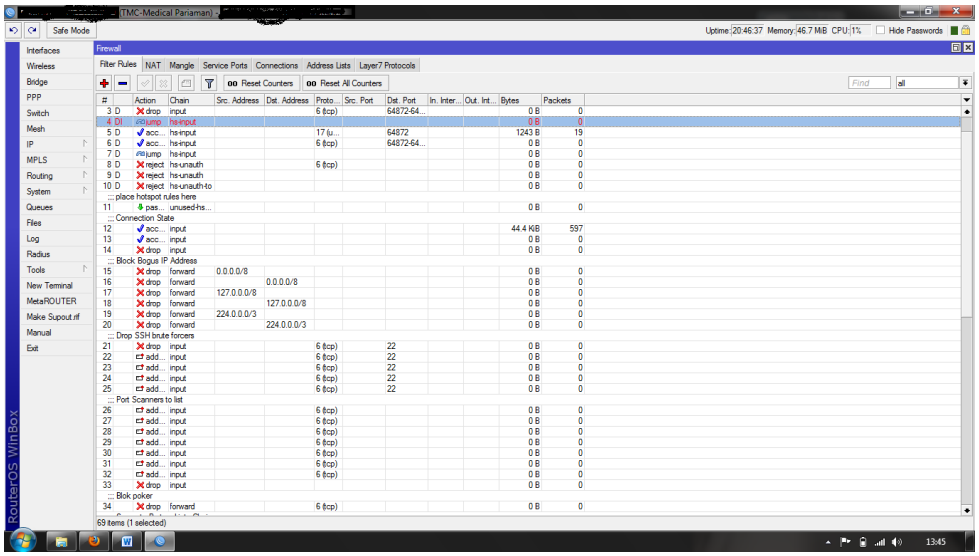
```
add action=add-src-to-Address-list Address-list=ssh_stage3 Address-list-timeout=1m chain=input comment="" connection-state=new disabled=no dst-Port=22 protocol=tcp src-Address-list=ssh_stage2
```

```
add action=add-src-to-Address-list Address-list=ssh_stage2 Address-list-timeout=1m
chain=input comment="" connection-state=new disabled=no dst-Port=22
protocol=tcp src-Address-list=ssh_stage1
add action=add-src-to-Address-list Address-list=ssh_stage1 Address-list-timeout=1m
chain=input comment="" connection-state=new \
    disabled=no dst-Port=22 protocol=tcp
add action=add-src-to-Address-list Address-list="Port scanners" \
Address-list-timeout=2w chain=input comment="Port Scanners to list " \
    disabled=no protocol=tcp psd=21,3s,3,1
add action=add-src-to-Address-list Address-list="Port scanners" \
Address-list-timeout=2w chain=input comment="" disabled=no protocol=tcp \
    tcp-flags=fin,!syn,!rst,!psh,!ack,!urg
add action=add-src-to-Address-list Address-list="Port scanners" \
Address-list-timeout=2w chain=input comment="" disabled=no protocol=tcp \
    tcp-flags=fin,syn
add action=add-src-to-Address-list Address-list="Port scanners" \
Address-list-timeout=2w chain=input comment="" disabled=no protocol=tcp \
    tcp-flags=syn,rst
add action=add-src-to-Address-list Address-list="Port scanners" \
Address-list-timeout=2w chain=input comment="" disabled=no protocol=tcp \
    tcp-flags=fin,psh,urg,!syn,!rst,!ack
add action=add-src-to-Address-list Address-list="Port scanners" \
Address-list-timeout=2w chain=input comment="" disabled=no protocol=tcp \
    tcp-flags=fin,syn,rst,psh,ack,urg
add action=add-src-to-Address-list Address-list="Port scanners" \
Address-list-timeout=2w chain=input comment="" disabled=no protocol=tcp \
    tcp-flags=!fin,!syn,!rst,!psh,!ack,!urg
add action=DROPchain=input comment="" disabled=no src-Address-list=\
    "Port scanners"
add action=DROPchain=forward comment="Blok poker" disabled=no Port=9339 \
    protocol=tcp
add action=jump chain=forward comment="Separate Protocol into Chains" \
    disabled=no jump-target=tcp protocol=tcp
add action=jump chain=forward comment="" disabled=no jump-target=udp \
    protocol=udp
add action=jump chain=forward comment="" disabled=no jump-target=icmp \
    protocol=icmp
add action=jump chain=input comment="" disabled=no jump-target=tcp protocol=\
    tcp
add action=jump chain=input comment="" disabled=no jump-target=udp protocol=\
    udp
add action=DROPchain=udp comment="Blocking UDP Packet" disabled=no dst-
Port=\
    69 protocol=udp
add action=DROPchain=udp comment="" disabled=no dst-Port=111 protocol=udp
add action=DROPchain=udp comment="" disabled=no dst-Port=135 protocol=udp
```

```
add action=DROPchain=udp comment="" disabled=no dst-Port=135-139 protocol=\
    udp
add action=DROPchain=udp comment="" disabled=no dst-Port=445 protocol=udp
add action=DROPchain=udp comment="" disabled=no dst-Port=2049 protocol=udp
add action=DROPchain=udp comment="" disabled=no dst-Port=3133 protocol=udp
add action=DROPchain=tcp comment="Bloking TCP Packet" disabled=no dst-
Port=\
    69 protocol=tcp
add action=add-src-to-Address-list Address-list=mail-virus \
Address-list-timeout=1h chain=tcp comment="" disabled=no dst-Port=25 \
    protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=25 protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=111 protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=119 protocol=tcp
add action=add-src-to-Address-list Address-list=virus-sharing \
Address-list-timeout=2m chain=tcp comment="" disabled=no dst-Port=135 \
    protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=135 protocol=tcp
add action=add-src-to-Address-list Address-list=virus-sharing \
Address-list-timeout=2m chain=tcp comment="" disabled=no dst-Port=137-139 \
    protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=137-139 protocol=\
    tcp
add action=add-src-to-Address-list Address-list=virus-conficker \
Address-list-timeout=2m chain=tcp comment="" disabled=no dst-Port=445 \
    protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=445 protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=2049 protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=12345-12346 \
    protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=20034 protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=3133 protocol=tcp
add action=DROPchain=tcp comment="" disabled=no dst-Port=67-68 protocol=tcp
add action=accept chain=icmp comment="Limited Ping Flood" disabled=no \
    icmp-options=0:0-255 limit=5,5 protocol=icmp
add action=accept chain=icmp comment="" disabled=no icmp-options=3:3 limit=\
    5,5 protocol=icmp
add action=accept chain=icmp comment="" disabled=no icmp-options=3:4 limit=\
    5,5 protocol=icmp
add action=accept chain=icmp comment="" disabled=no icmp-options=8:0-255 \
    limit=5,5 protocol=icmp
add action=accept chain=icmp comment="" disabled=no icmp-options=11:0-255 \
    limit=5,5 protocol=icmp
add action=DROPchain=icmp comment="" disabled=no protocol=icmp
add action=DROPchain=forward comment="DROP Traceroute" disabled=no \
    icmp-options=11:0 protocol=icmp
```

```
add action=DROPchain=forward comment="" disabled=no icmp-options=3:3 \
   protocol=icmp
```

maka kalau di *remote* dari sistem berbasis *GUI* dapat dilihat pada gambar 6 sebagai berikut



Gambar 6 *FirewallMikroTik*

Setelah kita amankan jaringan denga *firewall* maka langkah yang tidak kalah penting nya adalah dengan mengatur NAT. Inti dari fungsi NAT sendiri adalah untuk mengintegrasikan *host-host* dalam sebuah jaringan lokal supayabisa mengakses jaringan di luar *network*nya dengan hanyamenggunakan satu *IP Address* saja. Jadi dengan cara inilah *server* tidak perlu satu *class* dengan ip client, agar *server* lebih aman. Karena mengelompokkan *server* dalam satu *class* ip, tanpa perlu terkoneksi dengan jaringan lain. Berikut adalah salah satu *syntax* NAT

```
ip firewall nat add chain=dstnat dst-Address=ip protocol=tcp dst-Port=80
action=dst-nat to-Addresses=server to-Ports=80
```

6. KESIMPULAN

Kesimpulan yang dapat diambil dalam pengaman jaringan dan data dengan *Linux* Demilitarized Zone adalah sebagai berikut:

1. Dalam menentukan sistem operasi sebagai *server*penulis memilih *Ubuntuserver*. Karena Gratis, Stabil, aman, *Update* OS dan aplikasinya secara keseluruhan, tidak terlalu membutuhkan *hardware* berspesifikasi tinggi, tidak ada *virus*, *trojan* ataupun malware, sudah terdapat banyak sekali repository yang diletakkan pada *server-server* local, tidak perlu kesulitan dalam mencari crack atau keygen.
2. Untuk merancang sebuah jaringan harus diperhatikan tipe dan model jaringan apa yang akan dibuat.
3. Untuk membuat *Linux* Demilitarized Zone terlebih dahulu menentukan *software* yang akan mendukung sistem yang dibangun.

4. Dalam mensetting sebuah *Server* harus teliti, karena kalau tidak dapat menyebabkan kesalahan yang fatal.
5. Dari sisi pemilihan topologi yang akan digunakan kita harus melihat data teknis terlebih dahulu.

DAFTAR PUSTAKA

Andi, 2009, Langkah Mudah Administrasi Jaringan Menggunakan Linux Ubuntu 9.10, Yogyakarta.

Community, Ubuntu, Reborn 2011
<Url : Http://Www.Kaskus.Us/Showthread.Php?T=11214772> Juni 2011

Dmz, Linux, 2011, Linux Dmz
<Url : Http://Www.Cyberciti.Biz/Faq/Linux-Demilitarized-Zone-Howto/> September 2011.

Help, Ubuntu, 2011, Help Ubuntu <Url : Https://Help.Ubuntu.Com/> Setember 2011.

Jaringan, Topologi, 2011 <Url : Http://Id.Wikipedia.Org/Wiki/Topologi_Jaringan/> Juni 2011.

Kaskus, Debian, Community 2011
<Url : Http://Www.Kaskus.Us/Showthread.Php?T=2317660> Agustus 2011

Rpl, Rekayasa Perangkat Lunak, 2011
<Url : Http://Id.Wikipedia.Org/Wiki/Rekayasa_Perangkat_Lunak/> Agustus 2011

Sembiring, Jhony H, 2001, Jaringan Komputer Berbasis Linux, Bandung.

Solution, Linux, 2011, Linux Solution <Url : Http://Www.Linuxsolutions.Fr/> September 2011.

Ubuntu, Forums, 2011, Forum Ubuntu <Url : Http://Ubuntuforums.Org/> Oktober 2011.

Uml, 2011 <Url : Http://En.Wikipedia.Org/Wiki/Unified_Modeling_Language/> Agustus 2011.