BAB 11

SISTEM KEAMANAN JARINGAN (FIREWALL)

Tujuan:
Pembahasan ini bertujuan agar siswa dapat :
1. Menentukan jenis-jenis keamanan jaringan /firewall
2. Memasang Firewall
3. Mengidentifikasi pengendalian jaringan yang diperlukan
4. Mendesain sistem keamanan jaringan

Pokok Bahasan
Dalam pembahasan ini meliputi:
1. Jenis jenis keamanan jaringan, Firewall, Pengendalian jaringan,
2. Cara Mendesain system keamanan jaringan

11.1 FIREWALL

Dalam jaringan komputer, khususnya yang berkaitan dengan aplikasi yang melibatkan berbagai kepentingan, akan banyak terjadi hal yang dapat mengganggu kestabilan koneksi jaringan komputer tersebut, baik yang berkaitan dengan hardware (pengamanan fisik, sumber daya listrik) maupun yang berkaitan dengan software (sistem, konfigurasi, sistem akses, dll).

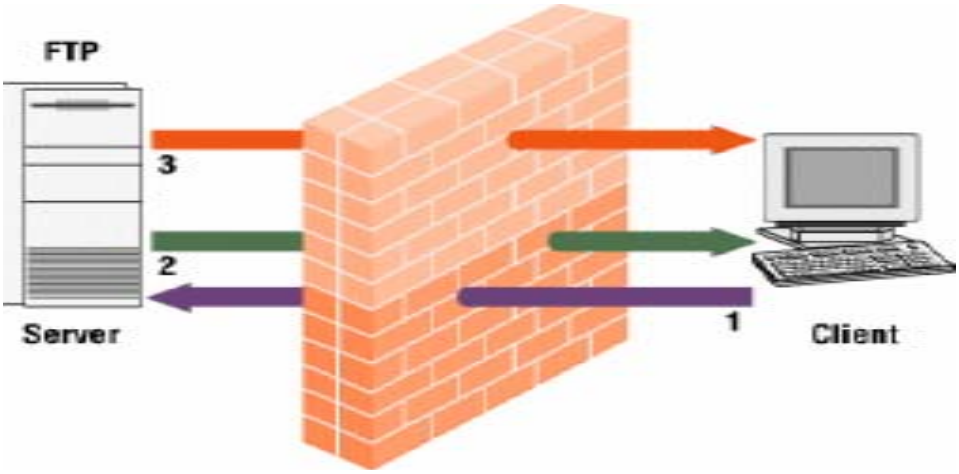
Gangguan pada sistem dapat terjadi karena faktor ketidaksengajaan yang dilakukan oleh pengelola (*human error*), akan tetapi tidak sedikit pula yang disebabkan oleh pihak ketiga.

Gangguan dapat berupa perusakan, penyusupan, pencurian hak akses, penyalahgunaan data maupun sistem, sampai tindakan kriminal melalui aplikasi jaringan komputer.

Pengamanan terhadap sistem hendaknya dilakukan sebelum sistem tersebut difungsikan. Percobaan koneksi (*trial*) sebaiknya dilakukan sebelum sistem yang sebenarnya difungsikan. Dalam melakukan persiapan fungsi sistem hendaknya disiapkan pengamanan dalam bentuk:

1. Memisahkan terminal yang difungsikan sebagai pengendali jaringan atau titik pusat akses (Server) pada suatu area yang digunakan untuk aplikasi tertentu.
2. Menyediakan pengamanan fisik berupa ruangan khusus untuk pengamanan perangkat yang disebut pada butir nomor 1. Ruangan tersebut dapat diberikan label Network Operating Center (NOC) dengan membatasi personil yang diperbolehkan masuk.
3. Memisahkan sumber daya listrik untuk NOC dari pemakaian yang lain. Hal ini untuk menjaga kestabilan fungsi sistem. Perlu juga difungsikan Uninteruptable Power Supply (UPS) dan Stabilizer untuk menjaga kestabilan supply listrik yang diperlukan perangkat pada NOC.
4. Merapikan wiring ruangan dan memberikan label serta pengklasifikasian kabel.
5. Memberikan Soft Security berupa Sistem Firewall pada perangkat yang difungsikan di jaringan.
6. Merencanakan maintenance dan menyiapkan Back Up sistem.

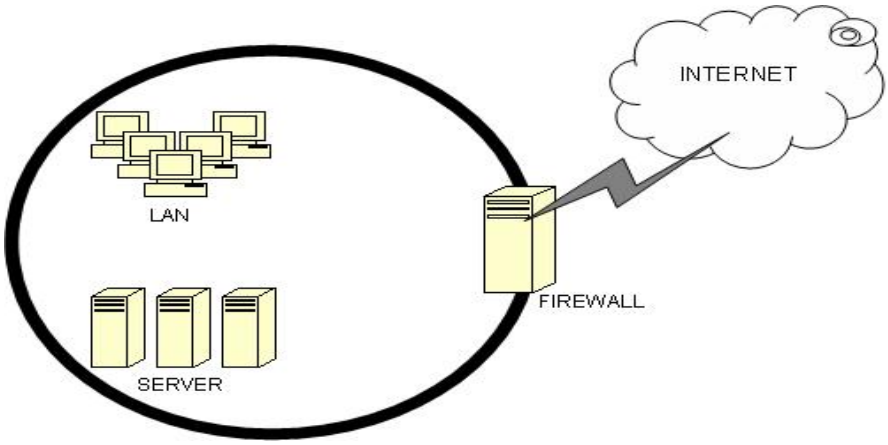
Firewall (Gambar 11.1) adalah salah satu aplikasi pada sistem operasi yang dibutuhkan oleh jaringan komputer untuk melindungi intergritas data/sistem jaringan dari serangan-serangan pihak yang tidak bertanggung jawab atau lalu lintas jaringan yang tidak aman. Caranya dengan melakukan filterisasi terhadap paket-paket yang melewatinya.



Gambar 11 - 1 Ilustrasi Penerapan Firewall

Firewall tersusun dari aturan-aturan yang diterapkan baik terhadap *hardware*, *software* ataupun sistem itu sendiri dengan tujuan untuk melindungi jaringan, baik dengan melakukan filterisasi, membatasi, ataupun menolak suatu permintaan koneksi dari jaringan luar lainnya seperti internet.

Oleh karena seringnya firewall digunakan untuk melindungi jaringannya, maka firewall juga berfungsi sebagai pintu penyangga antara jaringan yang dilindunginya dengan jaringan lainnya atau biasa disebut *gateway.*



Gambar 11 - 2 Arsitektur Firewall Pada Jaringan Komputer

Gambar 11.2 menunjukkan firewall yang melindungi jaringan lokal dengan cara mengendalikan aliran paket yang melewatinya. Firewall dirancang untuk mengendalikan aliran paket berdasarkan asal, tujuan, port dan informasi tipe paket.Firewall berisi sederet daftar aturan yang digunakan untuk menentukan nasib paket data yang datang atau pergi dari firewall menurut kriteria dan parameter tertentu. Semua paket yang diperiksa firewall akan melakukan mengalami perlakuan yang diterapkan pada *rule* atau *policy* yang diterapkan pada *chains firewall*. Masing-masing tabel dikenakan untuk tipe aktivitas paket tertentu dan dikendalikan oleh rantai aturan filter paket yang sesuai. Rantai (*chains*) adalah daftar aturan yang dibuat untuk mengendalikan paket.

Pada firewall terjadi beberapa proses yang memungkinkannya melindungi jaringan. Proses yang terjadi pada firewall ada tiga macam yaitu:

- Modifikasi header paket,
- Translasi alamat jaringan, dan
- Filter paket

Modifikasi header paket digunakan untuk memodifikasi kualitas layanan bit paket TCP sebelum mengalami proses routing.

Translasi alamat jaringan antara jaringan privat dan jaringan publik terjadi pada firewall.. Translasi yang terjadi dapat berupa translasi satu ke satu (*one to one*), yaitu satu alamat IP privat dipetakan kesatu alamat IP publik atau translasi banyak kesatu (*many to one*) yaitu beberapa alamat IP privat dipetakan kesatu alamat publik.

Filter paket digunakan untuk menentukan nasib paket apakah dapat diteruskan atau tidak.

11.2 Jenis-Jenis Firewall

Firewall dapat dibedakan berdasarkan caranya bekerja. Jenis-jenis firewall tersebut adalah:

1. Packet Filtering Gateway
2. Application Layer Gateway
3. Circuit Level Gateway
4. Statefull Multilayer Inspection Firewall

11.2.1. Packet Filtering Gateway

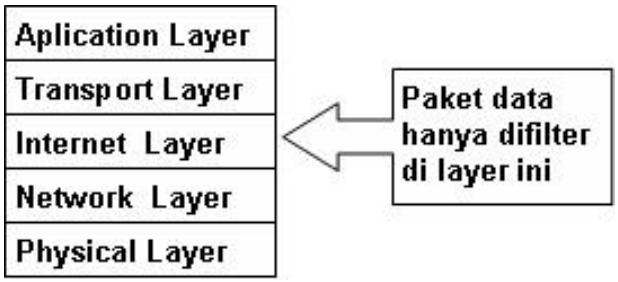
Packet filtering gateway dapat diartikan sebagai firewall yang bertugas melakukan filterisasi terhadap paket-paket yang datang dari luar jarigan yang dilindunginya.

Filterirasi paket ini hanya terbatas pada sumber paket, tujuan paket, dan atribut-atribut dari paket tersebut, misalnya paket tersebut bertujuan ke server kita yang menggunakan alamat IP 202.51.226.35 dengan port 80. Port 80 adalah atribut yang dimiliki oleh paket tersebut.

Seperti yang terlihat pada gambar 11.4, firewall tersebut akan melewatkan paket dengan tujuan ke Web Server yang menggunakan port 80 dan menolak paket yang menuju Web Server dengan port 23.

Bila kita lihat dari sisi arsitektur TCP/IP, firewall ini akan bekerja pada layer internet. Firewall ini biasanya merupakan bagian dari sebuah **router firewall.**

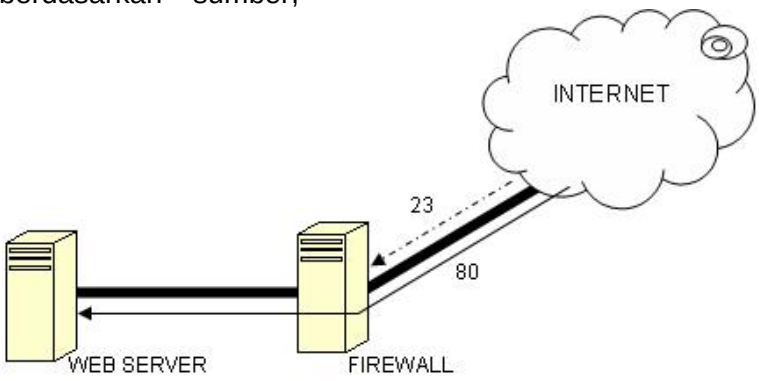
Software yang dapat digunakan untuk implementasi packet filtering diantaranya adalah *iptables* dan *ipfw*.



Gambar 11 - 3 Lapisan untuk Proses Packet Filtering Gateway

11.2.2. Application Layer Gateway

Model firewall ini juga dapat disebut Proxy Firewall. Mekanismenya tidak hanya berdasarkan sumber, tujuan dan atribut paket, tapi bisa mencapai isi (*content*) paket tersebut.

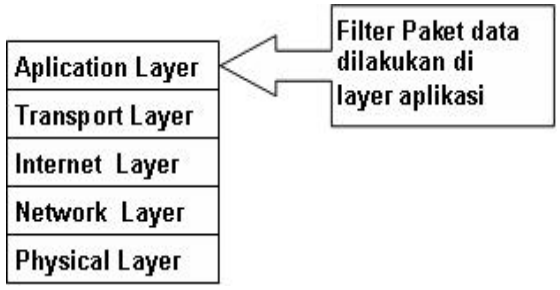


Gambar 11 - 4 Web server dengan Firewall

Mekanisme lainnya yang terjadi adalah paket tersebut tidak akan secara langsung sampai ke server tujuan, akan tetapi hanya sampai firewall saja.

Selebihnya firewall ini akan membuka koneksi baru ke server tujuan setelah paket tersebut diperiksa berdasarkan aturan yang berlaku.

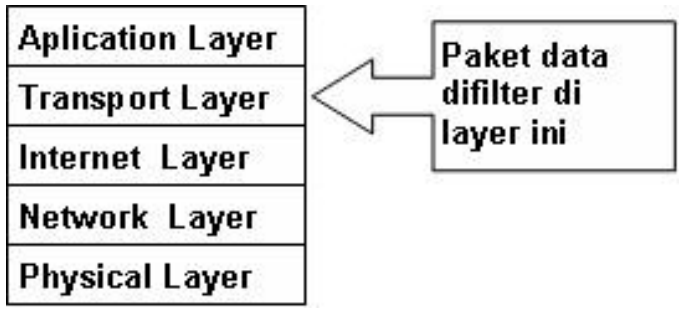
Bila kita melihat dari sisi layer TCP/IP, firewall jenis ini akan melakukan filterisasi pada layer aplikasi (*Application Layer*).



Gambar 11 - 5 Proxy Firewall dilihat pada Model TCP/IP

11.2.3. Circuit Level Gateway

Model firewall ini bekerja pada bagian Lapisan Transport model referensi TCP/IP. Firewall ini akan melakukan pengawasan terhadap awal hubungan TCP yang biasa disebut sebagai TCP Handshaking, yaitu proses untuk menentukan apakah sesi hubungan tersebut diperbolehkan atau tidak. Bentuknya hampir sama dengan *Application Layer Gateway*, hanya saja bagian yang difilter terdapat ada lapisan yang berbeda, yaitu berada pada layer Transport.

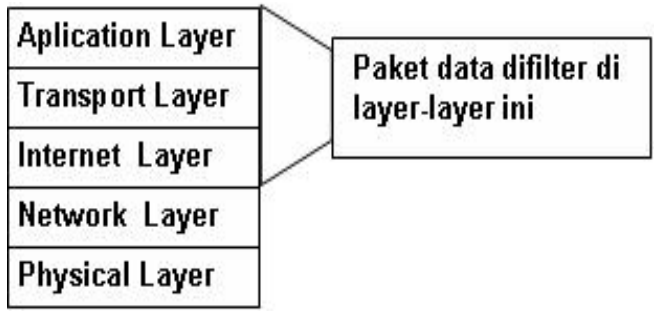


Gambar 11 - 6 Circuit Level Gateway dilihat pada Model TCP/IP

11.2.4. Statefull Multilayer Inspection Firewall

Model firewall ini merupakan penggabungan dari ketiga firewall sebelumnya. Firewall jenis ini akan bekerja pada lapisan Aplikasi, Transport dan Internet.

Dengan penggabungan ketiga model firewall yaitu *Packet Filtering Gateway, Application Layer Gateway* dan *Circuit Level Gateway,* mungkin dapat dikatakan firewall jenis ini merupakan firewall yang,memberikan fitur terbanyak dan memberikan tingkat keamanan yang paling tinggi.



Gambar 11 - 7 Statefull Multilayer Inspection Firewall dilihat pada Model TCP/IP

11.3 Pengendalian Jaringan

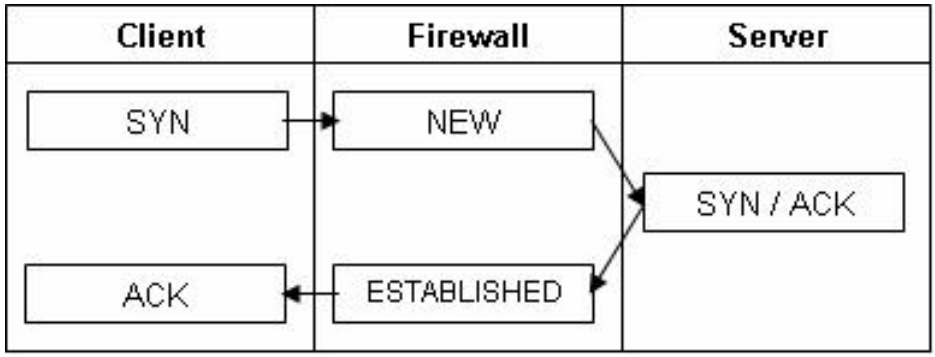
Dalam hal pengendalian jaringan dengan menggunakan firewall, ada dua hal yang harus diperhatikan yaitu koneksi firewall yang digunakan (dalam hal ini yang digunakan adalah koneksi TCP), dan konsep firewall yang diterapkan, yaitu *IPTables*.

Dengan dua hal ini diharapkan *firewall* dapat mengenali apakah koneksi yang ada berupa koneksi baru (*NEW)*, koneksi yang telah ada (*ESTABLISH*), koneksi yang memiliki relasi dengan koneksi lainnya (*RELATED*) atau koneksi yang tidak valid (*INVALID*). Keempat macam koneksi itulah yang membuat *IPTables* disebut *Statefull Protocol*.

11.3.1. Koneksi TCP

Sebuah koneksi TCP dikenal sebagai koneksi yang bersifat *Connection Oriented*, pada permulaan koneksi, sebuah klien akan mengirimkan sinyal *SYN* ke server tujuannya, selanjutnya proses pada firewall menganggap input ini sebagai paket baru yang akan di kirimkan ke server.

Server akan mengolah masukan tersebut, dan akan meneruskan ke tujuannya apabila paket tersebut diperbolehkan untuk lewat atau diterima selanjutnya menjadi paket ACK bagi klien. Namun apabila perlakukan bagi paket tersebut adalah menolak atau membuangnya, maka paket tidak akan di perlakukan seperti yang diminta oleh aturan pada firewall.



Gambar 11 - 8 Koneksi TCP Pada Firewall

Setelah sinyal tersebut diterima, pada setiap koneksi yang terjadi klien juga akan mengirimkan sinyal ACK kepada server. Pengenalan koneksi oleh firewall seperti *NEW, ESTABLISHED*, dan *RELATED* dikenal dengan nama ***connection tracking.***

Koneksi TCP juga dikenal sebagai koneksi yang reliable dan menggunakan mekanisme byte stream service. Konsep reliable pada koneksi TCP berarti TCP akan mendeteksi error pada paket yang dikirim dan bila itu terjadi paket akan dikirim kembali. Konsep byte stream service berarti paket-paket dikirim ke tujuan secara urut.

Setelah koneksi TCP selesai dilakukan, klien atau server akan mengirimkan signal *FIN/ACK* kepada mesin tujuannya.

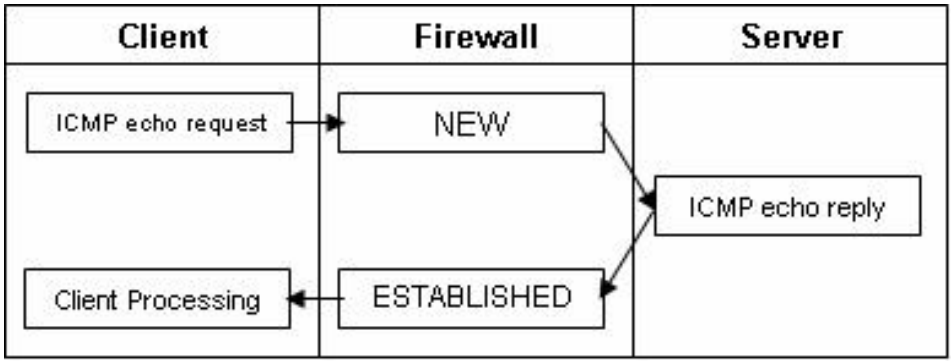
Sinyal ini masih dianggap sebagai koneksi yang sudah terjadi (*ESTASBLISHED*). Setelah mesin tujuannya menerima sinyal *FIN/ACK*, mesin tersebut akan membalas dengan sinyal *ACK* kepada mesin itu kembali dan koneksi akan terputus.

Protokol TCP mendominasi penggunaan aplikasi jaringan komputer, namun untuk penyelenggaraan jaringannya protocol IP yang memegang peranan.

Dlaam hal uji koneksi termasuk didalamnya monitoring jaringan, maka

ICMP (*Internet Control Message Protocol*) diimplmentasikan untuk keperluan ini. ICMP utamanya digunakan oleh sistem operasi komputer jaringan untuk mengirim pesan kesalahan yang menyatakan, sebagai contoh, bahwa komputer tujuan tidak bisa dijangkau. Salah satu aplikasi ICMP adalah tools ping yang digunakan untuk monitoring jaringan dengan mengirim pesan ICMP *Echo Request* (dan menerima *Echo Reply*) untuk menentukan apakah komputer tujuan dapat dijangkau dan berapa lama paket yang dikirimkan dibalas oleh komputer tujuan.

Sebuah koneksi ICMP (Gambar 11.10) hanyalah sebuah permintaan (*request*) echo dan balasannya (*reply*). Ada empat macam tipe echo yang akan mendapat paket balasan, yaitu *echo request* dan *reply, timestamp request* dan *reply, infomation request* dan *reply*, serta *address mask request* dan *reply*.



Gambar 11 - 9 Sebuah Koneksi ICMP

UDP (*User Datagram Protocol*), adalah salah satu protokol lapisan transport pada model referensi TCP/IP yang mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara host-host dalam jaringan yang menggunakan TCP/IP. Protokol ini didefinisikan dalam RFC 768.

UDP memiliki karakteristik-karakteristik berikut:

- ***Connectionless*** (tanpa koneksi): Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak berukar informasi.
- ***Unreliable*** (tidak andal): Pesan-pesan UDP akan dikirimkan sebagai datagram tanpa adanya nomor urut atau pesan acknowledgment. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan yang hilang selama transmisi. Umumnya, protokol lapisan aplikasi yang berjalan di atas UDP mengimplementasikan layanan keandalan mereka masing-masing, atau mengirim pesan secara periodik atau dengan menggunakan waktu yang telah didefinisikan.

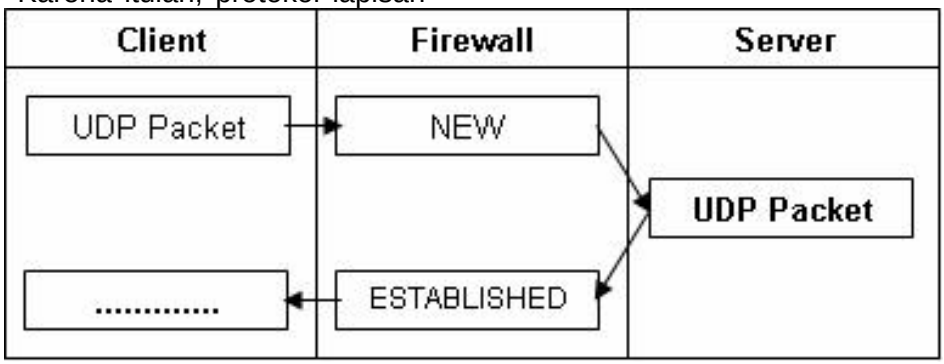
- UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah host dalam jaringan yang menggunakan TCP/IP. ***Header*** UDP berisi ***field*** Source Process Identification dan Destination Process Identification.
- UDP menyediakan penghitungan checksum berukuran 16-bit terhadap keseluruhan pesan UDP.

UDP tidak menyediakan layanan-layanan antar-host berikut:

- UDP tidak menyediakan mekanisme penyanggaan (buffering) dari data yang masuk ataupun data yang keluar. Tugas buffering merupakan tugas yang harus diimplementasikan oleh protokol lapisan aplikasi yang berjalan di atas UDP.
- UDP tidak menyediakan mekanisme segmentasi data yang besar ke dalam segmen-segmen data, seperti yang terjadi dalam protokol TCP. Karena itulah, protokol lapisan aplikasi yang berjalan di atas UDP harus mengirimkan data yang berukuran kecil (tidak lebih besar dari nilai Maximum Transfer Unit/MTU) yang dimiliki oleh sebuah antarmuka di mana data tersebut dikirim. Karena, jika ukuran paket data yang dikirim lebih besar dibandingkan nilai MTU, paket data yang dikirimkan bisa saja terpecah menjadi beberapa fragmen yang akhirnya tidak jadi terkirim dengan benar.
- UDP tidak menyediakan mekanisme ***flow-control***, seperti yang dimiliki oleh TCP.

koneksi UDP (Gambar 11.11) bersifat *connectionless*. Sebuah mesin yang mengirimkan paket UDP tidak akan mendeteksi kesalahan terhadap pengiriman paket tersebut.

Paket UDP tidak akan mengirimkan kembali paket-paket yang mengalami error. Model pengiriman paket ini akan lebih efisien pada koneksi *broadcasting* atau *multicasting*.



Gambar 11 - 10 Sebuah Koneksi UDP

Seperti halnya TCP, UDP juga memiliki saluran untuk mengirimkan informasi antar host, yang disebut dengan UDP Port**.** Untuk menggunakan protokol UDP, sebuah

aplikasi harus menyediakan alamat IP dan nomor UDP Port dari host yang dituju. Sebuah UDP port berfungsi sebagai sebuah *multiplexed message queue***,** yang berarti bahwa UDP port tersebut dapat menerima beberapa pesan secara sekaligus. Setiap port diidentifikasi dengan nomor yang unik, seperti halnya TCP, tetapi meskipun begitu, UDP Port berbeda dengan TCP Port meskipun memiliki nomor port yang sama. Tabel di bawah ini mendaftarkan beberapa UDP port yang telah dikenal secara luas.

Tabel 11 - 1 Tabel Port UDP

Nomor Port UDP	Aplikasi
53	Domain Name System (DNS) Name Query
67	*BOOTP klien* (Dynamic Host Configuration Protocol [DHCP])
68	BOOTP server (DHCP)
69	Trivial File Transfer Protocol (TFTP)
137	NetBIOS Name Service
138	NetBIOS Datagram Service
161	Simple Network Management Protocol (SNMP)
445	Server Message Block (SMB)
520	Routing Information Protocol (RIP)
1812/1813	Remote Authentication Dial-In User Service (RADIUS)

Ketika paket dari suatu jaringan masuk pada firewall melalui kartu jaringan, pertama kali paket akan diperiksa oleh aturan rantai *PREROUTING* sebagai aksi yang dilakukan sebelum routing paket data dilakukan pada tabel *mangle*. Selanjutnya paket diperiksa oleh

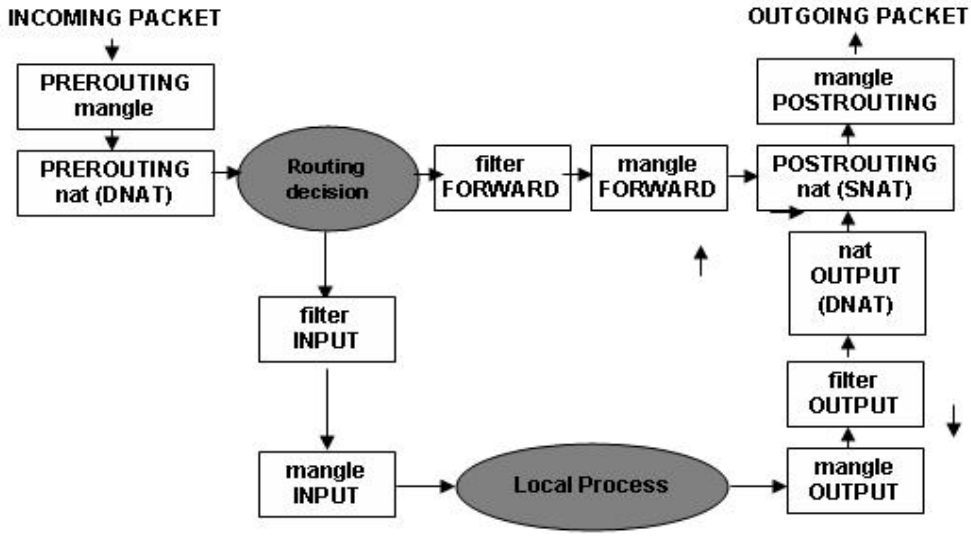
11.3.2. Mata Rantai IPTABLES

Untuk membangun sebuah firewall, yang harus kita ketahui pertama-tama adalah bagaimana sebuah paket diproses oleh firewall, apakah paket-paket yang masuk akan di buang (*DROP*) atau diterima (*ACCEPT*), atau paket tersebut akan diteruskan (*FORWARD*) ke jaringan yang lain.

Salah satu tool yang banyak digunakan untuk keperluan proses pada firewall adalah *iptables*. Program *iptables* adalah program administratif untuk *Filter Paket* dan *NAT* (*Network Address Translation).* Untuk menjalankan fungsinya, *iptables* dilengkapi dengan tabel *mangle, nat* dan *filter*.

Proses yang terjadi pada paket yang melewati suatu firewall dapat diperlihatkan pada gambar 11-11.

aturan rantai *PREROUTING* pada tabel *nat*, apakah paket akan memerlukan Tujuan yang terdapat pada aturan tujuan yang di NAT-kan (DNAT) atau tidak. Setelah itu paket mengalami routing. Di bagian ini paket tersebut akan ditentukan berdasarkan tujuan dari paket tersebut.



Gambar 11 - 11 Proses Pada Paket yang Melewati Firewall

Jika tujuan paket adalah jaringan lain, maka paket akan difilterkan oleh aturan rantai *FORWARD* pada tabel *filter*. Jika perlu, paket akan diperiksa oleh aturan rantai *POSTROUTING* pada tabel *nat*, apakah paket berasal dari sumber yang mempunyai aturan NAT, yang dalam istilah firewall dikenal dengan istilah SNAT (source NAT).

Jika tujuan paket adalah firewall, maka paket akan difilter oleh aturan rantai *INPUT* pada tabel *filter*. Selanjutnya paket akan mengalami proses lokal, paket tersebut akan di teruskan ke tabel *INPUT* untuk di proses firewall. Bila paket tersebut bertujuan untuk ke komputer lain yang berbeda jaringan, paket tersebut akan di teruskan ke kolom *FORWARD*.

Proses lokal yang terjadi pada firewall dapat berupa pengiriman paket kembali. Paket ini akan diperiksa oleh aturan rantai *OUTPUT* pada tabel *MANGLE*. Selanjutnya paket diperiksa oleh aturan rantai *OUTPUT* pada tabel *NAT*, apakah memerlukan *DNAT*. Sebelum routing, paket akan difilter oleh aturan rantai *OUTPUT* pada tabel filter.

Setelah paket tersebut memasuki kolomnya (*INPUT* atau *FORWARD*) maka paket tersebut akan dicocokkan dengan aturan-aturan yang ada pada kolom tersebut. Paket diperiksa kecocokannya dengan aturan-aturan yang ada. Beberapa aturan yang ada pada urutan firewall akan dibaca oleh sistem secara berurut dari nomor teratas berdasarkan prioritas.

Tabel 11 - 2 Tabel Filter Pada IPTABLES

No	INPUT	OUTPUT	FORWARD
1	Aturan no 1	Aturan no 1	Aturan no 1
2	Aturan no 2	Aturan no 2	Aturan no 2
3	Aturan no 3	Aturan no 3	Aturan no 3
N	Aturan n	Aturan n	Aturan n
POLICY	ACCEPT/ DROP	ACCEPT/ DROP	ACCEPT/ DROP

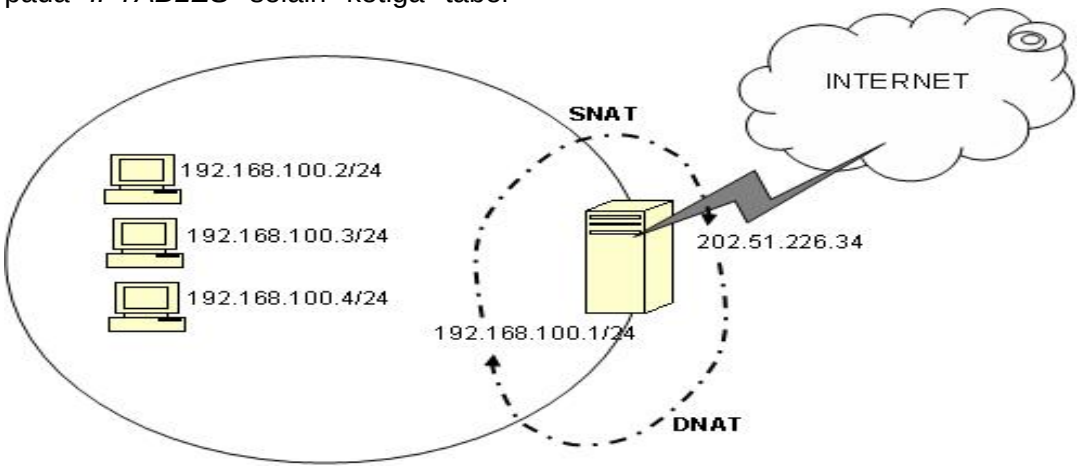
Sebagai contoh, ada paket dengan tujuan alamat IP komputer kita. Paket tersebut akan masuk ke tabel *INPUT*, kemudian paket tersebut akan di cocokkan dengan aturan no 1. Jika aturan tersebut tidak cocok dengan paket yang datang maka paket tersebut akan di cocokkan dengan aturan ke dua. Bila paket tidak

cocok maka paket akan diteruskan ke aturan paket nomor 3. Jika sistem telah mencocokkan dengan aturan yang terakhir (aturan nomor *n*) tetapi tetap tidak ada kecocok\kan juga maka *POLICY* pada tabel yang akan berlaku, yaitu apakah paket tersebut akan di terima (*ACCEPT*) atau paket tersebut akan di buang (*DROP*).

Salah satu kelebihan *IPTABLES* adalah untuk membuat komputer kita menjadi sebuah gateway menuju internet. Untuk keperluan tersebut, kita akan membutuhkan tabel lain pada *IPTABLES* selain ketiga tabel diatas. Tabel tersebut adalah tabel *NAT* (*Network Address Translation*).

Tabel 11 - 3 NAT pada IPTABLES

No	Post Routing (SNAT)	Pre Routing (DNAT)	OUTPUT
1	Aturan no 1	Aturan no 1	Aturan no 1
2	Aturan no 2	Aturan no 2	Aturan no 2
3	Aturan no 3	Aturan no 3	Aturan no 3
N	Aturan n	Aturan n	Aturan n
POLICY	**ACCEPT/ DROP**	**ACCEPT/ DROP**	**ACCEPT/ DROP**



Gambar 11 - 12 SNAT dan DNAT

SNAT digunakan untuk mengubah alamat IP pengirim (*source IP address*). Biasanya *SNAT* berguna untuk menjadikan komputer sebagai gateway menuju ke internet.

Misalnya komputer kita menggunakan alamat IP 192.168.0.1. IP tersebut adalah IP lokal. *SNAT* akan mengubah IP lokal tersebut menjadi IP publik, misalnya 202.51.226.35. begitu juga sebaliknya, bila komputer lokal kita bisa di akses dari internet maka *DNAT* yang akan digunakan.

Mangle pada *IPTABLES* banyak digunakan untuk menandai (*marking*) paket-paket untuk di gunakan di proses-proses selanjutnya. *Mangle* paling banyak di gunakan untuk *bandwidth limiting* atau pengaturan bandwidth.

Tabel 11 - 4 Tabel Mangle

No	PRE ROUTING	INPUT	FORWARD	OUTPUT	POST ROUTING
1	Aturan no 1	Aturan no 1	Aturan no 1	Aturan no 1	Aturan no 1
2	Aturan no 2	Aturan no 2	Aturan no 2	Aturan no 2	Aturan no 2
3	Aturan no 3	Aturan no 3	Aturan no 3	Aturan no 3	Aturan no 3
N	Aturan n	Aturan n	Aturan n	Aturan n	Aturan n
POLICY	**ACCEPT/ DROP**	**ACCEPT/ DROP**	**ACCEPT/ DROP**	**ACCEPT/ DROP**	**ACCEPT/ DROP**

Fitur lain dari *mangle* adalah kemampuan untuk mengubah nilai Time to Live (TTL) pada paket dan TOS (*type of service*).

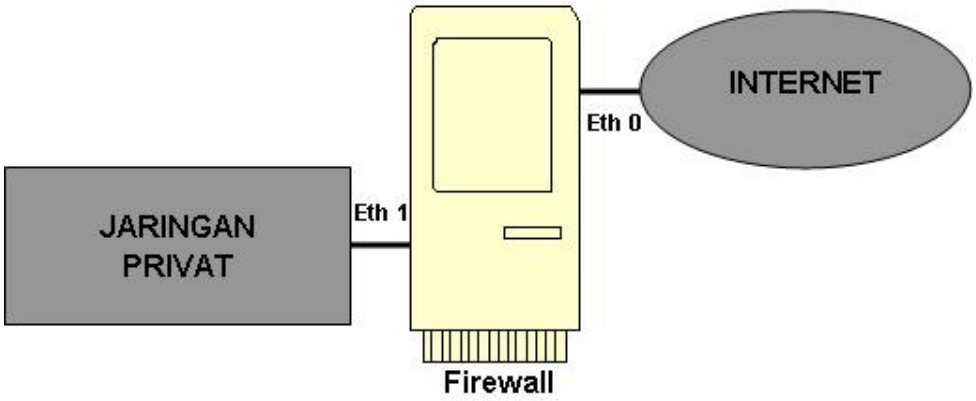
11.4 MENDESAIN SISTEM KEAMANAN JARINGAN

Berikut ini adalah langkah-langkah yang diperlukan dalam membangun sebuah firewall:

1. Menentukan topologi jaringan yang akan digunakan. Topologi dan kofigurasi jaringan akan menentukan bagaimana firewall akan dibangun.
2. Menentukan kebijakan atau *policy*. Kebijakan yang perlu di atur di sini adalah penentuan aturan-aturan yang akan diberlakukan.
3. Menentukan aplikasi– aplikasi atau servis-servis apa saja yang akan berjalan. Aplikasi dan servis yang akan berjalan harus kita ketahui agar kita dapat menentukan aturan-aturan yang lebih spesifik pada firewall kita.
4. Menentukan pengguna-pengguna mana saja yang akan dikenakan oleh satu atau lebih aturan firewall.
5. Menerapkan kebijakan, aturan, dan prosedur dalam implementasi firewall.
6. Sosialisasi kebijakan, aturan, dan prosedur yang sudah diterapkan. Batasi sosialisasi hanya kepada personil teknis yang diperlukan saja.

Dengan melakukan sosialisasi kepada pengguna-pengguna yang di kenai aturan-aturan firewall kita, di harapkan tidak terjadi kesalah-pahaman terhadap peraturan-peraturan yang diberlakukan.

Berikut ini diberikan contoh penerapan *iptables* pada firewall. Konfigurasi network yang digunakan untuk contoh diilustrasikan pada gambar 11-13.



Gambar 11 - 13 Skema Firewall dalam Jaringan

Pada gambar di atas terdapat suatu firewall yang mempunyai dua antar muka. Firewall berhubungan dengan jaringan internet melalui antar muka *eth0* dan berhubungan dengan jaringan privat melalui antar muka *eth1*. Kadang-kadang firewall berhubungan dengan jaringan internet menggunakan modem, dalam hal ini antarmuka *eth0* dapat diganti dengan *ppp0*.

Kemampuan pertama yang harus di miliki firewall adalah melakukan *forward IP Address* dari antarmuka *eth0* menuju antarmuka *eth1* dan sebaliknya dari antarmuka *eth1* menuju antarmuka *eth0*. Caranya adalah dengan memberi nilai 1 pada parameter *ip_forward* dengan perintah

```
# echo ”1”
>/proc/sys/net/ipv4/ip_forward
```

Dalam beberapa variant Linux dilakukan dengan memberi baris konfigurasi pada file */etc/sysconfig/network*.

```
FORWARD_IPV4=yes
```

11.4.1. MEMBUAT INISIALISASI

Inisialisasi aturan *iptables* digunakan untuk membuat kebijakan umum terhadap rantai *iptables* yang akan di terapkan pada firewall. Kebijakan ini akan di terapkan jika tidak ada aturan yang sesuai. Kebijakan umum yang diterapkan dalam suatu firewall umumnya adalah sebagai berikut:

- Kebijakan untuk membuang semua paket yang menuju, melintas dan keluar dari firewall. Kebijakan ini akan di terapkan pada paket apabila tidak ada satupun aturan yang sesuai dengan paket tersebut. Kebijakan ini di terapkan dengan memberikan status *DROP* untuk semua rantai pada tabel filter.

```
# iptables –p input DROP
# iptables –p forward DROP
# iptables –p output DROP
```

- Kebijakan untuk menerima semua paket yang menuju dan meninggalkan perangkat *loopback*. Kebijakan ini di terapkan dengan memberikan status *ACCEPT* pada

semua paket yang masuk dan keluar perangkat loopback.

```
# iptables – A INPUT – i lo – j ACCEPT
# iptables – A OUTPUT– o lo – j ACCEPT
```

- Kebijakan menerima semua paket sebelum mengalami routing. Kebijakan ini diterapkan dengan memberikan status *ACCEPT* untuk rantai *POSTROUTING* dan *PREROUTING* pada tabel *NAT*.

```
# iptables – t nat – p POSTROUTING – j ACCEPT
# iptables – t nat – p PREROUTING – j ACCEPT
```

Tentu saja kebijakan umum yang di terapkan untuk suatu sistem sangat tergantung pada pengelolaan jaringan. Kebijakan tersebut tidak harus seperti di atas, tapi dapat disesuaikan dengan keperluan.

11.4.2. MENGIJINKAN LALU-LINTAS PAKET ICMP

Paket ICMP biasanya digunakan untuk menguji apakah suatu peralatan jaringan sudah terhubung secara benar dalam jaringan. Biasanya untuk menguji apakah suatu peralatan sudah terhubung secara benar dalam jaringan dapat dilakukan dengan perintah *ping*. Perintah ini akan mencoba mengirim paket ICMP ke alamat IP tujuan dan menggunakan tanggapan dari alamat IP tersebut. Untuk memberikan keleluasaan keluar, masuk dan melintasnya paket ICMP diterapkan dengan aturan tersebut.

```
# iptables – A INPUT –p icmp -j ACCEPT
# iptables – A FORWARD –p icmp -j ACCEPT
# iptables – A OUPUT –p icmp -j ACCEPT
```

Maksud perintah di atas adalah sebagai berikut:

- Firewall mengijinkan paket ICMP yang akan masuk.
- Firewall mengijinkan paket ICMP yang akan melintas.
- Firewall mengijinkan paket ICMP yang akan keluar.

Perintah ketiga ini memungkinkan firewall untuk mananggapi paket ICMP yang dikirim ke firewall. Jika perintah ketiga tidak diberikan, maka firewall tidak dapat mengirim keluar tanggapan paket ICMP.

Catatan: Kadang-kadang paket ICMP digunakan untuk tujuan yang tidak benar, sehingga kadang-kadang firewall ditutup untuk menerima lalu lintas paket tersebut. Jika firewall tidak diijinkan untuk menerima lalu lintas paket

ICMP, maka perintah diatas tidak perlu dicantumkan.

11.4.3. Mengijinkan Paket SSH Masuk Firewall

Untuk mengkonfigurasi komputer dalam jaringan, biasanya dilakukan secara jarak jauh. Artinya pengelolaan tidak harus datang dengan berhadapan dengan komputer tersebut. Termasuk dalam hal ini untuk pengelolaan firewall. Untuk mengelola firewall dari jarak jauh, dapat digunakan program *SSH*.

Program *SSH* menggunakan paket TCP dengan port 22 untuk menghubungkan antara dua komputer. Oleh sebab itu firewall harus mengijinkan paket dengan tujuan port 22 untuk masuk ke firewall. Firewall juga harus mengijinkan paket yang berasal dari port 22 untuk keluar dari firewall. Berikut ini perintah yang diterapkan untuk mengijinkan akses *SSH* melalui antarmuka *eth1* yaitu dari jaringan privat.

```
# iptables – A INPUT –p tcp –dport 22 –i eth1 -j ACCEPT
# iptables – A OUTPUT –p tcp –sport 22 –o eth1 -j ACCEPT
```

Maksud dari perintah di atas adalah sebagai berikut:

- Firewall mengijinkan masuk untuk paket TCP yang punya tujuan port 22 melalui antarmuka *eth1*
- Firewall mengijinkan keluar untuk paket TCP yang berasal dari port 22 melalui antarmuka *eth1*

Aturan tersebut memungkinkan akses *SSH* hanya dari jaringan privat melalui antarmuka *eth1*. Untuk alasan keamanan, akses *SSH* dari jaringan privat dapat dibatasi untuk akses yang hanya berasal dari alamat jaringan tertentu atau bahkan dari komputer tertentu. Hal ini dilakukan dengan menambah opsi *–s* diikuti alamat jaringan atau alamat IP pada perintah pertama, contohnya diijinkan dari sumber yang mempunyai alamat IP hanya 192.168.0.1

```
# iptables – A OUTPUT –s 192.168.0.1. –p tcp –sport 22 –o eth1 -j ACCEPT
```

11.4.4. . Mengijinkan Akses HTTP Melintas Firewall

Akses *http* merupakan protokol yang paling banyak digunakan untuk berselancar di internet. Informasi yang disajikan pada internet umumnya menggunakan akses *http* ini. Akses *http* menggunakan port 80 dengan jenis paket TCP.

Firewall biasanya mengijinkan akses *http* terutama yang melintas firewall baik yang keluar atau masuk jaringan privat. Akses *http* yang keluar jaringan privat digunakan untuk memberi akses *http* bagi komputer yang berada di jaringan privat. Sedangkan akses *http* dari internet terjadi apabila pada jaringan privat terdapat server web yang dapat diakses dari jaringan internet.

Penerapan aturan *iptables* untuk mengijinkan akses *http* adalah sbb:

```
# iptables – A FORWARD –p tcp –dport 80 –i eth1 -j ACCEPT
# iptables – A FORWARD –p tcp –sport 80 –o eth1 -j ACCEPT
# iptables – A FORWARD –p tcp –dport 80 –i eth0 -j ACCEPT
# iptables – A FORWARD –p tcp –sport 80 –o eth0 -j ACCEPT
```

Maksud dari perintah di atas adalah sebagai berikut:

- Firewall mengijinkan melintas untuk paket TCP yang punya tujuan port 80 melalui antarmuka *eth1*
- Firewall mengijinkan melintas untuk paket TCP yang punya asal port 80 melalui antarmuka *eth1*
- Firewall mengijinkan melintas untuk paket TCP yang punya tujuan port 80 melalui antarmuka *eth0*
- Firewall mengijinkan melintas untuk paket TCP yang punya asal port 80 melalui antarmuka *eth0*.

Perintah pertama dan kedua digunakan untuk mengijinkan akses *http* yang berasal dari jaringan privat, sedangkan perintah ketiga dan keempat digunakan untuk mengijinkan akses *http* yang berasal dari internet.

Keempat perintah tersebut dapat diganti dengan satu perintah menggunakan opsi *multiport* sebagai berikut:

```
# iptables – A FORWARD –p tcp –m multiport --port 80 -j ACCEPT
```

Perintah tersebut menyatakan bahwa firewall mengijinkan paket TCP yang punya port 80 (tujuan / asal) untuk melintas (dari *eth0* atau *eth1*).

11.4.5. Mengijinkan QUERY Server DNS

Firewall biasanya mempunyai minimal satu alamat IP untuk server DNS. Untuk query server DNS digunakan paket UDP melalui port 53. Firewall memerlukan query server DNS untuk menentukan alamat IP yang berhubungan dengan suatu nama host. Query server DNS pada firewall ini biasanya diijinkan untuk query server DNS keluar firewall (baik via *eth0* atau *eth1*) dan query server DNS melintasi server firewall. Aturan *iptables* yang diterapkan untuk mengijinkan query sever DNS keluar dari firewall adalah sebagai berikut:

```
# iptables – A OUTPUT –p udp –dport 53 –o eth1 -j ACCEPT
# iptables – A INPUT –p udp –dport 53 –i eth1 -j ACCEPT
# iptables – A OUTPUT –p udp –dport 53 –o eth0 -j ACCEPT
# iptables – A INPUT –p udp –dport 53 –i eth0 -j ACCEPT
```

Maksudnya:
- Firewall mengijinkan keluar untuk paket UDP yang punya tujuan port 53 melalui antarmuka *eth1*.
- Firewall mengijinkan keluar untuk paket UDP yang punya asal port 53 melalui antarmuka *eth1*
- Firewall mengijinkan keluar untuk paket UDP yang punya tujuan port 53 melalui antarmuka *eth0*.
- Firewall mengijinkan keluar untuk paket UDP yang punya asal port 53 melalui antarmuka *eth0*

Perintah pertama dan kedua digunakan untuk query server DNS keluar melalui antarmuka *eth1*, sedangkan perintah ketiga dan keempat digunakan untuk mengijinkan query server DNS keluar melalui antarmuka *eth0*.

Selanjutnya firewall akan mengijinkan query server DNS untuk melintas. Aturan *iptables* untuk mengijinkan query server DNS melintasi firewall adalah sebagai berikut:

```
# iptables – A FORWARD –p udp –m multiport –ports 53 -j ACCEPT
```

Perintah tersebut menyatakan bahwa firewall mengijinkan paket UDP yang punya port 53 untuk melintas.

11.5 IP Masquerade

Alamat IP yang digunakan untuk menyusun jaringan lokal umumnya menggunakan alamat IP privat. Alamat IP ini tidak diroutingkan oleh jaringan publik, sehingga komputer yang ada pada jaringan lokal tidak dapat langsung berhubungan dengan internet.

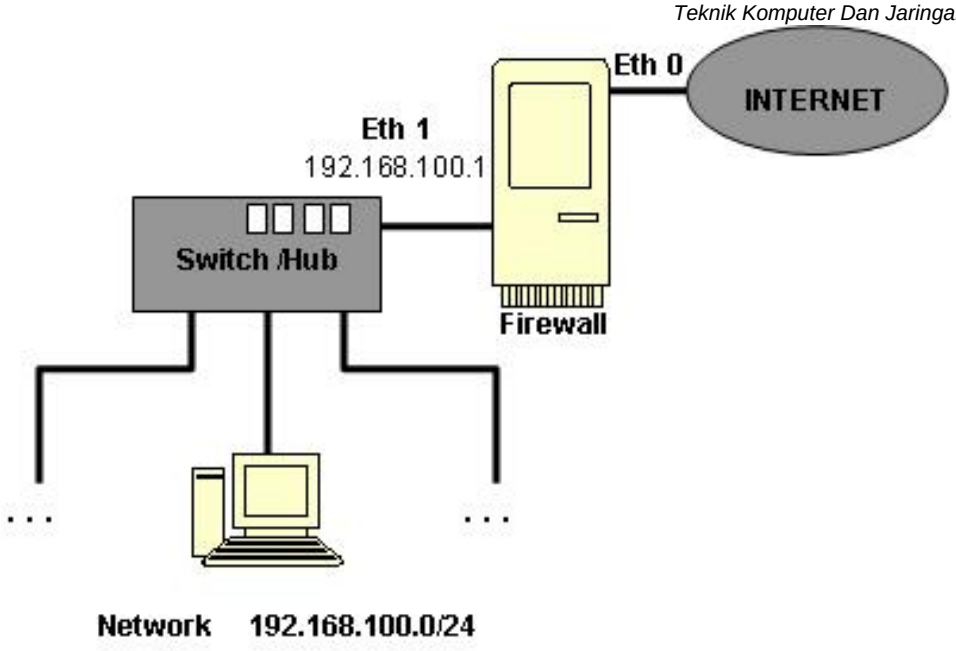
Hubungan antara komputer pada jaringan lokal dengan jaringan publik dilakukan dengan cara menyamarkan alamat IP privat dengan alamat IP yang dipunyai oleh kartu jaringan dengan alamat IP publik. Proses penyamaran alamat IP privat menjadi alamat IP publik ini disebut dengan *IP MASQUERADE*.

Dengan cara yang diterapkan oleh konsep *IP MASQUERADE*, semua komputer pada jaringan lokal ketika berhubungan dengan jaringan publik seperti mempunyai alamat IP kartu jaringan yang punya alamat IP publik.

IP MASQUERADE adalah salah satu bentuk translasi alamat jaringan (*NAT*), yang memungkinkan bagi komputer-komputer yang terhubung dalam jaringan lokal yang menggunakan alamat IP privat untu berkomunikasi ke internet melalui firewall.

Teknik *IP MASQUERADE* adalah cara yang biasanya digunakan untuk menghubungkan jaringan lokal dengan publik (internet). Bagi pelanggan internet yang hanya diberi satu alamat IP dinamis (*dial up*) menggunakan modem.

Berikut ini diberikan contoh penerapan *IP MASQUERADE* (*NAT*).



Gambar 11 - 14 Jaringan untuk Penerapan IP MASQUERADE

Pada gambar 11-14, jaringan privat dengan alamat IP 192.168.100.0/24 berhubungan dengan internet melalui firewall. Pada komputer firewall terdapat dua antarmuka (*eth0* dan *eth1*). Komputer firewall berhubungan dengan jaringan privat melalui *eth1* yang diberi alamat IP 192.168.100.254. sedangkan dengan jaringan internet berhubungan melalui *eth0* dengan alamat IP publik.

Syarat utama supaya dapat menjalankan fungsi *IP MASQUERADE*, komputer firewall harus memiliki kebijakan untuk meneruskan paket yang akan dikirim melalui *eth0* maupun paket yang diterima melalui *eth1*. Jenis paket dan nomor port yang akan diteruskan diatur melalui chains tertentu.

Selanjutnya paket yang akan dikirim melalui antarmuka *eth0* harus menjalani translasi alamat IP dengan proses *IP MASQUERADE* dengan perintah:

```
# iptables – t nat –A POSTROUTING –o eth0 –s 192.168.100.0/24 –j MASQUERADE
```

Perintah tersebut menyatakan bahwa setelah mengalami routing, paket yang akan dikirim melalui antarmuka *eth0* yang berasal dari jaringan 192.168.100.0/24 akan mengalami proses *IP MASQUERADE*.

Jika firewall berhubungan dengan internet melalui suatu modem, maka antarmuka untuk berhubungan dengan internet adalah *ppp0*, sedangkan antarmuka untuk berhubungan dengan jaringan privat adalah *eth0*, dengan demikian harus diberikan perintah:

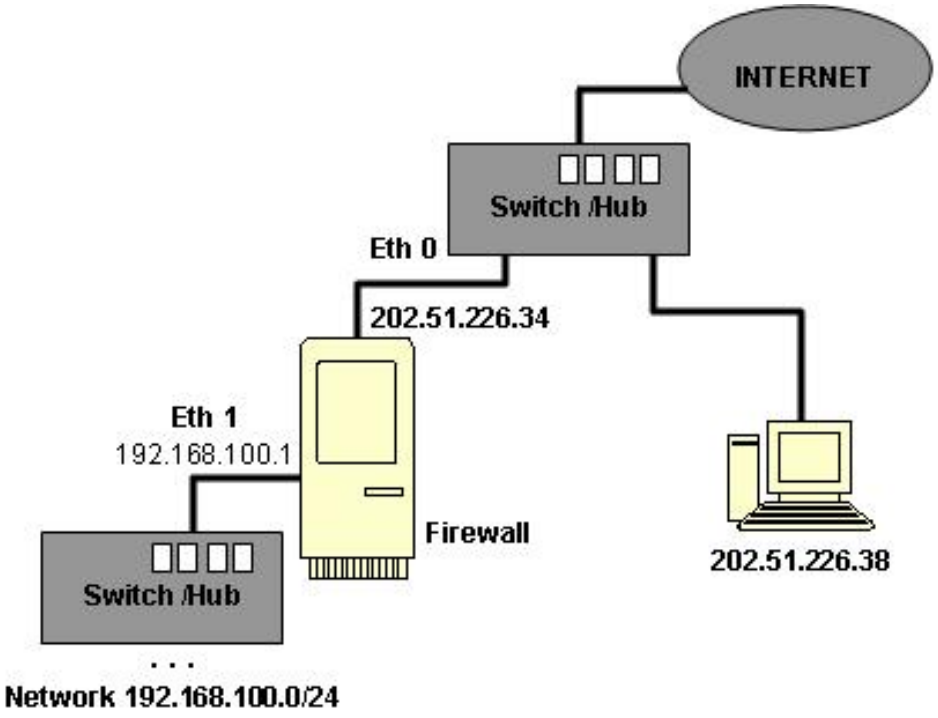
```
# iptables – t nat –A POSTROUTING      –o ppp0 –s 192.168.100.0/24 –j MASQUERADE
```

IP MASQUERADE pada hubungan dial up dengan modem dapat juga diterapkan pada pelanggan rumah yang ingin membagi hubungan internet pada beberapa komputer.

Translasi alamat IP secara statis dapat dilakukan dengan penerapan konsep *subnetting* pada pengalamatan jaringan privat. Begitu juga untuk penerapan alamat IP publik yang diberikan oleh Internet Service Provider (biasanya terbatas hanya dua alamat IP), maka akses dari jaringan lokal dapat dilakukan dengan beberapa cara. Dua contoh yang dapat dilakukan adalah teknik hubungan langsung dan DMZ (*De-Militarize Zone)*.

11.6 Teknik Hubungan Langsung

Pada teknik hubungan langsung, komputer-komputer yang dirancang dapat untuk diakses melalui jaringan internet, diberi alamat IP publik dan langsung dihubungkan pada internet, tanpa melalui firewall. Sehingga komputer tersebut akan dirouting oleh jaringan publik. Contoh struktur nya:



Gambar 11 - 15 Jaringan Hubungan Langsung

Pada struktur diatas, komputer-komputer yang mempunyai alamat IP publik dihubungkan langsung dengan internet. Komputer dengan alamat IP 202.51.226.35 tidak diletakkan dibawah firewall, sehingga tidak diperlukan translasi alamat IP. Yang diletakkan di bawah firewall hanya

komputer dengan alamat IP privat 192.168.100.0/24. jaringan privat inilah yang memerlukan translasi alamat jaringan ketika berhubungan dengan jaringan publik. Jaringan privat ini dapat dihubungkan ke internet dengan menggunakan teknik *IP Masquerade*.

Karena alamat IP untuk *eth0* diketahui secara pasti, dapat juga digunakan opsi *–to-source* untuk menentukan asal alamat IP pada alamat publikj. Dengan perintah pada firewall sebagai berikut:

```
# iptables – t nat –A POSTROUTING –o eth0 –s 192.168.100.0/24 –j snat –to-source 202.51.226.34
```

Perintah ini menyatakan bahwa setelah mengalami routing, paket yang akan dikirim melalui antarmuka eth0 yang berasal dari jaringan 192.168.100.0/24 akan mengalami SNAT menjadi alamat IP 202.51.226.34.

11.7 DMZ (*DE-MILITARIZED ZONE)*.

Pada teknik ini, baik komputer yang dirancang untuk dapat diakses dari internet maupun yang tidak dapat diakses dari internet semuanya diberi alamat IP privat dan diletakkan dibawah firewall. Alamat IP komputer yang dirancang dapat diakses dari internet dipetakan ke alamat IP publik yang diberikan pada firewall. Pemetaan yang terjadi adalah dari satu ke satu.

Ada dua teknik DMZ yang dapat digunakan. Yang pertama adalah meletakkan komputer DMZ pada jaringan yang terpisah dari jaringan privat. Yang kedua adalah meletakkan komputer DMZ pada jaringan yang sama dengan jaringan prvat.

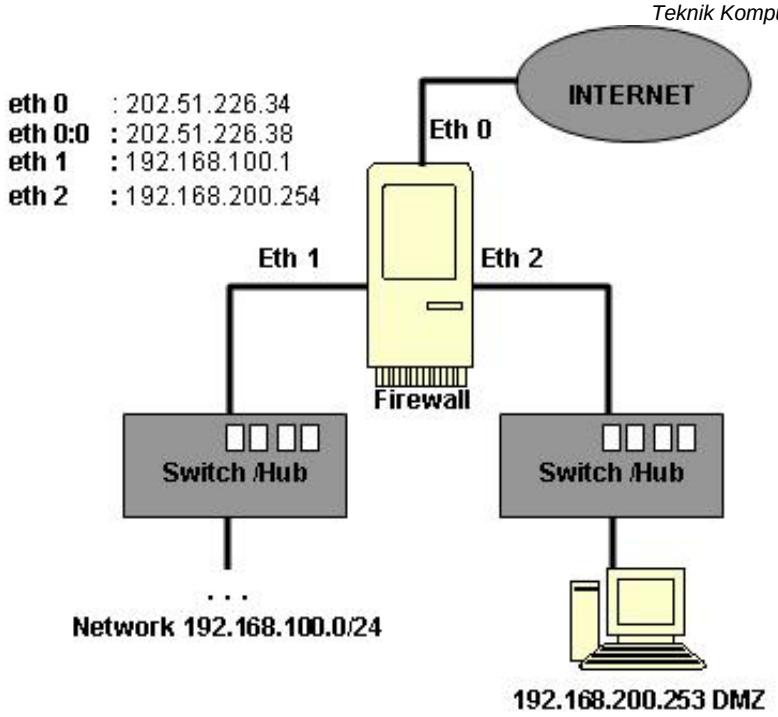
11.7.1 DMZ Pada Jaringan Terpisah

Pada teknik ini, untuk komputer yang berada pada DMZ dibuatkan jaringan tersendiri yang terpisah dari jaringan privat lain. Komputer pada DMZ tetap menggunakan alamat IP privat. Dalam hal ini firewall memerlukan tiga kartu jaringan, yaitu:

- *eth0* berhubungan dengan internet
- *eth1* berhubungan dengan jaringan privat.
- *eth2* berhubungan dengan DMZ.

Topologinya dapat digambar pada gambar 11-16.

```
# iptables – t nat –A POSTROUTING –o eth0 –s 192.168.100.0/24 –j snat –to-source 202.51.226.34
```



Gambar 11 - 16 Jaringan DMZ Terpisah

Pada topologi diatas terdapat suatu firewall dengan tiga antarmuka, yaitu *eth0*, *eth1* dan *eth2*. Kartu *eth0* diberi dua alamat IP publik menggunakan teknik ip alias, yaitu 202.51.226.34 dan 202.51.226.38.

Alamat IP 202.51.226.34 digunakan untuk memetakan alamat IP seluruh komputer pada jaringan 192.168.0.100/24, sehingga terjadi pemetaan banyak ke satu.

Alamat IP 202.51.226.38 digunakan untuk memetakan satu komputer yang memiliki alamat 192.168.200.253, sehingga terjadi pemetaan satu ke satu.

Untuk keperluan translasi alamat jaringan 192.168.0.100/24 dapat digunakan teknik yang sudah dibahas pada bagian sebelumnya.

```
# iptables – t nat –A POSTROUTING –o eth0 –s 192.168.100.0/24 –j snat –to-source 202.51.226.34
```

Sedangkan untuk translasi alamat jaringan bagi komputer dengan alamat 192.168.200.253 dapat menggunakan pasangan perintah sebagai berikut:

```
# iptables – t nat –A POSTROUTING –i eth0 –d 202.51.226.38 –j DNAT --to-destination 192.168.200.253.
# iptables – t nat –A POSTROUTING –o eth0 –s 192.168.200.253.–j SNAT --to-source 202.51.226.38.
```

Maksudnya:

- Perintah pertama menyatakan bahwa sebelum routing, paket yang masuk melalui antarmuka *eth0* dengan tujuan 202.51.226.38 akan mengalami proses *DNAT* menjadi alamat IP tujuan 192.168.200.253.
- Perintah kedua menyatakan bahwa setelah routing, paket yang akan dikirim melalui antarmuka *eth0* yang berasal dari alamat 192.168.200.253 akan mengalami proses *SNAT* menjadi alamat tujuan 202.51.226.38.

Pada teknik ini, hubungan antara alamat jaringan DMZ dengan alamat jaringan privat dilakukan secara routing.

11.7.2 DMZ Pada Satu Jaringan

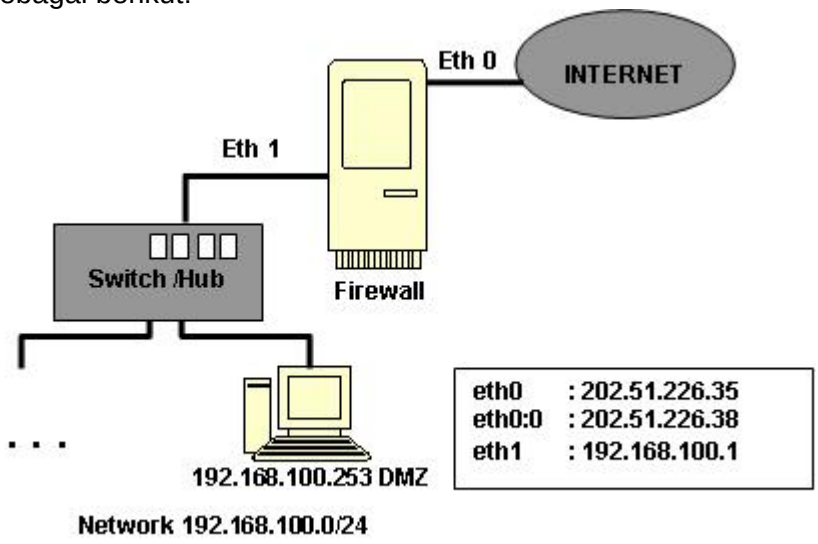
Pada teknik DMZ juga dimungkinkan untuk memasukkan komputer DMZ dengan alamat yang sama dengan alamat jaringan privat, Dalam hal ini komputer DMZ menggunakan alamat IP pada jaringan tersebut. Teknik ini akan menghemat penggunaan switch dan kartu jaringan.

Pada teknik ini komputer firewall cukup menggunakan dua antar muka *eth0* dan *eth1*.

Eth0 digunakan untuk berhubungan dengan internet, sedangkan *eth1* digunakan untuk berhubungan dengan jaringan privat.

```
# iptables – t nat –A POSTROUTING –i eth0 –d 202.51.226.38 –j DNAT --to-destination 192.168.200.253.
# iptables – t nat –A POSTROUTING –o eth0 –s 192.168.200.253.–j SNAT --to-source 202.51.226.38.
```

Struktur nya sebagai berikut:



Gambar 11 - 17 Jaringan DMZ dalam Satu Jaringan

Pada struktur diatas terdapat satu firewall yang mempunyai dua antarmuka (*eth0* dan *eth1*). Antarmuka *eth0* diberi dua alamat IP

publik menggunakan teknik IP Alias, yaitu 202.51.226.34 dan 202.51.226.38. Alamat IP 202.51.226.34 digunakan untuk memetakan alamat IP seluruh komputer pada jaringan 192.168.100.0/24, sehingga terjadi pemetaan banyak ke satu.

Alamat IP 202.51.226.38 digunakan untuk memetakan satu komputer itu yang memiliki alamat IP 192.168.100.253, sehingga terjadi pemetaan satu ke satu.

Alamat jaringan 192.168.100.0/24 dapat dianggap sebagai jaringan DMZ, tapi hanya ada satu komputer yang menggunakan pemetaan satu ke satu, sedangkan komputer yang lain menggunakan pemetaan banyak ke satu,

Untuk translasi alamat jaringan 192.168.100.0/24 dapat digunakan teknik *masquerade* sebelumnya. Sedangkan untuk translasi alamat jaringan untuk komputer dengan alamat IP 192.168.100.253, dapat menggunakan pasangan perintah sebagai berikut:

```
# iptables – t nat –A PREROUTING –i eth0 –d 202.51.226.38 .–j DNAT --to-destination 192.168.100.253
# iptables – t nat –A POSTROUTING –o eth0 –s 192.168.100.253.–j SNAT --to-source 202.51.226.38
```

Maksudnya:

- Perintah pertama menyatakan bahwa sebelum routing, paket yang masuk melalui antarmuka *eth0* dengan tujuan 202.51.226.38 akan mengalami *DNAT* menjadi alamat tujuan 192.168.100.253
- Perintah kedua menyatakan bahwa setelah routing, paket yang akan dikirim melalui antarmuka *eth0* yang berasal dari alamat IP 192.168.100.253 akan mengalami proses *SNAT* menjadi alamat asal 202.51.226.38

Perlu dicatat bahwa komputer yang dirancang untuk berhubungan dengan internet dengan teknik DMZ tidak terbatas pada satu komputer.

11.7.3 Firewall Dengan Hardware Khusus

Fungsi firewall seperti disebutkan diatas dapat juga dilakukan dengan menggunakan hardware khusus dari vendor yang telah didesain untuk keperluan pembuatan *chains* tertentu. Walaupun demikian, teknik dan penerapannya sama saja dengan menggunakan *IP Tables*.

Pada hardware khusus Firewall penerapan chains-nya didesain sedemikian, agar memudahkan administrator dalam mengimplementasikan rule/*policy* firewall. Satu hal yang membedakan adalah perangkat firewall dari vendor hanya didesain khusus untuk keperluan *chains* tanpa fungsi lain, sementara PC Firewall dapat digunakan selain untuk Firewall juga untuk fungsi terminal jaringan yang lain.

11.8 Soal-Soal Latihan

Soal-soal latihan ini diperuntukkan bagi siswa yang telah selesai melakukan pemahaman Bab 11 mengenai Keamanan Komputer.

Jawablah pertanyaan dibawah ini dengan tepat.

1. Apa yang dimaksud dengan Firewall?
2. Jelaskan jenis-jenis firewall untuk jaringan komputer.
3. Gambarkan hubungan kerja Firewall dengan susunan lapisan Model Referensi TCP/IP.
4. dari keempat jenis firewall, manakah yang mudah diimpelementasi tetapi mempunyai kehandalan yang tinggi?
5. Jelaskan perbedaan antara Prerouting dan Postrouting.
6. Bagaimana menerapkan suatu rule/*policy* untuk memperbolehkan akses *http* pada suatu server?
7. Apa yang dimaksud dengan DMZ?
8. Bagaimana cara untuk mengimplementasikan *NAT* untuk IP Private 192.168.0.0/24 dengan Publik IP 202.203.204.2/30
9. Gambarkan topologi untuk nomor 8.
10. Apa yang dimaksud dengan Firewall dengan hardware khusus