

Penggunaan *Firewall* Untuk Menjaga Keamanan Sistem Jaringan Komputer

Agus Aan Jiwa P.

studywithaan@gmail.com

<http://agus-aan.web.ugm.ac.id>

Lisensi Dokumen:

Copyright © 2009 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Abstrak :

Dalam sebuah jaringan, istilah “*firewall*” tentunya terdengar sudah tidak asing lagi. Karena saat ini *firewall* sudah banyak digunakan, terutama dalam sebuah jaringan komputer yang terkoneksi langsung ke jaringan publik atau yang dikenal dengan *internet*. Dengan pesatnya perkembangan *internet*, dapat memberikan dampak positif bagi kita sebagai penyedia layanan informasi dan komunikasi, selain itu internet juga dapat memberikan dampak negatif sekaligus ancaman bagi penggunaannya. Sehingga akses jaringan kita dengan internet harus dibatasi oleh sebuah pembatas yang dikenal dengan *firewall*.

Kata Kunci : *firewall*, jaringan komputer.

Pendahuluan

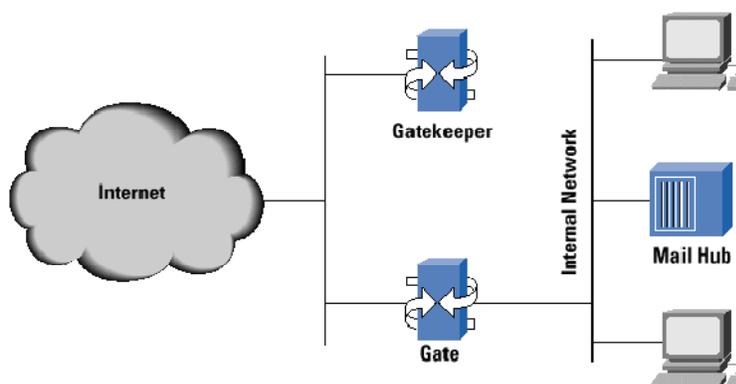
Saat ini internet sudah semakin banyak diakses oleh banyak orang. Penggunaan internet nampaknya sudah semakin tidak dapat dipisahkan di berbagai bidang dalam kehidupan manusia di dunia ini. Dengan adanya internet, seseorang dapat dengan mudah mengetahui dan mendapatkan informasi, mudah berkomunikasi dengan rekan tanpa memandang jarak dan waktu, mudah melakukan transaksi dimanapun dan kapanpun, mudah melakukan aktivitas belajar-mengajar jarak jauh dan masih banyak lagi kemudahan yang diberikan oleh internet. Seolah-olah dengan adanya internet kita merasakan bahwa dunia itu seperti tanpa batas. Di belahan dunia manapun saat ini sudah dapat dihubungkan dengan internet, yang menyediakan beragam informasi yang dapat diakses oleh siapapun.

Sejalan dengan pesatnya perkembangan internet, selain memberikan dampak positif sebagai penyedia layanan informasi dan komunikasi, internet juga dapat memberikan dampak negatif sekaligus ancaman bagi penggunaannya. Ancaman itu bentuknya berbagai macam dari virus, trojan, *cacker*, dan yang lainnya. Dengan akses yang tak terbatas, diibaratkan rumah yang tidak memiliki tembok yang dapat dimasuki oleh siapa saja yang berkepentingan tanpa dapat diketahui niatnya baik ataupun buruk. Dengan keadaan seperti itu, sudah seharusnya kita memberikan perlindungan terhadap rumah kita dengan mendirikan tembok baik dari beton atau kayu, sehingga akses ke rumah lebih mudah dikontrol. Sama halnya dengan komputer yang terhubung dengan internet, juga harus diberikan tembok pelindung yang sering disebut dengan “*firewall*” untuk melindungi komputer dari ancaman yang datang dari internet.

Mengenal Sejarah *Firewall*

Arman (2007) menyatakan bahwa, *network firewall* yang pertama muncul pada akhir era 1980-an yaitu berupa perangkat router yang dipakai untuk memisahkan suatu network menjadi jaringan lokal (LAN) yang lebih kecil, dimana kondisi ini penggunaan *firewall* hanya dimaksudkan untuk mengurangi masalah peluberan (*spill over*) data dari LAN ke seluruh jaringan untuk mencegah masalah-masalah semacam *error* pada manajemen jaringan, atau aplikasi yang terlalu banyak menggunakan sumber daya meluber ke seluruh jaringan. Penggunaan *firewall* untuk keperluan sekuriti (*security firewall*) pertama kali digunakan pada awal dekade 1990-an, berupa router IP dengan aturan *filter* tertentu. Aturan sekuriti saat itu berupa sesuatu seperti: ijinan setiap orang “di sini” untuk mengakses “ke luar sana”, juga cegahlah setiap orang (atau apa saja yang tidak disukai) “di luar sana” untuk masuk “ke sini”. *Firewall* semacam ini cukup efektif, tetapi memiliki kemampuan yang terbatas. Seringkali sangat sulit untuk menggunakan aturan filter secara benar. Sebagai contoh, dalam beberapa kasus terjadi kesulitan dalam mengenali seluruh bagian dari suatu aplikasi yang dikenakan restriksi. Dalam kasus lainnya, aturan filter harus dirubah apabila ada perubahan “di luar sana”.

Firewall generasi selanjutnya lebih fleksibel, yaitu berupa sebuah *firewall* yang dibangun pada apa yang disebut “Bastion Host”. *Firewall* komersial yang pertama dari tipe ini, yang menggunakan *filter* dan *gateway* aplikasi (*proxies*), kemungkinan adalah produk dari *Digital Equipment Corp* (DEC). DEC yang dibangun berdasarkan *firewall* korporat DEC. Brian Reid dan tim engineering di laboratorium sistem jaringan DEC di Palo Alto adalah pencipta *firewall* DEC. *Firewall* komersial pertama dikonfigurasi untuk, dan dikirimkan kepada pelanggan pertamanya, sebuah perusahaan kimia besar yang berbasis di pantai timur AS pada 13 Juni 1991. Dalam beberapa bulan kemudian, Marcus Ranum dari Digital Corp. menciptakan security proxies dan menulis ulang sebagian besar kode program *firewall*. Produk *firewall* tersebut kemudian diproduksi massal dengan nama dagang DEC SEAL (singkatan dari *Security External Access Link*). DEC SEAL tersusun atas sebuah sistem eksternal yang disebut *gatekeeper* sebagai satu-satunya sistem yang dapat berhubungan dengan internet, sebuah *filtering gateway* yang disebut *gate*, dan sebuah mailhub internal (gambar 1).



Gambar 1 DEC SEAL *Firewall* komersial pertama (Arman 2007)

“Bastion Host” adalah sistem/bagian yang dianggap tempat terkuat dalam sistem keamanan jaringan oleh administrator. atau dapat disebut bagian terdepan yang dianggap paling kuat dalam menahan serangan, sehingga menjadi bagian terpenting dalam pengamanan jaringan, biasanya merupakan komponen *firewall* atau bagian terluar sistem publik. Umumnya Bastion host akan menggunakan Sistem operasi yang dapat menangani semua kebutuhan misal : *Unix, linux, NT* (Muammar W. K, 2004). *Firewall* untuk pertama kalinya dilakukan dengan menggunakan prinsip “*non-routing*” pada sebuah *Unix host* yang menggunakan 2 buah *network interface card, network interface card* yang pertama di hubungkan ke internet (jaringan lain) sedangkan yang lainnya di hubungkan ke PC (jaringan lokal)(dengan catatan tidak terjadi “*route*” antara kedua *network interface card* di PC ini).

Definisi Firewall

Istilah “*firewall*” sendiri sebenarnya juga dikenal dalam disiplin lain, dan dalam kenyataannya, istilah ini tidak hanya bersangkutan dengan terminologi jaringan. Kita juga menggunakan *firewall*, misalnya untuk memisahkan garasi dari rumah, atau memisahkan satu apartemen dengan apartemen lainnya. Dalam hal ini, *firewall* adalah penahan (*barrier*) terhadap api yang dimaksudkan untuk memperlambat penyebaran api seandainya terjadi kebakaran sebelum petugas pemadam kebakaran datang untuk memadamkan api. Contoh lain dari *firewall* juga bisa ditemui pada kendaraan bermotor, dimana *firewall* memisahkan antara ruang penumpang dan kompartemen mesin.

Untuk *firewall* di dalam terminologi jaringan, memiliki beberapa pengertian antara lain adalah sebagai berikut :

Firewall didefinisikan sebagai suatu cara atau mekanisme yang diterapkan baik terhadap *hardware, software* ataupun sistem itu sendiri dengan tujuan untuk melindungi, baik dengan menyaring, membatasi atau bahkan menolak suatu atau semua hubungan/kegiatan suatu segmen pada jaringan pribadi dengan jaringan luar yang bukan merupakan ruang lingkungannya (Muammar W. K, 2004).

Firewall (dari buku *Building Internet Firewalls*, oleh Chapman dan Zwicky) didefinisikan sebagai sebuah komponen atau kumpulan komponen yang membatasi akses antara sebuah jaringan yang diproteksi dan internet, atau antara kumpulan-kumpulan jaringan lainnya.

Stiawan (2008) mengatakan bahwa, *firewall* adalah sebuah komputer yang memproteksi jaringan dari jaringan yang tidak dipercaya yang memisahkan antara jaringan lokal dengan jaringan publik, dengan melakukan metode filtering paket data yang masuk dan keluar (Marcus Goncalves, *Firewall Completed:227*)

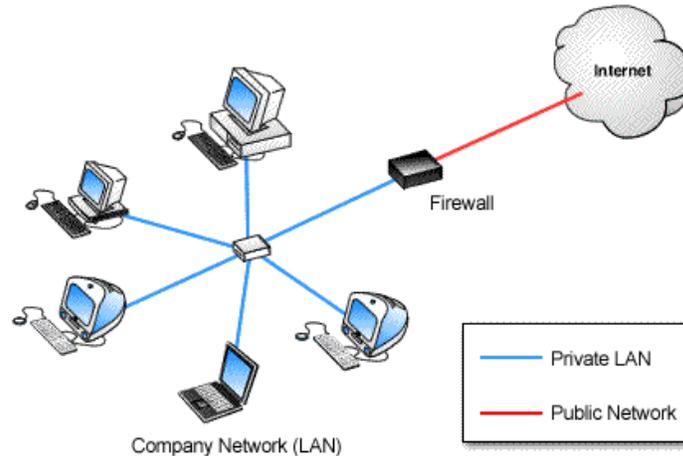
Arman (2007) mendefinisikan *firewall* sebagai sebuah titik diantara dua/lebih jaringan dimana semua lalu lintas (trafik) harus melaluinya (*chooke point*); trafik dapat dikendalikan oleh dan diautentifikasi melalui suatu perangkat, dan seluruh trafik selalu dalam kondisi tercatat (*logged*).

Dari beberapa definisi diatas, penulis dapat memberikan definisi dimana *firewall* adalah sebuah pembatas antara suatu jaringan lokal dengan jaringan lainnya yang sifatnya publik (dapat diakses oleh siapapun) sehingga setiap data yang masuk dapat diidentifikasi untuk dilakukan penyaringan sehingga aliran data dapat dikendalikan untuk mencegah bahaya/ancaman yang datang dari jaringan publik .

Tujuan Penggunaan

Terdapat beberapa tujuan penggunaan *firewall*, antara lain :

- a) *Firewall* biasanya digunakan untuk mencegah atau mengendalikan aliran data tertentu. Artinya, setiap paket yang masuk atau keluar akan diperiksa, apakah cocok atau tidak dengan kriteria yang ada pada standar keamanan yang didefinisikan dalam *firewall*.
- b) Untuk melindungi dengan menyaring, membatasi atau bahkan menolak suatu atau semua hubungan/kegiatan suatu segmen pada jaringan pribadi dengan jaringan luar yang bukan merupakan ruang lingkungannya. Segmen tersebut dapat merupakan sebuah *workstation*, *server*, *router*, atau *local area network* (LAN) seperti gambar 2.



Gambar 2 *Firewall* sebagai pembatas LAN dengan internet

(sumber : www.singapore-pc-servicing.com , diakses 7 jan 2009 10:21)

- c) penggunaan *firewall* yang dapat mencegah upaya berbagai trojan horses, virus, *phishin*, *spyware* untuk memasuki sistem yang dituju dengan cara mencegah hubungan dari luar, kecuali yang diperuntukan bagi komputer dan *port* tertentu seperti gambar 3.



Gambar 3 *Firewall* mencegah virus dan ancaman lain masuk ke jaringan

(sumber : <http://www.gwirken.nl/>, diakses 7 jan 2009 10:19)

- d) *Firewall* akan mem-*filter* serta meng-*audit traffic* yang melintasi perbatasan antara jaringan luar maupun dalam.

Teknik-Teknik yang Digunakan *firewall*

Adapun beberapa teknik yang digunakan dalam *firewall* adalah sebagai berikut :

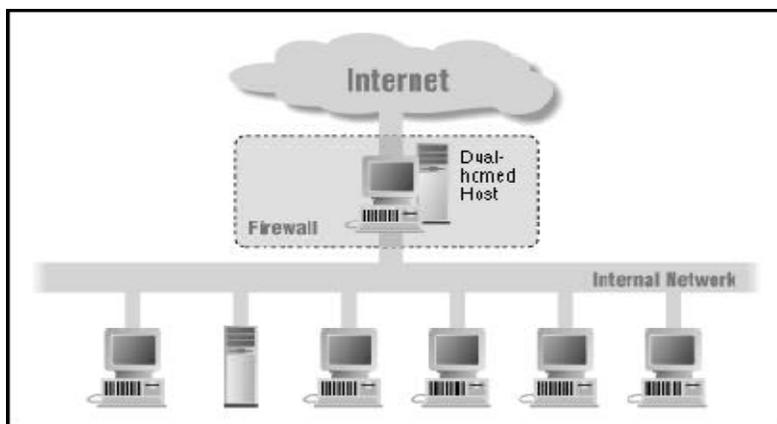
- ✓ *Service control* (kendali terhadap layanan) Berdasarkan tipe-tipe layanan yang digunakan di Internet dan boleh diakses baik untuk kedalam ataupun keluar *firewall*. Biasanya *firewall* akan mengecek no IP Address dan juga nomor port yang di gunakan baik pada protokol TCP dan UDP, bahkan bisa dilengkapi software untuk proxy yang akan menerima dan menterjemahkan setiap permintaan akan suatu layanan sebelum mengijinkannya. Bahkan bisa jadi software pada server itu sendiri , seperti layanan untuk web ataupun untuk mail.
- ✓ *Direction Control* (kendali terhadap arah) Berdasarkan arah dari berbagai permintaan (*request*) terhadap layanan yang akan dikenali dan diijinkan melewati *firewall*.
- ✓ *User control* (kendali terhadap pengguna) Berdasarkan pengguna/*user* untuk dapat menjalankan suatu layanan, artinya ada *user* yang dapat dan ada yang tidak dapat menjalankan suatu servis,hal ini di karenakan user tersebut tidak di ijinakan untuk melewati *firewall*. Biasanya digunakan untuk membatasi user dari jaringan lokal untuk mengakses keluar, tetapi bisa juga diterapkan untuk membatasi terhadap pengguna dari luar.
- ✓ *Behavior Control* (kendali terhadap perlakuan) Berdasarkan seberapa banyak layanan itu telah digunakan. Misalnya, *firewall* dapat memfilter email untuk menanggulangi/mencegah spam.

Arsitektur *Firewall*

Ada beberapa arsitektur atau konfigurasi dari *firewall*. Pada makalah ini hanya akan dijelaskan beberapa diantaranya, yaitu : *dual-homed host architecture*, *screened host architecture*, dan *screened subnet architecture*. Adapun penjelasannya dapat dijelaskan sebagai berikut.

- *Dual-homed host architecture*

Arsitektur *dual-home host* dibuat disekitar komputer *dual-homed host*, yaitu komputer yang memiliki paling sedikit dua *interface* jaringan. Untuk mengimplementasikan tipe arsitektur *dual-homed host*, fungsi routing pada *host* ini di non-aktifkan. Sistem di dalam *firewall* dapat berkomunikasi dengan *dual-homed host* dan sistem di luar *firewall* dapat berkomunikasi dengan *dual-homed host*, tetapi kedua sistem ini tidak dapat berkomunikasi secara langsung. Gambaran arsitektur ini seperti gambar 4.

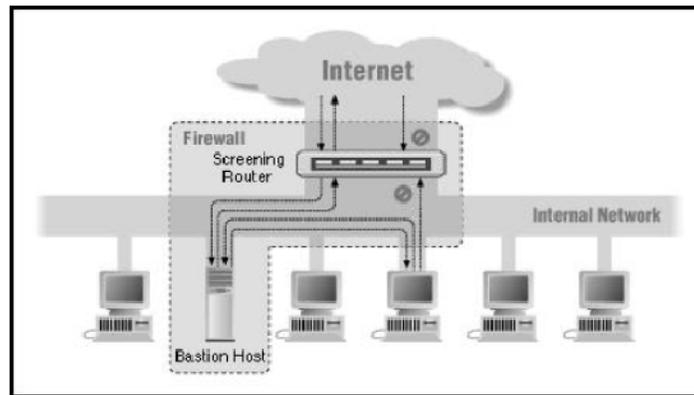


Gambar 4 *Dual-homed host architecture*

Dual-homed host dapat menyediakan *service* hanya dengan menyediakan proxy pada *host* tersebut, atau dengan membiarkan *user* melakukan *logging* secara langsung pada *dual-homed host*.

- **Screened host architecture**

Arsitektur *screened host* menyediakan *service* dari sebuah *host* pada jaringan internal dengan menggunakan router yang terpisah. Pada arsitektur ini, pengamanan utama dilakukan dengan *packet filtering* seperti gambar 5.

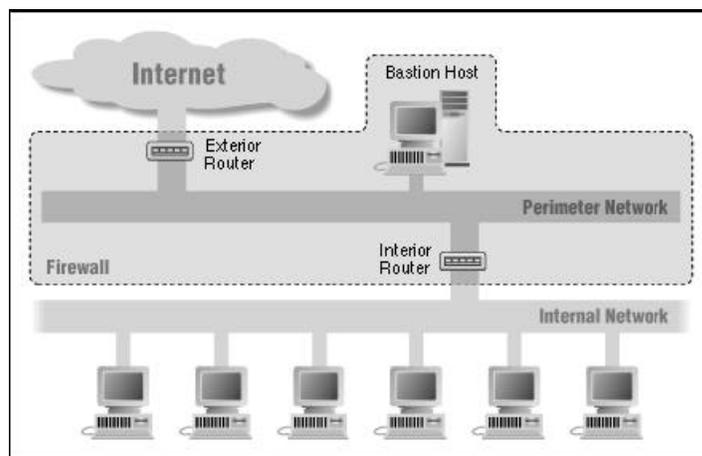


Gambar 5 Screened host architecture

Bastion host berada dalam jaringan internal. *Packet filtering* pada *screening router* dikonfigurasi sehingga hanya bastion host yang dapat melakukan koneksi ke Internet (misalnya mengantarkan mail yang datang) dan hanya tipe-tipe koneksi tertentu yang diperbolehkan. Tiap sistem eksternal yang mencoba untuk mengakses sistem internal harus berhubungan dengan host ini terlebih dulu. Bastion host diperlukan untuk tingkat keamanan yang tinggi.

- **Screened subnet architecture**

Arsitektur *screened subnet* menambahkan sebuah layer pengamanan tambahan pada arsitektur *screened host*, yaitu dengan menambahkan sebuah jaringan perimeter yang lebih mengisolasi jaringan internal dari jaringan Internet. Jaringan perimeter mengisolasi bastion host sehingga tidak langsung terhubung ke jaringan internal. Arsitektur *screened subnet* yang paling sederhana memiliki dua buah *screening router*, yang masing-masing terhubung ke jaringan perimeter. Router pertama terletak di antara jaringan perimeter dan jaringan internal, dan *router* kedua terletak di antara jaringan perimeter dan jaringan eksternal (biasanya Internet).



Gambar 6 Screened subnet architecture

Untuk menembus jaringan internal dengan tipe arsitektur *screened subnet*, seorang intruder harus melewati dua buah *router* tersebut sehingga jaringan internal akan relatif lebih aman. Gambar 6 menunjukkan gambar arsitektur *screened subnet*.

Tipe-Tipe Firewall

Ada beberapa tipe dari *firewall* yang ada. Selanjutnya akan dijelaskan secara lebih rinci seperti berikut. Ada empat jenis *firewall*, atau lebih tepatnya tiga jenis ditambah dengan satu tipe hybrid (campuran). Disini kita tidak akan membahas setiap jenis secara rinci karena itu membutuhkan pembahasan tersendiri yang lebih teknis dan umumnya sudah tersedia dalam dokumentasi-dokumentasi tentang *firewall* Arman (2007). Keempat jenis tersebut masing-masing adalah:

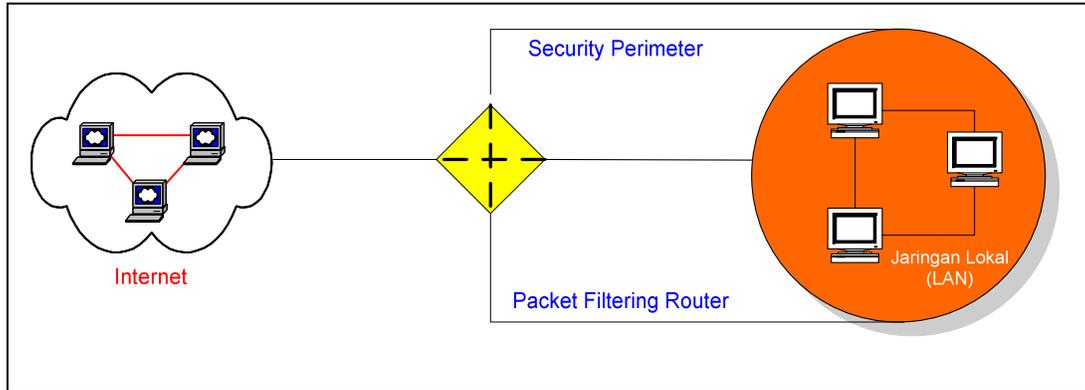
1. Packet Filtering Router

Firewall jenis ini memfilter paket data berdasarkan alamat dan pilihan yang sudah ditentukan terhadap paket tersebut. Ia bekerja dalam level internet protokol (IP) paket data dan membuat keputusan mengenai tindakan selanjutnya (diteruskan atau tidak diteruskan) berdasarkan kondisi dari paket tersebut. *Firewall* ini dapat digambarkan seperti gambar 7. *Firewall* jenis ini terbagi lagi menjadi tiga sub tipe:

- *Static Filtering*: Jenis *filter* yang diimplementasikan pada kebanyakan router, dimana modifikasi terhadap aturan-aturan filter harus dilakukan secara manual.
- *Dynamic Filtering*: Apabila proses-proses tertentu di sisi luar jaringan dapat merubah aturan filter secara dinamis berdasarkan even-even tertentu yang diobservasi oleh router (sebagai contoh, paket FTP dari sisi luar dapat diijinkan apabila seseorang dari sisi dalam me-request sesi FTP).
- *Stateful Inspection*: Dikembangkan berdasarkan teknologi yang sama dengan *dynamic filtering* dengan tambahan fungsi eksaminasi secara bertingkat berdasarkan muatan data yang terkandung dalam paket IP.

Baik dynamic maupun static filtering menggunakan tabel status (*state table*) dinamis yang akan membuat aturan-aturan filter sesuai dengan even yang tengah berlangsung. Muammar W. K (2004) menambahkan bahwa kelemahan tipe ini adalah cukup rumitnya untuk menyetting paket yang akan difilter secara tepat, serta lemah dalam hal autentikasi. Adapun serangan yang dapat terjadi pada *firewall* dengan tipe ini adalah:

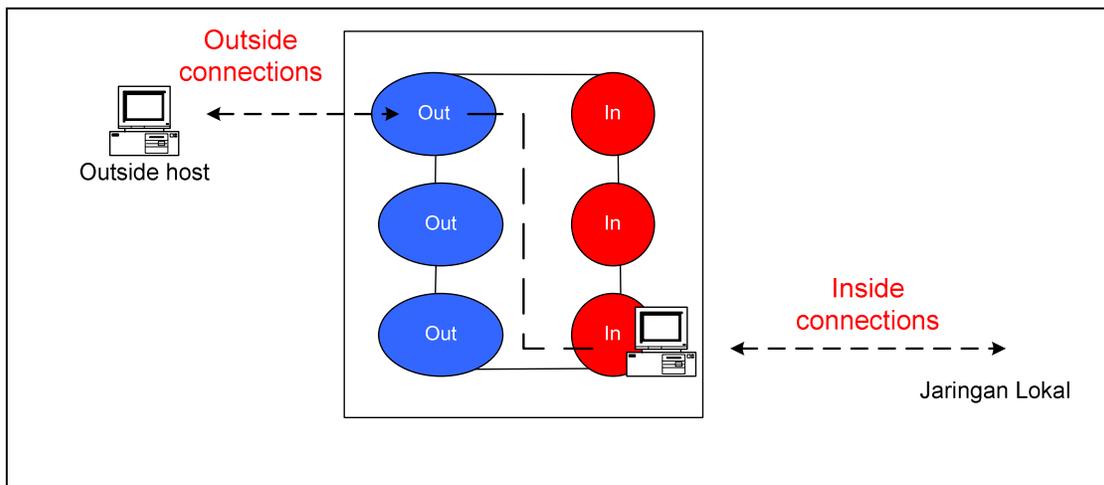
- ✱ *IP address spoofing* : Intruder (penyusup) dari luar dapat melakukan ini dengan cara menyertakan/menggunakan ip address jaringan lokal yang telah diijinkan untuk melalui *firewall*.
- ✱ *Source routing attacks* : Tipe ini tidak menganalisa informasi routing sumber IP, sehingga memungkinkan untuk membypass *firewall*.
- ✱ *Tiny Fragment attacks* : Intruder membagi IP kedalam bagian-bagian (*fragment*) yang lebih kecil dan memaksa terbaginya informasi mengenai TCP *header*. Serangan jenis ini di *design* untuk menipu aturan penyaringan yang bergantung kepada informasi dari TCP *header*. Penyerang berharap hanya bagian (*fragment*) pertama saja yang akan di periksa dan sisanya akan bisa lewat dengan bebas. Hal ini dapat di tanggulangi dengan cara menolak semua *packet* dengan protokol TCP dan memiliki Offset = 1 pada IP fragment (bagian IP)



Gambar 7 Packet filtering

2. Circuit Gateways

Firewall jenis ini beroperasi pada layer (lapisan) transpor pada network, dimana koneksi juga diautorisasi berdasarkan alamat. Sebagaimana halnya Packet Filtering, *Circuit Gateway* (biasanya) tidak dapat memonitor trafik data yang mengalir antara satu *network* dengan *network* lainnya, tetapi ia mencegah koneksi langsung antar *network*. Cara kerjanya adalah *gateway* akan mengatur kedua hubungan tcp tersebut, 1 antara dirinya (gw) dengan TCP pada pengguna lokal (inner host) serta 1 lagi antara dirinya (gw) dengan TCP pengguna luar (*outside host*). Saat dua buah hubungan terlaksana, *gateway* akan menyalurkan TCP segment dari satu hubungan ke lainnya tanpa memeriksa isinya. Fungsi pengamanannya terletak pada penentuan hubungan mana yang di ijin. Penggunaan tipe ini biasanya dikarenakan *administrator* percaya dengan pengguna *internal* (*internal users*). Dapat digambarkan seperti gambar 8.



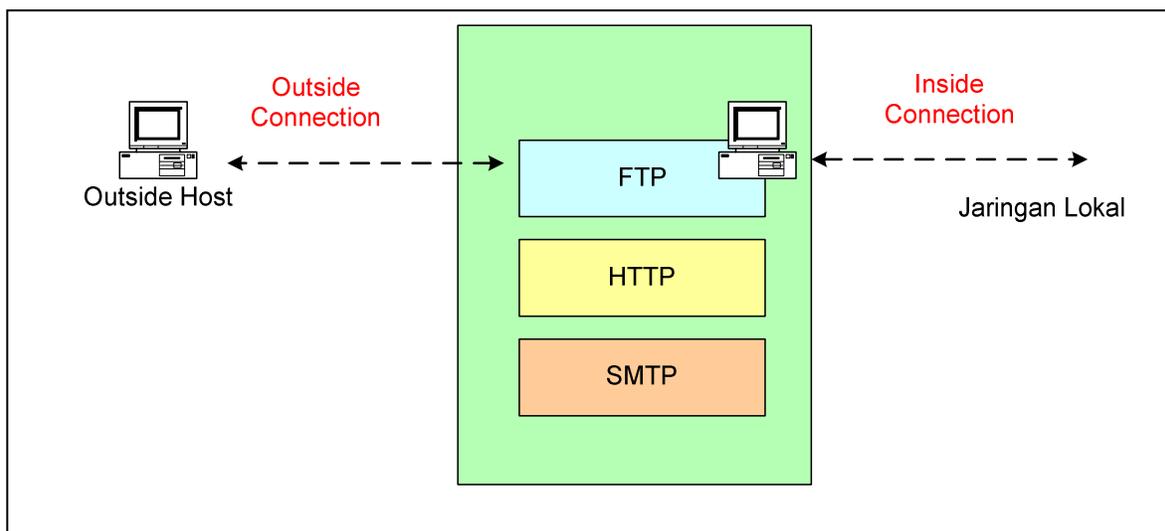
Gambar 8 Circuit Gateways

3. Application Gateways

Firewall tipe ini juga disebut sebagai *firewall* berbasis proxy. Ia beroperasi di level aplikasi dan dapat mempelajari informasi pada level data aplikasi (yang dimaksudkan disini adalah isi (*content*) dari paket data karena proxy pada dasarnya tidak beroperasi pada paket data). Filterisasi dilakukan berdasarkan data aplikasi, seperti perintah-perintah FTP atau URL yang diakses lewat HTTP. Dapat

dikatakan bahwa *firewall* jenis ini “memecah model *client-server*”. Cara kerjanya adalah apabila ada pengguna yang menggunakan salah satu aplikasi semisal FTP untuk mengakses secara remote, maka gateway akan meminta user memasukkan alamat *remote host* yang akan di akses. Saat pengguna mengirimkan *user ID* serta informasi lainnya yang sesuai maka gateway akan melakukan hubungan terhadap aplikasi tersebut yang terdapat pada *remote host*, dan menyalurkan data diantara kedua titik. apabila data tersebut tidak sesuai maka *firewall* tidak akan meneruskan data tersebut atau menolaknya. Lebih jauh lagi, pada tipe ini *Firewall* dapat di konfigurasi untuk hanya mendukung beberapa aplikasi saja dan menolak aplikasi lainnya untuk melewati *firewall*.

Kelebihannya adalah relatif lebih aman daripada tipe *packet filtering router* lebih mudah untuk memeriksa (audit) dan mendata (log) semua aliran data yang masuk pada level aplikasi. Kekurangannya adalah pemrosesan tambahan yang berlebih pada setiap hubungan. yang akan mengakibatkan terdapat dua buah sambungan koneksi antara pemakai dan *gateway*, dimana *gateway* akan memeriksa dan meneruskan semua arus dari dua arah. Agar lebih jelas, dapat digambarkan seperti gambar 9.



Gambar 9 Application Gateways

4. Hybrid Firewalls

Firewall jenis ini menggunakan elemen-elemen dari satu atau lebih tipe *firewall*. *Hybrid firewall* sebenarnya bukan sesuatu yang baru. *Firewall* komersial yang pertama, DEC SEAL, adalah *firewall* berjenis hybrid, dengan menggunakan proxy pada sebuah bastion hosts (mesin yang dilabeli sebagai “*gatekeeper*”) dan *packet filtering* pada gateway (“*gate*”). Sistem hybrid seringkali digunakan untuk menambahkan layanan baru secara cepat pada sistem *firewall* yang sudah tersedia. Kita bisa saja menambahkan sebuah *circuit gateway* atau *packet filtering* pada *firewall* berjenis *application gateway*, karena untuk itu hanya diperlukan kode proxy yang baru yang ditulis untuk setiap *service* baru yang akan disediakan. Kita juga dapat memberikan autentifikasi pengguna yang lebih ketat pada *Stateful Packet Filer* dengan menambahkan proxy untuk tiap *service*.

Sistem Pengamanan Menggunakan Firewall

Pada dasarnya kita manusia memerlukan privasi dimana kita dapat menuangkan seluruh pemikiran dan ide-ide yang muncul dipikiran kita. Dilihat dari segi penyerangan banyak jaringan yang terserang karena kurangnya pengawasan. Berangkat dari Pengetahuan akan jaringan terdapat dua tipe sistem pengamanan yang dapat dibuat sebagai implementasi dari *firewall*. Rodiah (2004) mengatakan tipe sistem pengamanan tersebut antara lain :

1. *Packet Filtering*.

2. *Proxy Services*.

Pada era abad ke-21 ini kita memerlukan suatu pengamanan yang terintegrasi. Di subbab yang berikutnya akan dijelaskan secara detil tentang dua tipe sistem pengamanan yang telah disebutkan diatas.

Packet Filtering

Sistem pada paket filtering merupakan sistem yang digunakan untuk mengontrol keluar, masuknya paket dari antara host yang didalam dan host yang diluar tetapi sistem ini melakukannya secara selektif. Sistem ini dapat memberikan jalan atau menghalangi paket yang dikirimkan, sistem ini sangat mengkitalkan arsitektur yang disebut dengan '*Screened Router*'. Router ini menjadi filter dengan menganalisa bagian kepala dari setiap paket yang dikirimkan. Karena bagian kepala dari paket ini berisikan informasi penting yaitu :

- ✓ IP *source address*.
- ✓ IP *destination address*.
- ✓ Protocol (dengan melihat apakah paket tersebut berbentuk TCP, UDP atau ICMP).
- ✓ Port sumber dari TCP atau UDP.
- ✓ Port tujuan dari TCP atau UDP.
- ✓ Tipe pesan dari ICMP.
- ✓ Ukuran dari paket.

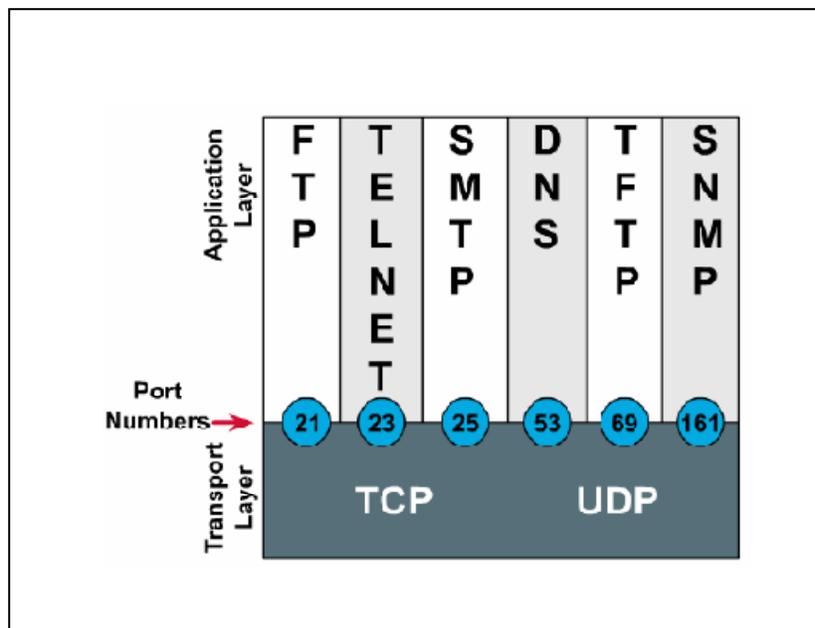
Cara Kerja Sistem *Packet Filtering* ini adalah mengawasi secara individual dengan melihat melalui router, sedangkan router yang telah dimaksud adalah sebuah perangkat keras yang dapat berfungsi sebagai sebuah server karena alat ini harus membuat keputusan untuk me-rout seluruh paket yang diterima. Alat ini juga harus menentukan seperti apakah pengiriman paket yang telah didapat itu kepada tujuan yang sebenarnya. Dalam hal ini router tersebut saling berkomunikasi dengan protokol-protokol untuk me-rout. Protokol yang dimaksudkan adalah *Routing Information Protocol* (RIP) atau *Open Shortest Path First* (OSPF) yang menghasilkan sebuah table routing. Tabel routing itu menunjukkan kemana tujuan dari paket yang diterima. Router yang menjadi filter pada packet filtering dapat menyediakan sebuah *choke point* (sebuah channel yang sempit yang sering digunakan untuk dipakai oleh penyerang sistem dan tentu saja dapat dipantau juga dikontrol oleh kita) untuk semua pengguna yang memasuki dan meninggalkan network. Karena sistem ini beroperasi ditingkat *Network Layer* dan *Transport Layer* dari tingkatan protokol pada tingkatan pada *Transmission Control Protocol* (TCP/IP). Bagian kepala dari *network* dan *transport* mengawasi informasi-informasi berikut:

- Protokol (IP *header*, pada *network layer*); didalamnya byte 9 mengidentifikasi protokol dari paket.
- *Source address* (IP *header*, pada *network layer*); alamat sumber merupakan alamat IP 32 bit dari host yang menciptakan oleh paket.
- *Destination address* (IP *header*, pada *network layer*); alamat tujuan yang berukuran 32 bit dari host yang menjadi tujuan dari paket.
- *Source port* (TCP atau UDP *header*, pada *transport layer*); pada setiap akhir dari koneksi TCP atau UDP tersambung dengan sebuah port, Walaupun port-port TCP terpisah dan cukup jauh dari port-port *user datagram protocol* (UDP). *Port-port* yang mempunyai nomor dibawah 1024 diterbalikan karena nomor-nomor ini telah didefinisikan secara khusus, sedangkan untuk *port-port* yang bernomor diatas 1024 (inklusif) lebih dikenal dengan *port*

ephemeral. Konfigurasi dari nomor pengalamatan ini diberikan sesuai dengan pilihan dari vendor.

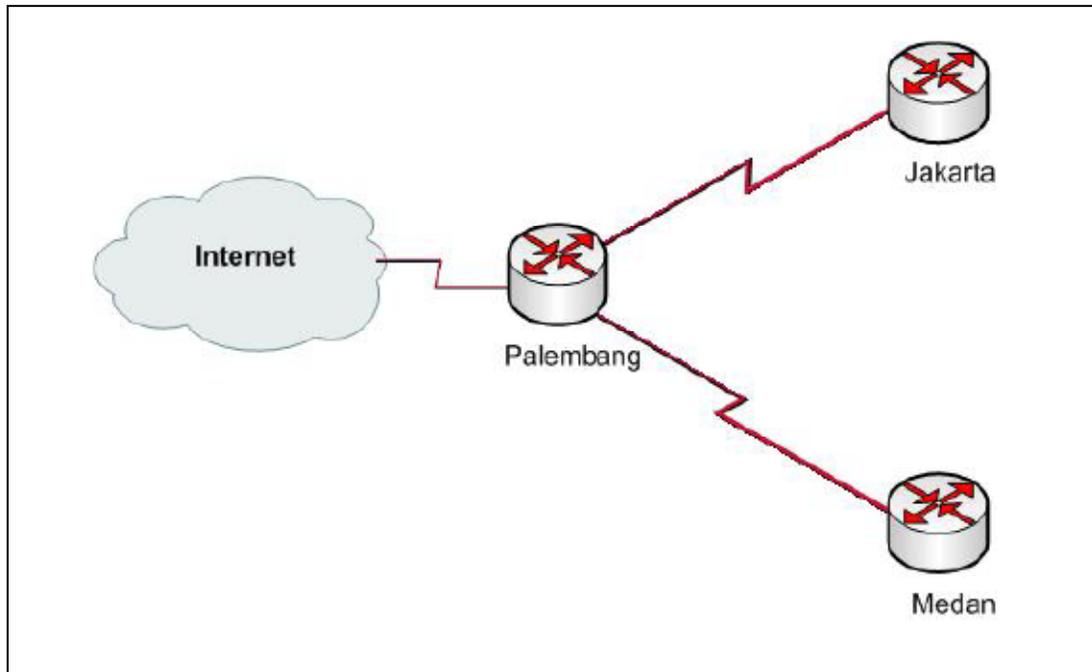
- *Destination port* (TCP atau UDP header, transport layer); nomor *port* dari tujuan mengindikasikan *port* yang dikirim paket. Servis yang akan diberikan pada sebuah *host* dengan mendengarkan *port*. Adapun *port* yang difilter adalah 20/TCP dan 21/TCP untuk koneksi ftp atau data, 23/TCP untuk telnet, 80/TCP untuk http dan 53/TCP untuk zona transfer DNS.
- *Connection status* (TCP atau UDP header, transport layer); status dari koneksi memberitahukan apakah paket yang dikirim merupakan paket pertama dari sesi di *network*. Jika paket merupakan paket pertama maka pada TCP header diberlakukan 'false' atau 0 dan untuk mencegah sebuah *host* untuk mengadakan koneksi dengan menolak atau membuang paket yang mempunyai bit set 'false' atau 0.

TCP & UDP menggunakan *port number* ini untuk membedakan pengiriman paket data ke beberapa aplikasi berbeda yang terletak pada komputer yang sama (Stiawan, 2008). Pada saat paket data di alamatkan ke tujuan, komputer tujuan harus mengetahui yang harus dilakukan pada paket tersebut, protocol TCP/IP menggunakan salah satu dari 65,536 pengalaman penomoran *port*. *Port number* inilah yang akan membedakan antara satu aplikasi dengan aplikasi lainnya atau satu protocol dengan protocol lainnya pada saat proses transmisi data antara sumber dan tujuan. Port number dapat digambarkan pada gambar 10.



Gambar 10 Port number Sumber: (Stiawan, 2008)

Untuk dapat melewati paket data dari sumber ke tujuan pada router terdapat protocol pengalaman atau routing protocol yang saling mengupdate antara satu dengan yang lainnya agar dapat melewati data sesuai dengan tujuannya. Di peralatan router layer 3 diperlukan konfigurasi khusus agar paket data yang masuk dan keluar dapat diatur, *Access Control List* (ACL) adalah pengelompokan paket berdasarkan kategori yang mengatur lalu lintas *network*. Dengan menggunakan ACL ini kita bisa melakukan *filtering* dan *blocking* paket data yang masuk dan keluar dari *network* atau mengatur akses ke sumber daya di *network* (Stiawan, 2008). Contoh sebuah topologi jaringan dengan menggunakan router dapat ditunjukkan oleh gambar 11.



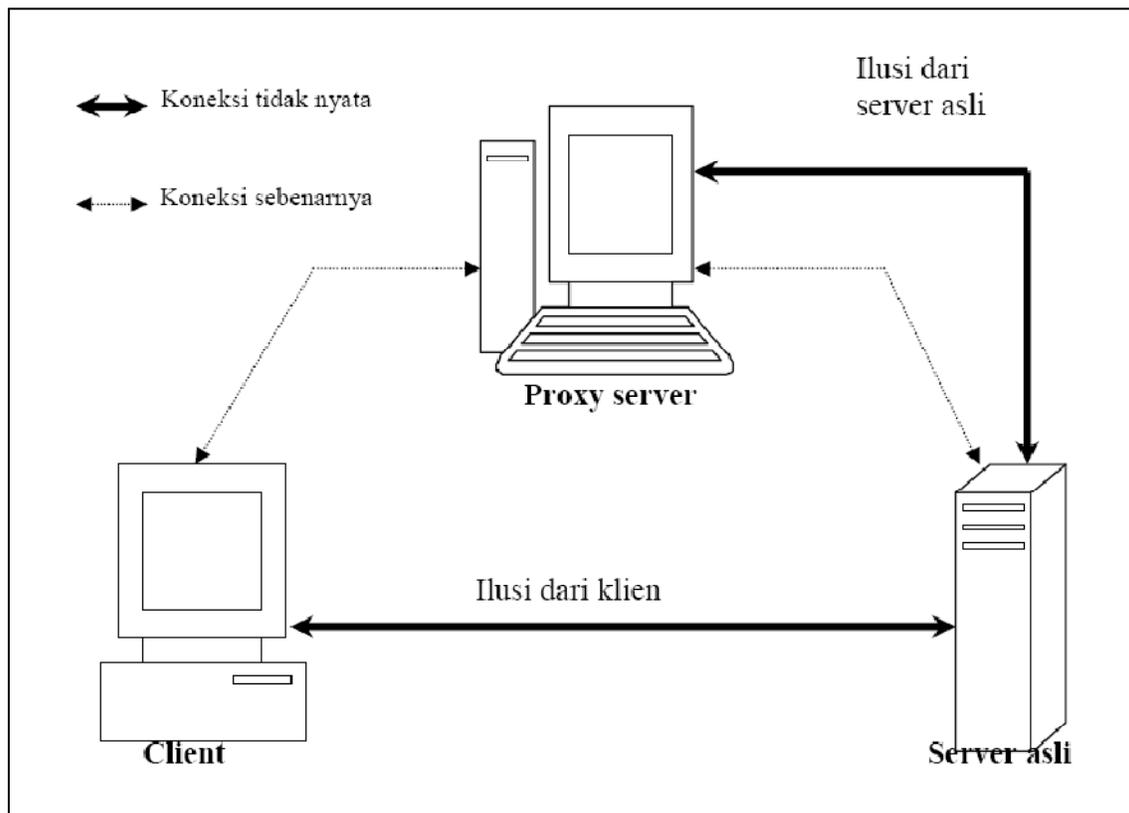
Gambar 11 Topologi jaringan dengan menggunakan router

Sumber: (Stiawan, 2008)

Proxy Services

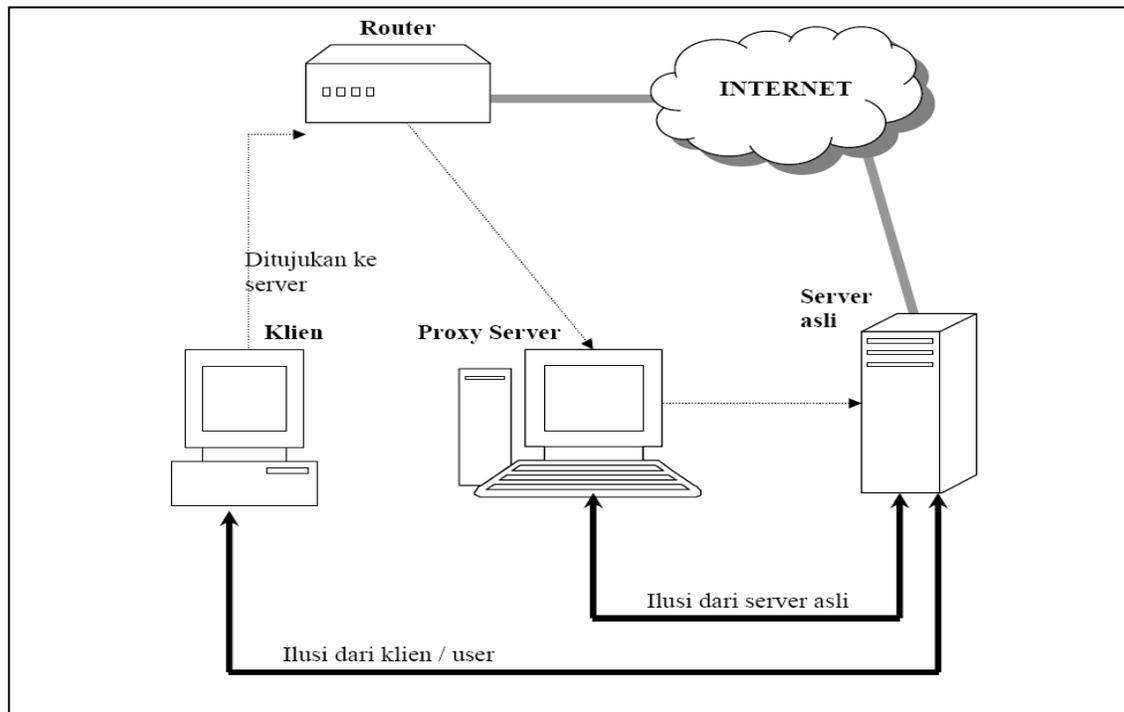
Proxy memberikan akses internet untuk satu buah *host* atau *host* yang dalam jumlah kecil dengan terlihat seperti menyediakan akses untuk seluruh *host* kita. Sebuah *proxy server* untuk protokol tertentu atau sebuah set dari protokol dapat dijalankan pada sebuah *dual-homed host* atau pada *bastion host*. Pada proxy ini sangat mendukung arsitektur dari *client/server*. *Client/server* ini membentuk sebuah sistem dimana komponen-komponen dari *software* saling berinteraksi. Dalam hal ini para klien dapat meminta seluruh kebutuhan dan pelayanan yang diinginkan dan server menyediakannya. Sistem proxy ini harus mendukung seluruh pelayanan yang diminta dan diperlukan oleh klien. Karena hal ini maka *server* harus mempunyai *file server* yang sangat besar dan selalu aktif dimana file-file yang terdapat pada server akan digunakan oleh setiap komputer yang terhubung baik dalam *Lokal Area Network (LAN)* ataupun *Wide Area Network (WAN)*. Pada *file server* selain dari *list* yang cukup panjang sebagai *database* yang dapat digunakan oleh setiap klien yang akan menggunakan alamat IP yang legal, terdapat juga *file-file* untuk aplikasi yang bekerja pada *server* utama. Proxy merupakan sistem pengamanan yang memerlukan alamat IP yang jelas dan valid, karena server yang utama terdapat di internet. Pada proxy terdapat empat pendekatan yang akan dilakukan pada sisi klien yang sangat berperan penting. Pendekatan-pendekatan tersebut yaitu :

- ✱ *Proxy-aware application software*. Dengan pendekatan ini software harus mengetahui bagaimana untuk membuat kontak dengan proxy server daripada dengan server yang sebenarnya ketika user membuat suatu permintaan; dan bagaimana memberitahukan proxy server, server asli yang mana yang harus dibuatkan koneksi. Ini terlihat pada gambar 12.



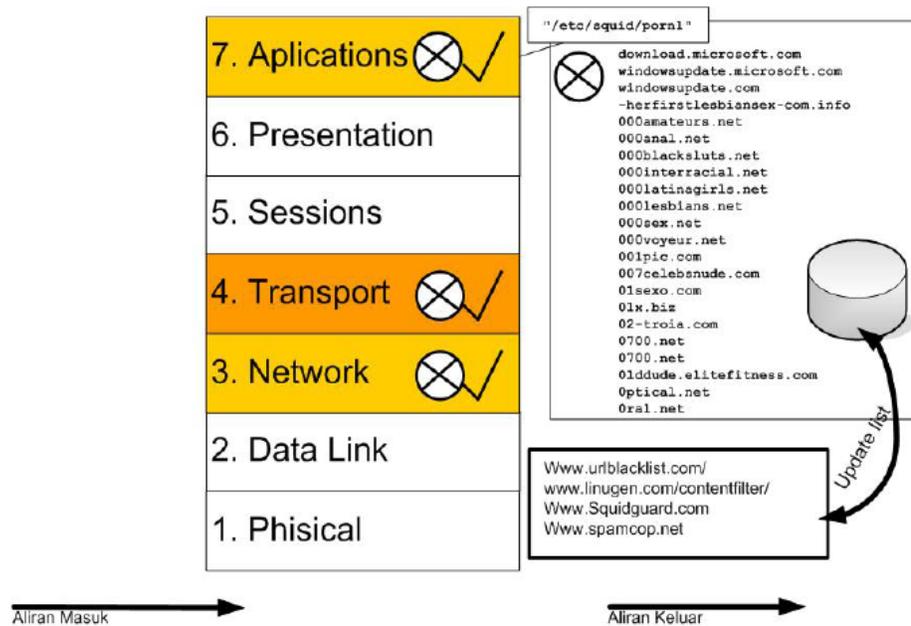
Gambar 12 Software untuk proxy-aware pada sistem proxy

- * *Proxy-aware operating system software.* Dengan pendekatan ini, sistem operasi yang dijalankan oleh user sudah harus dimodifikasikan sehingga koneksi IP yang sudah diperiksa untuk apakah paket tersebut harus dikirimkan kepada proxy server. Mekanisasi dari ini sangat bergantung sekali pada runtime linking yang dinamis (kemampuannya untuk memberikan library ketika program dijalankan). mekanisme ini tidak selalu berjalan dengan mulus dan dapat gagal yang tidak wajar untuk *user*.
- * *Proxy-aware user procedures.* Pendekatan ini pengguna menggunakan *software client* yang tidak mengerti bagaimana me-proxy, dimana untuk berbicara (berkomunikasi) ke server proxy dan memberitahukan proxy server untuk melakukan hubungan kepada *server* yang sebenarnya daripada memberitahukan *software* klien untuk berkomunikasi secara langsung ke *server* yang sebenarnya.
- * *Proxy-aware router.* Pendekatan yang terakhir ini *software* yang klien gunakan tidak dimodifikasikan tetapi sebuah router akan mengantisipasi koneksi dan melangsungkan ke proxy server atau proxy yang diminta. Mekanisme ini membutuhkan sebuah router yang pintar disamping *software* proxy (meskipun me-proxy dan me-rout tidak bisa tampil pada mesin yang sama). Mekanisme pada proxy ini diperlihatkan pada gambar 13.



Gambar 13 Proxy-aware router pada sistem proxy

Penggunaan *Proxy Server* dapat dijadikan solusi untuk melakukan *screening* dan *blocking* di *layer* 7, dengan menggunakan proxy dapat menyaring paket-paket berdasarkan *policy* yang dibuat, misalnya berdasarkan alamat web tertentu.



Gambar 14 Filtering content Web

Sumber: (Stiawan, 2008)

Blocking dengan *proxy* dapat dioptimalkan dengan menyaring alamat-alamat web yang mengandung *content pornography*, kekerasan, virus atau trojan, ilegal software dan sebagainya. Pada gambar 14 terlihat metode *filtering* di *layer 7* bisa menyaring content website berdasarkan *URL* yang tidak diperbolehkan mengakses ke jaringan kita, baik paket data yang keluar atau paket data yang masuk.

Kesimpulan

Berdasarkan penjelasan yang sudah disampaikan, dapat diambil beberapa kesimpulan yaitu keberadaan suatu *firewall* sangat penting digunakan dalam suatu jaringan yang terkoneksi langsung ke internet atau yang lebih dikenal dengan jaringan publik yang dapat diakses oleh siapapun dan dimanapun. Sehingga peran *firewall* disana sangat berguna karena sebagai pembatas yang mengatur dan mengendalikan akses yang dilakukan untuk mengurangi dan mencegah ancaman-ancaman dari internet yang masuk ke jaringan lokal.

Referensi

- [1] Arman, 2007, Firewall dari Masa ke Masa, <http://unms.unimal.ac.id/> diakses : 7 Januari 2009 [9:30]
- [2] Muammar W. K, 2004, Firewall, <http://ilmukomputer.com> diakses : 7 Januari 2009 [9:41]
- [3] Rodiah, 2004, *Sistem Keamanan Firewall Dengan Packet Filtering*,
Universitas Gunadarma : Jakarta
- [4] Stiawan, D., 2008, *Kombinasi Firewall di OSI Model*, FASILKOM UNSRI: Palembang



Agus Aan Jiwa Permana, lahir di Denpasar tanggal 4 Agustus 1987. Adapun riwayat pendidikan adalah sebagai berikut : Menamatkan pendidikan sekolah dasar di SDN 4 Peraan, melanjutkan ke SLTP N 2 Baturiti, kemudian masuk di SMU N 1 Tabanan, dan setelah tamat SMU tertarik terhadap IT sehingga melanjutkan kuliah di Universitas Pendidikan Ganesha (Undiksha) Singaraja-Bali, mengambil jurusan D-3 Manajemen Informatika.

Saat kuliah aktif di dalam organisasi kemahasiswaan seperti himpunan mahasiswa jurusan (HMJ), SENAT, dan UKM. Pada tahun 2007 lulus dari Undiksha dan kemudian melanjutkan S1 ke Yogyakarta di Universitas Gadjah Mada (UGM), mengambil program studi Ilmu Komputer di bawah jurusan Matematika.

Saat ini masih berstatus sebagai mahasiswa aktif di UGM. Berkeinginan menjadi seorang pengembang IT. Tertarik terhadap jaringan komputer(*Network*), sistem pendukung keputusan (SPK), dan sistem informasi. Saat ini sedang mengambil skripsi dengan minat sistem cerdas.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:

URL : <http://agus-aan.web.ugm.ac.id>

Email : studywithaan@gmail.com

agus-aan@mail.ugm.ac.id