

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu ini menjadi satu acuan penulis dalam membuat laporan akhir sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Pada penelitian sebelumnya, penulis menemukan judul penelitian yang identik sehingga dapat dijadikan sebagai referensi penulis dalam mengerjakan penelitian. Berikut merupakan referensi penelitian terdahulu berupa beberapa jurnal yang terkait dengan judul laporan akhir penulis.

Pada referensi penelitian pertama karangan Luan, Samuel Berek (2007) yang berjudul “Membangun *Firewall* Pada Sistem Operasi *FreeBSD 5.4* Dengan Menggunakan Aplikasi *IPFIREWALL* (IPFW)”. Penelitian tersebut membahas tentang perancangan aplikasi *firewall* berbasis *FreeBSD* yang digunakan untuk membatasi akses website. Adapun cara kerja penelitian ini adalah melindungi jaringan *local* dari jaringan publik (internet) dengan membatasi ataupun menolak suatu koneksi pada jaringan dengan *firewall* berbasis *FreeBSD*. Penelitian ini menyatakan bahwa *firewall* dapat mengatur paket data. *Firewall* dapat memblokir situs terlarang dan mengalihkannya ke IP *server* serta melewatkan situs-situs tertentu. Berdasarkan hasil penelitian tersebut, penulis menyimpulkan bahwa *firewall* berbasis *FreeBSD* dapat menjaga keamanan data di *server* dan *client* yang terhubung ke jaringan computer. Penulis juga menyarankan agar tidak menggunakan *FreeBSD* untuk mengalihkan suatu situs ke situs lainnya karena *FreeBSD* hanya bisa mengalihkan ke IP *server*.

Pada referensi penelitian kedua karangan Wahyudi, Arif Riyan dan Imam Riadi (2015) yang berjudul “Analisis *Firewall Layer 7* dan Metode *Caching* Untuk Optimasi Jaringan Komputer”. Penelitian tersebut membahas tentang perancangan alur sistem yang diperlukan untuk mengetahui langkah kerja *firewall layer 7* dan metode *caching* yang akan diimplementasikan. Adapun cara kerja penelitian ini adalah memblokir situs dan file dengan jenis 3gp yang tidak boleh

masuk ke dalam jaringan lokal Laboratorium Riset Kampus 3 UAD menggunakan fungsi *access list* dan *regex* kemudian diterapkan metode *caching* yang berfungsi menyimpan halaman yang pernah diakses oleh *user*, sehingga halaman yang diakses lebih cepat serta mengurangi pemakaian *bandwidth request*. Berdasarkan hasil penelitian tersebut, penulis menyimpulkan implementasi penggunaan teknologi *firewall layer 7* dan *caching* dapat meningkatkan keamanan jaringan lokal dari situs pornografi dan file dengan jenis 3gp yang sering dicurigai membawa berbahaya, serta mempersingkat waktu akses ke halaman *website*.

Pada referensi penelitian ketiga karangan Hambali (2018) yang berjudul “Membangun Blocking Situs Dengan Menggunakan *Web Proxy* Mikrotik RB750 Guna Mendukung Internet Sehat”. Dalam penelitiannya peneliti menggunakan mikrotik *winbox* yaitu untuk memblokir situs pada internet yang tidak boleh diakses oleh server tersebut untuk itu perlu dilakukan upaya pencegahan terhadap informasi negatif. Berdasarkan hasil penelitian tersebut, penulis menyimpulkan bahwa web proxy mikrotik os rb750 dapat berkerja dengan baik, pengaturan waktu pemblokiran pada saat-saat tertentu seperti jam belajar akan mendukung Internet sehat. Pada contoh blocking situs penulis tidak betul-betul memblokir situs negatif karena penulis berpikir khawatir turut menyebarluaskan situs negatif tersebut. Akhirnya Sistem blocking situs ini telah berhasil dibuat pada laboratorium komputer jaringan STMIK Royal Kisaran.

Pada referensi penelitian keempat karangan Hidayat, Arif (2018) yang berjudul “Komparasi Analisis Mikrotik Halaman Filter Menggunakan Beberapa Metode Seperti *Filtering IP Address, Layer 7 Protocols, Web Proxy, Mangle* dan DNS Berhasil Melakukan Blok Pada Situs *Facebook* dan *Youtube*”. Penelitian tersebut membahas tentang pengujian untuk mengetahui bagaimana performansi dari teknik-teknik tersebut dalam penyaringan situs/konten, pada pengujiannya terlihat hasil output *outer browser* terhadap *site filtering*. Berdasarkan hasil penelitian tersebut, penulis menyimpulkan bahwa analisis perbandingan *Site Filter* pada Mikrotik menggunakan Teknik Daftar Alamat, *Protokol Layer7, Proxy*

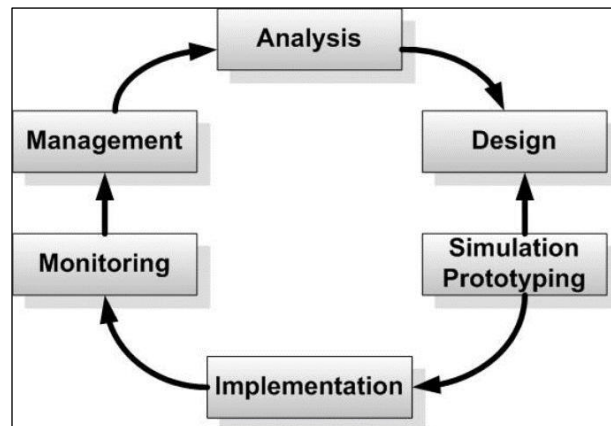
Web, *Mangle* dan *DNS Statis*. Lima teknik *filter* situs berhasil memblokir *youtube.com*, dan *facebook.com* dengan keluaran *filter* yang berbeda.

Pada referensi penelitian terakhir karangan Yohaness, Febrison (2018) yang berjudul “Perancangan dan Implementasi Policy Management Layer 7 di Batam Tourism Polytechnic”. Cara kerja penelitian ini adalah memudahkan network administrator atau staf IT Center Batam Tourism Polytechnic mengawasi traffic jaringan internet di lingkungan kampus dan Mahasiswa-mahasiwi serta staf di Batam Tourism Polytechnic dapat menggunakan internet untuk hal-hal yang berhubungan dengan kegiatan belajar mengajar di kampus serta menunjang kinerja pelayanan staf kampus kepada mahasiswa-mahasiswi Batam Tourism Polytechnic. Berdasarkan hasil penelitian tersebut, penulis menyarankan agar mengalokasikan dana untuk membangun Proxy Server yang dapat mengoptimalkan fungsi dari policy management layer7 sehingga pengguna internet tidak dengan mudah melakukan bypass dari pemblokiran situs serta ekstensi video streaming.

2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan penulis pada laporan ini menggunakan metode NDLC (*Network Development Life Cycle*). Menurut Goldman dan Rawles (2004:470) *Network Development Life Cycle* (NDLC) adalah metode yang dapat digunakan untuk mengembangkan suatu jaringan komputer.

Berikut merupakan tahapan dari metode NDLC sebagai berikut:



Gambar 2.1 Ilustrasi Model NDLC

Adapun penjelasan dari tahapan diatas yaitu sebagai berikut :

1. Tahap *Analysis*, Tahapan awal yang dilakukan dalam menganalisis adalah analisa kebutuhan, analisa permasalahan yang ada, analisa keinginan *user*, dan analisa topologi jaringan yang sudah ada, tahapan ini adalah tahapan pengumpulan data yang dibutuhkan untuk perumusan masalah dalam menyelesaikan kendala yang ada. Dengan mengidentifikasi sistem yang sedang berjalan lalu mencoba untuk menganalisa suatu pengembangan sistem seperti apa yang akan diterapkan pada sistem tersebut,
2. Tahap *Design*, Tahap ini dari data-data yang didapatkan sebelumnya, tahapan desain ini penulis akan membuat desain gambar topologi jaringan yang akan dibangun, desain akses data dan sebagainya.
3. Tahap *Simulation Prototyping*, Tahap ini melakukan pengembangan jaringan yang akan membuat dalam bentuk simulasi. Hal ini dimaksudkan untuk melihat kinerja dari *network* yang akan dibangun dan menjadi bahan presentasi dan *sharing* dengan pengembangan *system* jaringan,
4. Tahap *Implementation*, Tahap ini akan sedikit memakan waktu lama. dalam melakukan implementasi, penulis telah menerapkan semua yang direncanakan dan dirancang sebelumnya. Pada tahapan ini akan terlihat bagaimana pengembangan yang akan dibangun akan memberikan pengaruh terhadap system yang ada.

5. Tahap *Monitoring*, Tahap ini setelah diimplementasi, tahapan monitoring merupakan tahapan penting agar jaringan dan komunikasi dapat berjalan sesuai dengan keinginan dan tujuan penulis pada tahap awal analisis. Penulis akan menggunakan *tools* yang ada di mikrotik yang berfungsi untuk memonitor lalu lintas jaringan.
6. Tahap *Management*, Tahap ini salah satu yang menjadi perhatian khusus adalah masalah kebijakan, yaitu dalam hal aktivitas, pemeliharaan dan pengelolaan dikategorikan pada tahap ini. Kebijakan perlu dibuat untuk membuat dan mengatur agar sistem yang telah dibangun dan berjalan dengan baik dapat berlangsung lama dan unsur *reliability* terjaga.

2.3 Sejarah FreeBSD

Freebsd merupakan salah satu dari sekian varian *unix BSD* selain *NetBSD*, dan *OpenBSD*. Sebelumnya, perjalanan varian *unix BSD* dimulai pada tahun 1973, dimana pada waktu itu Prof Bob Fabry dari Universitas *California Berkeley* menyatakan minat untuk mendapatkan sistem operasi *Unix* kepada Ken Thompson dan Dennis Ritchie pada kegiatan "Symposium on Operating Systems Principles" di Universitas Purdue. Prof Bob Fabry bermaksud mendapatkan *Unix* untuk eksperimen pada sebuah *mainframe* milik Universitas *Berkeley*. Pada tahun 1974 sebuah *tape* yang berisi *Unix versi 4* datang ke *Berkeley* dan di install- kan oleh mahasiswa pasca sarjana Keith Standiford pada komputer PDP-11/45. Meskipun dianggap cukup mudah dalam menginstal *unix*, namun pada kenyataannya berbagai masalah dihadapi oleh Keith Standiford dalam menjalankan *Unix* pada komputer tersebut.

Pada tahun 1975, *department Ilmu Komputer Universitas California Berkeley* membeli komputer baru sebuah DEC 11/70. Pada tahun yang sama Ken Thompson menjadi Profesor Tamu pada almamaternya yaitu Universitas *California Berkeley*, Ken Thompson datang dengan membawa sistem operasi *Unix versi 6*. Dua orang mahasiswa pasca sarjana yaitu Bill Jolitz's dan Chuck Haley membantu Ken Thompson untuk meng-hacked *Unix versi 6* tersebut pada komputer DEC 11/70.

Pada akhir musim panas 1976, Ken Thompson kembali ke Bell Labs New Jersey, seiring dengan kepergian Ken Thompson, Bill Jolitz's dan Chuck Haley mulai *mengoprek* kernel sistem operasi Unix versi 6 tersebut, berbekal dengan pengalaman satu tahun terakhir mengoprek Unix bersama Ken Thompson sebelumnya.

Akhirnya pada awal tahun 1977, Bill Jolitz's mengeluarkan versi free dari "*Berkeley Software Distribution*", pada distribusi pertama mencakup pula compiler Pascal dan editor Ex. Pada tahun 1978 Bill Jolitz's memutuskan software yang ada pada distribusi harus diperbaharui seiring dengan banyaknya feedback dari komunitas, hasilnya pada tahun 1978 tersebut keluar "*Second Berkeley Software Distribution*" atau disingkat *2BSD*, termasuk didalamnya *compiler Pascal, editor vi* dan *termcap*.

Pada tahun 1978, Departemen Ilmu Komputer Universitas *Berkeley*, membeli sebuah komputer VAX-11/780 dari DEC, meskipun komputer tersebut sudah memiliki sistem operasi sendiri yang dikenal dengan nama VMS, namun Departemen Ilmu Komputer menginginkan Unix 32/v (Seventh Edition) dapat berjalan diatas komputer VAX-11/780 tersebut. Lagi-lagi Bill Jolitz's diminta membantu melakukan porting Unix 32/V tersebut untuk mesin VAX-11/780, pada awal Januari 1979, akhirnya Unix 32/V (Seventh Edition) dapat berjalan dengan mulus pada komputer VAX tersebut, pada saat itu juga Bill Jolitz's, memutuskan untuk melakukan porting 2BSD untuk komputer VAX dengan pertimbangan komputer VAX tersebut jauh lebih canggih (berarsitektur 32 bit) daripada PDP-11 yang hanya 16 bit.

Pada bulan Januari 1979 distribusi lengkap telah diselesaikan hasilnya 3BSD sebagai distribusi sistem VAX pertama dari Berkeley. Pada musim gugur 1979, Prof Bob Fabry, merepson keinginan DARPA (Defense Advanced Research Projects Agency) untuk memperbaiki 3BSD untuk kepentingan komunitas DARPA, dimana pada waktu itu untuk keperluan mengkoneksikan semua komputer pada pusat-pusat riset. Untuk lebih memantapkan pekerjaan

dari DARPA tersebut, Prof Bob Fabry membentuk CSRG (*Computer System Research Group*).

Pada Oktober 1980 lahir 4BSD, selama 9 bulan kedepan sejak kelahirannya sebanyak 150 kopi telah dikirimkan. Lisensi dibuat berdasarkan institusi bukan per komputer. Karena sudah tersebar luas 4BSD banyak menuai kritik terutama masalah kinerja yang dinilai masih lamban daripada VMS. Untuk itu pada Juni 1981, 4.1BSD lahir dengan berbagai macam perbaikan. Pada awalnya distribusi tersebut akan diberi nama 5BSD, namun pihak AT&T keberatan karena akan membingungkan pelanggan, karena pada saat itu terdapat juga sistem operasi Unix system V, untuk itu Berkeley mengalah dan memberi nama distribusi tersebut 4.1BSD. Sebagai pendahuluan *release* pada April 1982, dikeluarkan 4.1aBSD untuk keperluan lokal saja (Berkeley dan DARPA), pada saat itu banyak kritik dan saran perbaikan untuk 4.1aBSD, untuk itu pada Juni 1982 dikeluarkan 4.1bBSD. Release 4.1b BSD ini cukup stabil dan baik maka pada April 1983 dikeluarkan 4.1c BSD. Dengan sedikit perbaikan pada 4.1c BSD, pada Agustus 1983 dikeluarkan 4.2BSD. 4.2BSD pada saat itu sangat populer, lebih dari 1000 institusi mempunyai lisensi 4.2BSD tersebut, para vendor pun pada saat itu lebih suka menawarkan 4.2BSD ketimbang *Unix system V* karena 4.2BSD mempunyai fasilitas *Networking* dan *Fast File System*.

Dengan berbagai macam kritik dan *feedback*, maka pada tengah 1986 di-*release* 4.3BSD, selanjutnya pada Juni 1988 di-*release* 4.3BSD Tahoe dan pada Juni 1990 di-*release* 4.3BSD Reno. Selain *release* tsb ada pula *release networking* yaitu: 4.3BSD Net1 pada Maret 1989 dan 4.3BSD Net2 pada Juni 1991.

Release ini tidak memiliki *source code* yang bersifat *proprietary* sehingga dapat secara bebas didistribusikan dalam bentuk *source code* maupun *binary*. *Release* terakhir dari CSRG adalah 4.4BSD, pada saat yang bersamaan juga CSRG me-*release* 4.4BSD-Lite yang berisi *source code non-proprietary* dan users tidak perlu memiliki lisensi Unix, namun 4.4BSD-Lite ini mendapat

aksi legal dari USL (*Unix System Laboratories*) yang mengklaim 4.4BSD-Lite mengandung source code asli Unix dari AT&T, hal ini berlanjut hingga ke pengadilan. Setelah 1 tahun proses pengadilan berlangsung akhirnya Bill Jolitz berwenang untuk mengambil bagian dari software yang bukan AT&T dan mengembalikannya menjadi *free UNIX*. Ini adalah awal lahirnya modern BSD. Pada tahun 1992 dan 1993, Jordan K Hubbard, Rod Grimes, dan Nate Williams yang menangani proyek 386BSD, merilis sebuah paket yang dikenal sebagai “Unofficial 386BSD Patchkit”. Dari proses *maintain patchkit* tersebut melahirkan mekanisme baru yang membentuk “*386BSD 0.5*”, yang berisi perubahan dan fungsi baru sebagai “the real operating system”. Bagaimanapun, Jolitz mencabut persetujuannya pada proyek *patchkit* tersebut pada tahun 1993. David Greenman kemudian mengajukan usulan sebuah sistem operasi baru dengan basis patchkit tersebut menjadi sebuah “*FreeBSD*”.

Hubbard akhirnya bekerjasama dengan David Greenman (Walnut Creek) untuk mempersiapkan sebuah penanganan distribusi CDROM. Rilis CDROM pertama dari *FreeBSD 1.0* dilakukan pada bulan Desember 1993. Dengan mengupgrade basis *FreeBSD* dari *Net/2* ke *4.4BSD Lite*. *FreeBSD 2.0* dirilis pada bulan November 1994 dan terus berkembang sampai sekarang yang telah mencapai *release 6.2*.



Gambar 2.2 Logo *FreeBSD*

2.4 Jaringan Komputer

Menurut Sofana (2014) Jaringan komputer (*computer network*) merupakan himpunan interkoneksi sejumlah komputer autonomus. Dalam bahasa populernya dapat dijelaskan bahwa jaringan komputer merupakan kumpulan beberapa komputer yang saling terhubung dengan lain melalui media perantara seperti media kabel ataupun media tanpa kabel (nirkabel).

Adapun manfaat jaringan komputer adalah sebagai berikut :

1. Sharingresources
2. Mediakomunikasi
3. Integrasidata
4. Sumber daya lebih efisien dan informasiterkini

2.4.1 Pengelompokan Jaringan (*Network*)

Dilihtdari sisi lingkupannya atau jangkauannya, jaringan dapat di bagi menjadi beberapa jenis, yaitu :

1. LAN (*Local AreaNetwork*)

Local Area Network merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantorsuatu perusahaan atau pabrik-pabrik untuk memakai bersama sumber daya (*resource*, misalnya printer) dan saling bertukar informasi.

2. MAN (*Metropolitan AreaNetwork*)

Metropolitan Area Network merupakan jaringan yang meliputi area geografis yang lebih luas, seperti suatu kota. Dengan interkoneksi jaringan dalam area geografis yang luas, informasi dapat disebarkan secara mudah melalui jaringan.

Dengan MAN suatu komputer di kantor cabang dapat berhubungan dengan server komputer yang ada pada kantor pusat melalui jaringan telpon, kabel koaksial atau komunikasi tanpa kabel.

3. WAN (*Wide Area Network*)

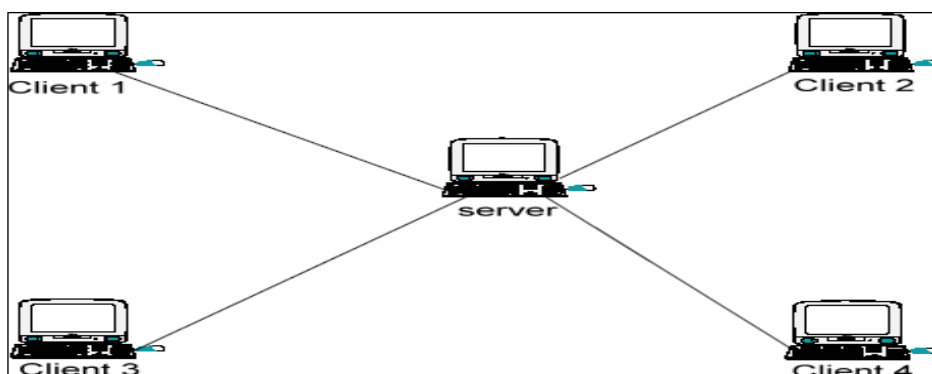
Wide Area Network merupakan jaringan yang meliputi area geografis yang lebih luas lagi, yang meliputi suatu negara atau dunia. Umumnya jaringan ditempatkan pada banyak lokasi yang berbeda. WAN digunakan untuk menghubungkan banyak LAN yang secara geografis terpisah. WAN dibuat dengan cara menghubungkan LAN menggunakan layanan seperti dial-up dan satelit.

2.4.2 Tipe Jaringan Komputer

Menurut fungsi komputer pada sebuah jaringan, maka tipe jaringan komputer dapat dibedakan menjadi 2 tipe, yaitu.

1. Client – Server

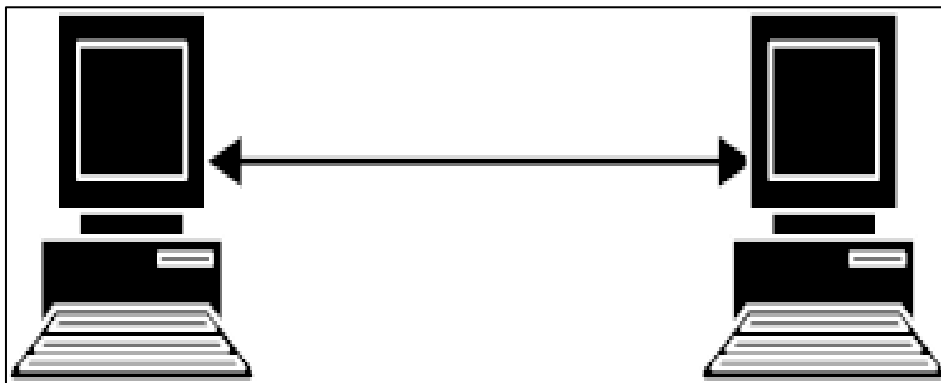
Dimana sebuah server atau lebih yang dihubungkan dengan beberapa *client*. *Server* bertugas menyediakan layanan, bermacam-macam jenis layanan yang dapat diberikan oleh *server*, misalnya adalah pengaksesan berkas, *peripheral*, *database*, dan lain sebagainya. Sedangkan *client* adalah sebuah terminal yang menggunakan layanan tersebut. Perbedaannya dengan hubungan *dumb terminal*, sebuah *terminal client* melakukan pemrosesan data di terminalnya sendiri dan hal itu menyebabkan spesifikasi dari *server* tidaklah harus memiliki performansi yang tinggi, dan kapasitas penyimpanan data yang besar karena semua pemrosesan data yang merupakan permintaan dari *client* dilakukan di terminal *client*.



Gambar 2.3 Tipe Jaringan Client – Server

2. Peer to Peer

Dimana terdapat beberapa terminal komputer yang dihubungkan dengan media kabel. Secara prinsip, hubungan *peer to peer* ini adalah bahwa setiap komputer dapat berfungsi sebagai server (penyedia layanan) dan *client*, keduanya dapat difungsikan dalam suatu waktu yang bersamaan.



Gambar 2.4 Tipe Jaringan Peer to peer

2.4.3 Jenis Topologi Jaringan Komputer

Topologi jaringan adalah pola node dan interkoneksi antar sesama komputer. Pada suatu jaringan, topologi berarti dengan medium transmisi yang dipakai menentukan tipe data yang bisa dikirimkan dengan kecepatan dan efisiensi komunikasi dalam jaringan.

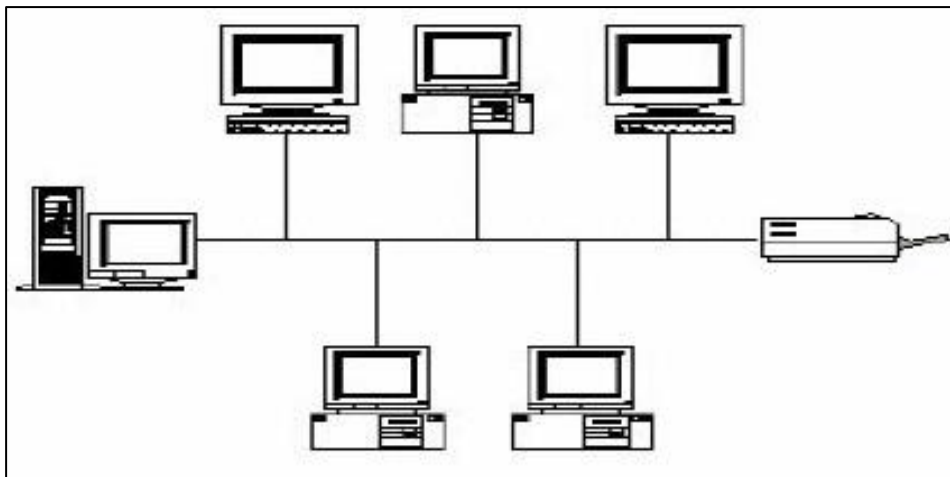
Dulu topologi digunakan untuk menggolongkan local network. Ini hanya berguna jika ada beberapa *local network* yang berbeda dan topologi itu mengidentifikasi cara jaringan itu beroperasi dan interkoneksi dari node.

Topologi-topologi yang umum untuk *local area network* adalah bus, cincin dan bintang.

a. Topologi Bus

Pada topologi bus biasanya digunakan kabel koaksial. Seluruh jaringan biasanya merupakan satu saluran kabel yang kedua ujungnya terterminasi dengan beban 50 ohm, komputer-komputer yang ingin mengaitkan dirinya ke jaringan men- tap kartu ethernet-nya sepanjang kabel.

Jelas bahwa instalasi topologi bus merupakan instalasi paling sederhana dan umumnya membutuhkan biaya yang paling murah dibandingkan jenis-jenis topologi lainnya. Topologi ini banyak digunakan di jaringan LAN yang kecil lima sampai sepuluh komputer saja.



Gambar 2.5 Topologi Bus

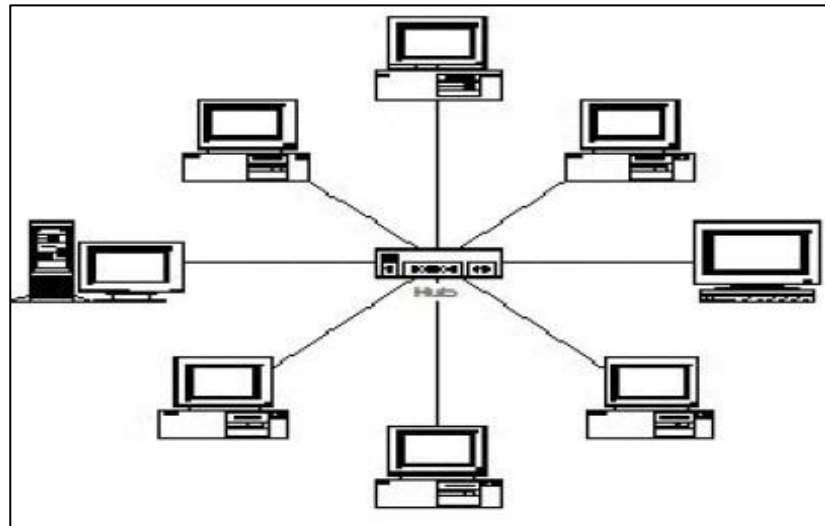
b. Topologi Bintang (*Star*)

Dalam topologi star sebuah terminal pusat bertindak sebagai pengatur dan pengendali semua komunikasi data yang terjadi. Topologi untuk instalasi umum yang dirancang di sekitar sistem komputer mainframe sentral adalah jaringan bintang.

Sekarang piranti pemroses sentral ini dapat berupa apa saja dari PC besar sampai mainframe, tergantung pada daya yang diperlukan, karena *server* dapat melakukan pemrosesan dan pensaklaran pesan dari satu saluran yang datang ke saluran lainnya.

Jaringan yang lebih rumit dapat dibangun dengan menginterkoneksi server dengan client. Salah satu ciri topologi jenis ini adalah tiap piranti dasar dihubungkan ke sistem sentral dengan hubungan titik ke titik untuk penggunaanya yang eksklusif atau untuk dipakai bersama oleh sejumlah kecil lainnya.

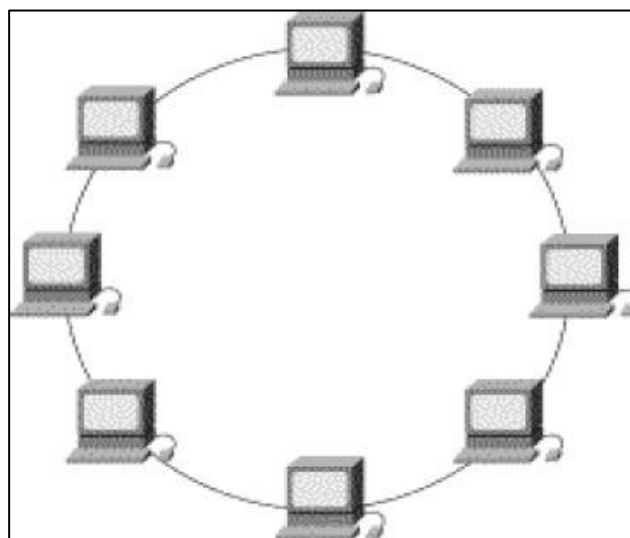
Hub pemrosesan sentral (*server*) biasanya menggunakan circuit switching untuk membentuk jalur khusus dari suatu workstation ke dalam dirinya sendiri atau ke workstation lainnya.



Gambar 2.6 Topologi Star

c. Topologi Cincin(*Ring*)

Pada topologi ring (cincin) mirip dengan topologi bus tetapi kedua terminal yang berada diujung saling dihubungkan, sehingga menyerupai seperti lingkaran. Karakteristik *local area network* cincin adalah cincin terdiri dari sejumlah *repeater*.



Gambar 2.7 Topologi Ring

2.4.4 Perangkat Jaringan

Dalam menyusun sebuah jaringan komputer, peran perangkat jaringan komputer sangatlah penting. Perangkat jaringan adalah semua komputer, *peripheral*, *interface card*, dan perangkat tambahan yang terhubung ke dalam suatu sistem jaringan komputer untuk melakukan komunikasi data. Adapun perangkat yang umum dipakai pada pembangunan jaringan komputer terdiri dari :

a. Server

Server adalah sebuah sistem komputer yang menyediakan jenis layanan (*service*) tertentu dalam sebuah jaringan komputer. *Server* didukung dengan *prosesor* yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan (*network operating system*). *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (*printer*), dan memberikan akses kepada *workstation* anggota jaringan.



Gambar 2.8 Server

b. Workstation

Workstation adalah komputer yang terhubung dengan sebuah *local-area network* (LAN) juga digunakan untuk aplikasi teknik (CAD/CAM), desktop *publishing*, pengembangan *software*, dan

aplikasi lainnya yang membutuhkan tingkat komputasi dan kemampuan grafis yang cukup tinggi.



Gambar 2.9 Workstations (Komputer Desktop)

c. Router

Router merupakan perangkat keras jaringan komputer yang dapat digunakan untuk menghubungkan beberapa jaringan yang sama atau berbeda. *Router* adalah sebuah alat untuk mengirimkan paket data melalui jaringan atau internet untuk dapat menuju tujuannya, proses tersebut dinamakan *routing*.



Gambar 2.10 Router

d. Switch

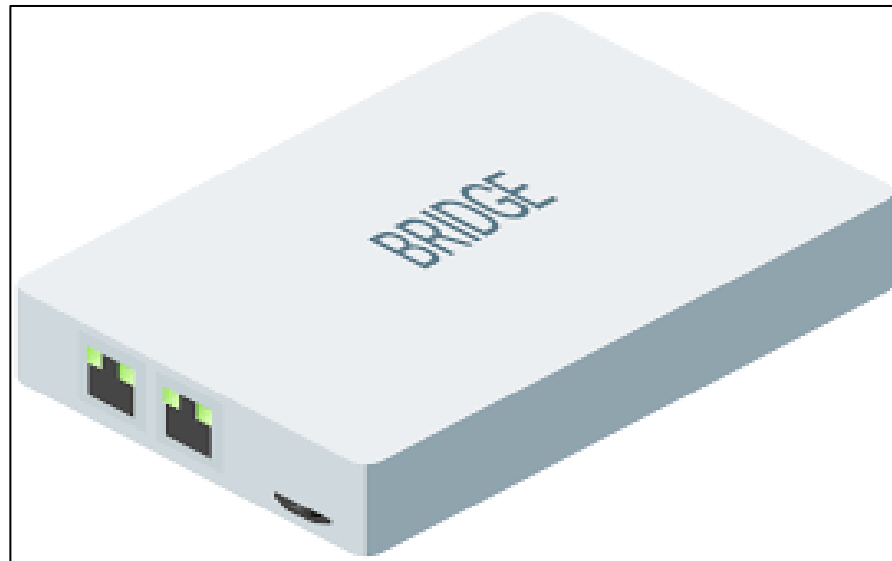
Switch adalah pengalih jaringan atau sebuah alat yang menjalankan penghubung tidak terlihat penghubung penyekat (segmentation) dari banyak jaringan dengan mengalihkan dengan melihat alamat MAC. Switch pada jaringan dapat dipakai untuk menghubungkan komputer atau penghala pada sebuah area yang terbatas, Switch juga bekerja di lapisan data terhubung (data link).



Gambar 2.11 Switch

e. Bridge

Bridge adalah suatu alat yang dapat menghubungkan jaringan komputer LAN “*Local arean Network*” dengan jaringan LAN yang lain. *Bridge* dapat menghubungkan tipe jaringan komputer berbeda-beda “misalnya seperti *Ethernet & Fast Ethernet*” ataupun tipe jaringan yang serupa atau sama. Alat ini bekerja pada data Link layer model OSI “*Open System Interconnection*” karena itu bridge bisa menyambungkan jaringan komputer yang memakai metode transmisi atau medium *access control* yang tidak sama atau berbeda. *Bridge* juga ialah alat yang bisa mempelajari alamat link yang ada pada setiap perangkat yang terhubung dengannya dan juga mengatur alur *frame* berdasarkan alamat tersebut.



Gambar 2.12 *Bridge*

f. Kabel Jaringan

Kabel jaringan adalah perangkat keras yang berbentuk kabel dan dirancang khusus untuk memenuhi dalam koneksi jaringan. Secara khusus hanya dipergunakan untuk jaringan, kabel jaringan digunakan bisa untuk menghubungkan perangkat jaringan ke perangkat lain atau 2 lebih komputer dengan berbagi daya. Fungsi utama dari kabel jaringan adalah menghubungkan satu perangkat dengan perangkat lainnya. Dalam hal ini biasanya sebagai penghubung *server* dan *user/client*. Dengan terhubung melalui kabel jaringan maka akan terbentuk berbagai macam topologi jaringan. Tentu saja untuk digunakan kabel jaringan ini harus didukung dan dilengkapi dengan *hardware* lainnya.



Gambar 2.13 Kabel Jaringan (*twisted pair*)

2.5 TCP/IP

TCP/IP (*Transmission Control Protocol / Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data komputer di Internet (Onno W. Purbo, 1999) Komputer-komputer yang terhubung ke Internet berkomunikasi dengan protokol ini..Sebuah komputer yang menggunakan protocol TCP/IP dan terhubung langsung ke Internet, maka komputer dapat berhubungan dengan komputer di belahan dunia mana pun yang juga terhubung ke Internet. Pada dasarnya komunikasi data merupakan proses pengiriman data dari satu komputer ke komputer yang lain. Untuk mengirimkan data, pada komputer harus ditambahkan alat khusus, yang dikenal sebagai *network interface* (interface jaringan). TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Protokol yang satu tidak perlu mengetahui cara kerja protokol lain, sepanjang masih bisa saling mengirim dan menerima data. Sekumpulan protokol TCP/IP ini dimodelkan dengan empat layer, keempat lapis/layer tersebut adalah:

1. *Network Interface Layer* (Ethernet, PPP,SLIP)
2. *Interface Layer* (IP, ICMP,ARP)
3. *Transport Layer* (TCP,UDP)
4. *Application Layer* (SMTP, FTP,HTTP,dll)

2.5.1 Protokol-Protokol TCP/IP

1. TCP(*Transmission Control Protocol*)

TCP (*Transmission Control Protocol*) berfungsi untuk mengubah suatu blok data yang besar menjadi segmen-segmen yang dinomori dan disusun secara berurutan agar si penerima dapat menyusun kembali segmen-segmen tersebut seperti waktu pengiriman. Sebuah koneksi TCP dikenal sebagai koneksi yang bersifat *connection oriented* yang memberikan layanan secara bergaransi.

2. UDP (*User Datagram Protocol*)

Berbeda dengan TCP, koneksi UDP bersifat *connection-less*. 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 ini akan lebih efisien pada koneksi broadcasting atau multicasting.

3. ICMP(*Internet Control Message Protocol*)

ICMP (*Internet Control Message Protocol*) adalah protokol yang berguna melaporkan jika terdapat suatu masalah dalam pengiriman data. Fungsinya adalah memberitahukan jika ada yang tidak sampai ketujuan dan memberitahukan jika memori di router penuh.

Penomoran Port dari TCP, UDP dan ICMP yang digunakan :

Tabel 2.1 Nomor-nomor TCP,UDP,ICMP

Service	Protokol	Port
FTP	TCP	21
FTP-Data	TCP	20
SSH	TCP	22
TELNET	TCP	23
SMTP	TCP	25
DNS	TCP/UDP	53
HTTP	TCP	80
POP3	TCP	110
IMAP	TCP	143
HTTPS	TCP	443
Webmin	TCP	10000
Ping	ICMP	8

2.5.2 IP Address

IP address adalah alamat yang diberikan ke jaringan dan peralatan jaringan yang menggunakan protokol TCP/IP. IP address terdiri dari 32 bit angka biner yang dapat ditulis sebagai empat angka desimal yang dipisahkan oleh tanda titik.

a. Pembagian Kelas IP

Alamat IP dibagi menjadi kelas-kelas yang masing-masing mempunyai kapasitas jumlah IP yang berbeda-beda.

Tabel 2.2 Default Subnet Pada TCP/IP

Kelas	Network ID	Host ID	Default Subnet Mask
A	W	X.Y.Z	255.0.0.0
B	W.X	Y.Z	255.255.0.0
C	W.X.Y	Z	255.255.255.0

IP address terdiri atas dua bagian yaitu network ID dan host ID, dimana network ID menentukan alamat jaringan komputer, sedangkan host ID menentukan alamat *host* (komputer, *router*, *switch*).

b. Subneting

Subneting adalah pembagian suatu kelompok alamat IP menjadi bagian-bagian yang lebih kecil lagi . Tujuan dalam melakukan subnetting ini adalah:

- a. Membagi suatu kelas jaringan menjadi bagian-bagian yang lebihkecil.
- b. Menempatkan suatu *host*, apakah berada dalam satu jaringan atautidak.
- c. Untuk memperbanyak Network ID pada satu kelas alamat IP. Kelas A, B dan C yang digunakan oleh publik memiliki nilai subnet bawaan (*default*). Kelas D dan E dialokasikan khusus untuk penggunaan multicasting daneksperimental.

2.6 Keamanan Jaringan Komputer

Sistem keamanan jaringan komputer yang terhubung ke Internet harus direncanakan dan dipahami dengan baik agar dapat melindungi investasi dan sumber daya di dalam jaringan komputer tersebut secara efektif. Sebelum mulai mengamankan suatu jaringan, harus ditentukan terlebih dahulu tingkat ancaman/serangan yang harus diatasi, dan resiko yang harus diambil maupun yang harus dihindari.

2.7 Firewall

Menurut Maiwald (2005) *Firewall* adalah sebuah perangkat/*software* di dalam jaringan yang dapat melakukan pemantauan lalu lintas jaringan, membuat

pemisah antara jaringan yang terpercaya dan tidak terpercaya. *Firewall* menolak semua lalu lintas yang tidak terpercaya agar jaringan menjadi aman dari serangan dan mengizinkan lalu lintas yang terpercaya untuk masuk ke dalam jaringan. *Firewall* merupakan garis pertahanan pertama dalam melindungi jaringan dan data-data yang ada di dalamnya.

2.7.1 Tugas Firewall

1. Pertama dan yang terpenting adalah harus dapat mengimplementasikan kebijakan *security* di jaringan. Jika aksi tertentu tidak diperbolehkan oleh kebijakan ini, maka *firewall* harus meyakinkan bahwa semua usaha yang mewakili operasi tersebut harus gagal atau digagalkan.
2. Melakukan *filtering* mewajibkan semua trafik yang ada untuk dilewatkan melalui firewall bagi semua proses pemberian dan pemanfaatan layanan informasi. Dalam konteks ini, aliran paket data dari/menjuhu firewall, diseleksi berdasarkan IP- *address*, nomor port, atau arahnya, dan disesuaikan dengan kebijakan *security*.
3. Firewall juga harus dapat merekam/mencatat *even-even* mencurigakan serta memberitahu *administrator* terhadap segala usaha-usaha menembus kebijakan *security*.

Dalam mengontrol lalu lintas keluar masuk data dalam jaringan, *firewall* menggunakan beberapa metode berikut ini :

1. Paket *Filtering*

Paket data diperiksa menggunakan seperangkat aturan penyaringan. Paket yang tidak lulus penyaringan akan dihapus.

2. *Proxy service*

Data yang berasal dari internet diterima oleh *firewall* kemudian diteruskan kepada komputer yang memintanya, begitu sebaliknya.

3. *Stateful Inspection*

Merupakan metode membandingkan bagian kunci dari paket data tersebut dengan *database* data-data terpercaya. Data yang keluar dari

firewall ditandai dengan ciri-ciri khusus. Selanjutnya, data yang masuk akan dibandingkan dengan ciri-ciri khusus tersebut. Jika dalam proses perbandingan terdapat kecocokan, data diizinkan masuk. Jika terdapat ketidakcocokan, *firewall* akan menghapus data tersebut.

Selain itu, administrator dapat menambah dan mengurangi saringan yang dapat digunakan pada *firewall*, antara lain melalui :

a. Alamat IP

Setiap alat yang terhubung dengan internet memiliki IP. Jika sebuah alamat IP terlalu banyak mengirimkan paket data yang mencurigakan, *firewall* dapat memblokir semua paket dari alamat IP tersebut.

b. Nama Domain

Semua server di internet memiliki nama domain untuk memudahkan kita menghapalnya daripada harus menggunakan sederetan alamat IP. *Firewall* dapat memblokir semua akses menuju nama domain tertentu, atau hanya mengizinkan akses kepada beberapa nama domain tertentu.

c. Protokol

Dalam berkomunikasi, sesama komputer menggunakan jalur-jalur tertentu. Jalur-jalur komunikasi tersebut dikenal dengan nama protokol. *Firewall* dapat menyaring lalu lintas data pada beberapa protokol yang sering digunakan seperti HTTP, FTP, UDP, ICMP, SNMP, dll.

d. Port

Setiap server menggunakan port-port bernomor untuk menyediakan layanan di internet. Satu port untuk satu layanan.

e. Kata-kata atau frasetertentu

Firewall dapat memblokir semua data keluar dan data masuk yang mengandung kata atau frase tertentu dengan cara mengendus (*sniff*) setiap paket data dan mencocokkan setiap kata atau frase yang ada didalamnya dengan daftar *black list*.

2.7.2 Layanan Firewall

Firewall secara umum untuk melayani :

1. Mesin/Komputer

Setiap mesin/komputer yang terhubung langsung ke jaringan luar atau internet dan menginginkan semua yang terdapat pada komputernya terlindungi.

2. Jaringan

Jaringan komputer yang terdiri lebih dari satu buah komputer dan berbagai jenis topologi jaringan yang digunakan, baik yang dimiliki oleh perusahaan, organisasi dsb.

2.7.3 Bentuk Firewall

Adapun bentuk *firewall* secara umum terdapat 2 macam yakni :

a. *Firewall Hardware*

Berupa sebuah piranti keras yang sudah dilengkapi dengan perangkat lunak tertentu, sehingga tinggal melakukan konfigurasi dari *firewall* tersebut saja.

b. *Firewall Software*

Firewall juga dapat berupa piranti lunak yang ditambahkan kepada sebuah *server*, yang dikonfigurasi menjadi *firewall*.

2.7.4 Karakteristik Firewall

Karakteristik dari *firewall* yang mempunyai tujuan sebagai berikut :

1. Segala lalu lintas jaringan baik dari dalam maupun dari luar harus melewati *firewall*.
2. Kebijakan keamanan, hanya memberikan izin untuk masuk ke server atau jaringan komputer yang memenuhi syarat tertentu, tentu dalam hal ini bisa jadi berbagai jenis firewall yang bisa digunakan untuk suatu kebijakan.
3. *Firewall* itu sendiri haruslah kebal atau relatif kuat terhadap serangan/kelemahan. Hal ini berarti penggunaan sistem yang dapat dipercaya dan dengan sistem yang relatif aman.

2.7.5 Hal-hal yang tidak bisa diatasi oleh Firewall

Berikut adalah hal-hal yang tidak bisa diatasi *firewall* :

1. Serangan yang dilakukan dengan mem *bypass firewall*.
2. *Firewall* tidak bisa melindungi serangan yang dilakukan di dalam sistem(*internal*)
3. *Firewal* tidak bisa melindungi pengiriman virus yang disisipkan ke suatu *program* atau *file* karena *firewall* tidak bisa melakukan *scanning* terhadap *file*, *program*, *email* yang berisi virus.

2.7.6 Keuntungan Firewall

Firewall dapat mengerjakan banyak hal untuk jaringan dan *server*. Pada prakteknya, beberapa keuntungan utama yang dapat diperoleh dengan adanya *firewall* adalah sebagai berikut:

1. *Firewall* sebagai fokus keputusan *security*
2. *Firewall* mendukung *security policy*
3. *Firewall* mencatat log aktifitas internet.

2.7.7 Kebijakan-kebijakan Firewall

1. *Pass*

Dengan opsi ini setiap paket akan langsung diterima oleh *firewall* dan diteruskan kepada tujuan dari paket tersebut. Misalnya paket tersebut menuju *server* kita dengan tujuan port 80 maka paket tersebut akan langsung diteruskan untuk diproses oleh *server*.

2. *Block*

Berbeda dengan *pass*, setiap paket yang memiliki kata *block* ini akan ditolak, tapi *firewall* akan mengirimkan pesan ICMP *error* kepada si pengirim paket. Secara default, *firewall* akan mengirimkan pesan ICMP berupa port-unreachable.

3. *Redirect*

Redirect bertugas untuk mengubah tujuan dari paket ke mesin *firewall* itu sendiri dan juga mengalihkan sebuah *website* terlarang menuju ke alamat IP atau *website* yang telah ditentukan.

2.8 IPFW

IPFIREWALL (IPFW)) adalah *software* aplikasi bawaan dari freebsd yang di kembangkan oleh para staff pengembang freebsd sendiri. Dalam instalasi standart sistem operasi freebsd, IPFW sudah terpasang. Kita tinggal menambahkan statement *firewall_enable="YES"* pada file */etc/rc.conf* untuk mengaktifkan *firewall*. Namun ada baiknya perlu di kompile ulang kernel agar fungsi nat dapat berjalan dengan baik dan firewall *default accept*.

2.8.1 Perintah IPFW

Perintah *ipfw* merupakan sebuah sarana untuk membuat rule tunggal seperti menambah atau menghapus ke aturan internal disaat firewall sedang dijalankan. Yang menjadi masalah adalah ketika sistem *shutdown* atau *reboot* maka semua aturan yang ditambahkan akan hilang.

- a. *Ipfw list*: untuk menampilkan semua aturan firewall berdasarkan urutan.
- b. *Ipfw -t list*: Untuk menampilkan semua aturan firewall dengan penanda waktu terakhir aturan tersebut digunakan.
- c. *Ipfw-d list*: Menampilkan aturan dinamik yang ditambahkan ke aturan static.
- d. *Ipfw -d -e list*: Untuk menampilkan aturan dinamik yang telah berakhir.
- e. *Ipfw zero*: Untuk melakukan reset login dengan melakukan reset counter.
- f. *Ipfw zero NUM*: Untuk melakukan reset counter hanya untuk nomor tertentu saja.

2.8.2 Aturan-aturan IPFW

Dalam penulisan aturan *firewall* terdapat tanda # yang berarti komentar akan fungsi aturan yang ditulis. Sintaks umum dari penulisan aturan pada IPFW adalah sebagai berikut dibawah ini:

CMD RULE_NUMBER ACTIONS LOGGING SELECTION
--

a. CMD

Setiap aturan yang harus harus menambahkan kata *add*. CMD sebenarnya merupakan gabungan dari kata ”*ipfw -q add*”. Ini dilakukan agar kita tidak harus

menuliskan dua kata atau lebih pada setiap baris aturan, tetapi cukup mewakili kepada satu kata saja.

b. *RULE NUMBER*

Nomor urut aturan tidak harus ada karena akan secara otomatis ditambahkan ketika aturan di muat kedalam tabel internal IPFW. Nomor urut ini untuk memudahkan dalam pemeliharaan sebuah firewall dimana log yang ada dimuat berdasarkan nomor aturan.

c. *ACTIONS*

Sebuah aturan akan berhubungan dengan apa yang harus dilakukan terhadap sebuah paket yang datang. Paket yang ada akan diseleksi berdasarkan aturan yang dibuat. Di dalam rule actions ini terdapat beberapa aturan yang dikenai kepada sebuah paket:

1. *Allow/accept/pass/permit*: Ke empat kata tersebut mempunyai kesamaan yaitu mengizinkan paket lewat.
2. *Check state*: Memeriksa paket yang berlawanan dengan rule set yang ada. Jika ditemukan akan dijalankan tindakan sesuai dengan rule yang telah dibuat. Apabila tidak ditemukan akan berlanjut ke aturan berikutnya.
3. *Deny/drop*: Kedua kata tersebut mempunyai arti yang sama yaitu membuang paket yang ada berdasarkan aturan yang telah ditentukan.

d. *LOGGING*

Log atau logamount yaitu ketika sebuah paket sesuai dengan kata kunci, sebuah pesan akan dimasukkan ke *syslog* dengan sebuah fasilitas dengan nama *SECURITY*. Pesan ini hanya terjadi jika nomor paket khususnya aturan tidak melebihi parameter *logamount*. Jika tidak ada logamount yang dispesifikasikan batas tersebut akan diambil dari variabel *sysctl net.inet.ip.fw.verbose_limit*.

Pencatatan dilakukan setelah semua kondisi paket selesai dibuktikan dan sebelum tindakan terakhir (*accept, deny*) pada paket tersebut. Ketika sebuah aturan dicatat, terdapat beberapa informasi yang akan disimpan. Antara lain :

- tanggal dan waktu
- *rule number*

- aksi yang dilakukan
- alamat IP sumber dan tujuan
- nomor *port* asal dan tujuan
- arah aliran
- *interface* yang dilalui

Sebuah contoh *log* yang diambil dari */var/log/security* adalah sebagai berikut:

```
Jan 20 10:57:48: ipfw: 0012 Deny TCP \
172.21.202.1:62307 192.168.0.1:23 in via rl0
```

e. SELECTION

Kata kunci pada bagian ini digunakan untuk menguraikan atribut yang menyangkut paket yang ada apakah sesuai dengan aturan yang berhubungan dengan paket tersebut. Atribut yang biasa digunakan dalam bagian ini adalah :

1. *Protocol* : *udp* | *tcp* | *icmp* atau nama *protokol* lain yang ditemukan dalam */etc/protocols* dikenali dan mungkin dapat digunakan. Ini merupakan sebuah kewajiban dalam mencantumkan nama *protokol*.

2. *From src to dst*

Kata ini berhubungan dengan alamat IP *address*. Sebuah *rule* atau aturan yang ada harus mencantumkan kedua parameter alamat sumber dan alamat tujuan sebuah IP address atau juga dapat menggunakan sebuah kata kunci khusus seperti *from any to any* atau *from any to me* atau *from 0.0.0.0/0 to any*. Spesifikasi umum sebuah alamat IP adalah nomor *ipaddress/netmask*. Dapat juga berupa sebuah alamat ip saja. Ini merupakan sesuatu yang wajib dalam penulisan aturan.

3. *Port number*

Untuk *protokol* yang mendukung nomor port seperti TCP dan UDP juga dicantumkan nomor *port* dalam aturan ini.

4. *In/out*

Kata *in* dan *out* ini merupakan bagian dari aturan yang menunjukkan paket yang masuk atau keluar.

5. *Via IF*

IF merupakan nama *interface card*. Kata ini berhubungan dengan aturan *in* dan *out* yang ada diatas.

6. *Setup*

Kata ini digunakan untuk mengidentifikasi permintaan paket TCP.

7. *Keep-state*

Ini merupakan sebuah kata kunci dimana setiap pencocokan *firewall* akan membuat suatu aturan dinamis yang secara default adalah mencocokkan *bidirectional* trafik antara sumber dan tujuan *port* yang menggunakan protokol yang sama.

8. `limit{src-addr|src-port|dst-addr|dst-port}`

Firewall hanya mengijinkan n koneksi dengan parameter yang sama yang ditetapkan dalam aturan. Satu atau lebih alamat dan port tujuan akan ditentukan. *Limit* dan *keep-state* tidak dapat digunakan pada aturan ysama. *Limit* menyediakan fungsi *stateful* yang sama sebagai *keep-state* ditambah fungsi itu sendiri.

Dalam aturan IPFW juga terdapat beberapa perintah lainnya seperti tambah rule baru dengan menggunakan perintah *add*, menghapus *rule* secara satu persatu atau dalam *grup* dengan perintah *delete*, menghapus rule keseluruhan dengan perintah *flush*.

2.9 VMWare

Menurut Athailah (2016) *VMWare* adalah *software* virtualisasi yang biasa digunakan untuk membuat virtualisasi *server*, komputer, *system* operasi, *storage device*, aplikasi, jaringan dll. Keuntungannya sendiri mudah digunakan, *fitur unity* berbeda dengan fitur yang lain, tidak perlu *restart* PC untuk beralih sistem operasi dan dapat mengembangkan perangkat dengan cepat karena adanya lebih dari satu sistem operasi yang berjalan bersamaan.












Gambar 2.14 Logo VMWare

2.10 Flowchart

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah.

Tabel 2.3. Simbol pada *Flowchart*

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inialisasi/ pemberian harga awal
	PROSES	Proses perhitungan/ proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/ proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda