PERANCANGAN *FIREWALL* MENGGUNAKAN *SHOREWALL* UNTUK KEAMANAN JARINGAN KOMPUTER PADA PT. INDOFARMA GLOBAL MEDIKA

SKRIPSI

Diajukan Untuk Menempuh Ujian Akhir Sarjana
Program Strata Satu Pada Program Studi
Teknik Komputer

OLEH

ROLAN EFFENDI

1514030174

PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS TEKNIK
UNIVERSITAS SERAMBI MEKKAH
BANDA ACEH

2020

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT yang dengan rahmat dan kasih sayangnya telah memberikan kekuatan dan kesehatan sehingga penulis telah dapat menyelesaikan skripsi ini.

Selawat dan salam tak lupa pula penulis sanjungkan kepangkuan Nabi besar Muhammad SAW, keluarga beserta para sahabatnya, berkat jasa beliaulah kita dapat menikmati indahnya hidup di alam yang disinari kilauan cahaya ilmu pengetahuan dibawah panji agama Allah SWT.

Selanjutnya pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan dalam penyelesaian skripsi ini terutama sekali kepada :

1. Ibu Dr. Irhamni ST, MT, selaku Dekan Fakultas Teknik Universitas Serambi Mekkah.
2. Bapak Zulfan, ST, MT, selaku Ketua Prodi Teknik Komputer Universitas Serambi Mekkah.
3. Ibu Yeni Yanti, ST, MT, selaku Dosen Pembimbing I.
4. Bapak Zahrul Maizi, S.SI, MT, selaku Dosen Pembimbing II.

Semua masukkan dan bantuan yang telah di berikan kepada penulis semoga amal baiknya mendapat pahala diisi Allah SWT. Penulis menyadari dalam penulisan skripsi ini masih banyak sekali terdapat kekurangan dan kekhilafan, oleh karena itu penulis mengharapkan saran dan kritikkan yang bersifat membangun untuk kesempurnaan penulisan skripsi ini pada masa yang akan datang.

Semoga amal baik yang telah di berikan kepada penulis mendapat pahala yang setimpal di berikan Allah SWT. Harapan penulis semoga skripsi ini bermanfaat bagi kita semua. Amin yarabbaalamin.

Banda Aceh, 12 Februari 2020

Penulis,

DAFTAR ISI

Halaman

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

ABSTRAK

Kemajuan teknologi khususnya jaringan komputer sangat membantu di bidang informasi dan pengolahan data informasi. Komputer memegang peranan penting dalam teknologi informasi, melalui komputer akan didapatkan informasi yang diinginkan tanpa keterbatasan ruang dan waktu dengan menggunakan secara maksimal, sistem komputer dalam jaringan komputer yang terintegrasi akan didapatkan informasi data dengan cepat dan tepat. PT. Indofarma Global Medika (PT. IGM) memiliki jaringan distribusi dan juga memiliki informasi-informasi atau integritas data yang penting untuk di lindungi. Untuk mencegah orang lain dalam mengakses informasi atau integritas data tersebut di perlukan suatu keamanan jaringan komputer. Pada penelitian ini penulis akan membangun sebuah *firewall* menggunakan *shorewall* yang terdapat pada komputer *server*, dimana *shorewall* ini adalah salah satu jenis *firewall* di *platform Operating System Linux* yang berbasis *iptables*, guna untuk menjaga informasi atau integritas data yang ada dengan cara membatasi izin akses jaringan yang masuk pada komputer *server*. Sistem operasi yang di gunakan adalah *Linux Debian* 10 *Server*. Metode yang di gunakan dalam pengamanan jaringan komputer pada *shorewall* ini adalah *ACCEPT, DROP* dan *REJECT*.

Kata Kunci : Jaringan Komputer, Keamanan Jaringan, *Linux*, *Firewall*, *Shorewall*

BAB I
PENDAHULUAN

1.1. Latar Belakang

Keamanan jaringan saat ini menjadi isu yang sangat penting dan terus berkembang. Perkembangan teknologi komputer, selain menimbulkan banyak manfaat juga memiliki banyak sisi buruk. Salah satunya adalah serangan terhadap sistem komputer yang terhubung ke *Internet*. Sebagai akibat dari serangan itu, banyak sistem komputer atau jaringan yang terganggu bahkan menjadi rusak. Untuk menanggulangi hal tersebut, diperlukan sistem keamanan yang dapat menanggulangi dan mencegah kegiatan-kegiatan yang mungkin menyerang sistem jaringan kita.

PT. Indofarma Global Medika (PT. IGM) merupakan anak perusahaan PT. Indofarma (Persero), merupakan bagian dari perusahaan BUMN yang bergerak di bidang farmasi, alat kesehatan, dan makanan sehat. Seiring dengan meningkatnya kebutuhan masyarakat terhadap pelayanan kesehatan dan seiring dengan era teknologi dalam bidang kesehatan di Indonesia yang semakin berkembang, PT. IGM yang didukung oleh Tim Sales dan Marketing yang profesional, IT dan teknologi yang mengikuti perkembangan zaman, serta didukung adanya jaringan distribusi di seluruh Indonesia yang dimiliki, PT. IGM siap menjadi Partner Handal di Industri Kesehatan (*Reliable Partner in Healthcare Industry*).

Dengan semakin besarnya jaringan distribusi yang di buat oleh administrator jaringan PT. IGM, maka keamanan jaringan PT. IGM menjadi prioritas penting bagi administrator. Salah satu metode yang baik untuk keamanan jaringan adalah membuat sebuah *firewall*.

Firewall merupakan suatu mekanisme untuk melindungi keamanan jaringan komputer dengan menyaring paket data yang keluar dan masuk di jaringan. Paket data yang baik diperbolehkan untuk melewati jaringan dan paket data yang dianggap jahat tidak diperbolehkan melewati jaringan. *Firewall* adalah salah satu aplikasi *Linux* yang dibutuhkan untuk menjaga integritas data yang ada dari serangan-serangan *hacker* yang tidak bertanggung jawab dengan melakukan *filterisasi* terhadap paket-paket yang datang kepadanya. Sistem operasi *Linux*

telah menyiapkan aplikasi untuk dijadikan sebuah *firewall* diantaranya *ipchains*, *iptables* dan *Shorewall*.

Dalam penelitian ini akan di gunakan aplikasi *Shorewall* dari *Linux Debian Desktop*. *Shorewall* (*Shoreline Firewall*) merupakan salah satu *firewall* yang handal dan murah untuk digunakan di sistem operasi *Linux* selain *Ipchains* dan *Iptables*, *shorewall* juga mudah dikonfigurasi. *Shorewall* yang berbasiskan kepada *iptables* untuk dipermudah dalam penggunaanya, *Shorewall* dapat di installasi pada kebanyakan sistem pada *Linux*. Selain karena *shorewall* perangkat bawaan *linux*, *shorewall* digunakan untuk mengatur paket data masuk dan keluar serta kebijakan-kebijakan yang perlu dilakukan dalam pengamanan komputer yang terhubung ke jaringan.

Dengan latar belakang inilah maka penulis memilih penulisan skripsi ini dengan judul "Perancangan Firewall Menggunakan Shorewall Untuk Keamanan Jaringan Komputer Pada PT. Indofarma Global Medika".

1.2. Batasan Masalah

Berdasarkan latar belakang penelitian maka batasan masalahnya adalah :

1. Penelitian ini hanya membahas tentang penggunaan dan pemanfaatan *firewall* dengan *shorewall* untuk keamanan jaringan komputer pada PT. Indofarma Global Medika berbasis *open source*.
2. Pengujian penggunaaan *firewall* menggunakan *shorewall* pada *server Linux Debian* 10 hanya dilakukan dalam ruang lingkup jaringan komputer *LAN (Local Area Network)*.

1.3. Rumusan Masalah

Berdasarkan latar belakang yang telah di jelaskan, maka dapat di ambil suatu rumusan masalah yaitu :

1. Bagaimana merancang *firewall* untuk keamanan jaringan komputer menggunakan *shorewall*.
2. Bagaimana mengimplementasikan dan mengkonfigurasi *firewall* menggunakan *shorewall* untuk keamanan jaringan komputer.

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah merancang dan mengimplementasikan *firewall* untuk sistem keamanan jaringan komputer menggunakan *shorewall* pada PT. Indofarma Global Medika.

1.5. Manfaat Penelitian

Berdasarkan tujuan penelitian maka manfaat penelitian ini adalah :

1. Sebagai media edukasi pemanfaatan sistem operasi *Linux* yang bersifat *free, open source*, dan legal yang dapat di implementasikan pada tempat studi bahkan kampus.
2. Dengan adanya penelitian merancang *firewall* untuk sistem keamanan jaringan komputer menggunakan *shorewall* ini di harapkan mampu memberikan keamanan yang handal dan mumpuni pada jaringan komputer PT. Indofarma Global Medika.

BAB II

TINJAUAN PUSTAKA

2.1. Firewall

Firewall adalah sebuah sistem atau kelompok sistem yang menerapkan sebuah *access control policy* terhadap lalu lintas jaringan yang melewati titik-titik akses dalam jaringan. Tugas *firewall* adalah untuk memastikan bahwa tidak ada tambahan diluar ruang lingkup yang diizinkan. *Firewall* bertanggung jawab untuk memastikan bahwa *access control policy* yang diikuti oleh semua pengguna di dalam jaringan tersebut. *Firewall* sama seperti alat-alat jaringan lain dalam hal untuk mengontrol aliran lalu lintas jaringan. Namun, tidak seperti alat-alat jaringan lain, sebuah *firewall* harus mengontrol lalu lintas *network* dengan memasukkan faktor pertimbangan bahwa tidak semua paket-paket data yang dilihatnya adalah apa yang seperti terlihat. *Firewall* digunakan untuk mengontrol akses antara *network internal* sebuah organisasi *Internet* Riadi, (2011).

Firewall merupakan 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 atau kegiatan suatu segmen pada jaringan lokal dengan jaringan luar yang bukan merupakan ruang lingkupnya" Aziz, (2012).

Firewall adalah suatu cara atau mekanisme yang diterapkan baik terhadap *hardware*, *software*, ataupun sistem dengan tujuan untuk melindungi. Perlindungan dapat dilakukan dengan menyaring, membatasi, atau bahkan menolak suatu atau semua hubungan/kegiatan dari suatu segmen pada jaringan pribadi dengan jaringan luar yang bukan merupakan ruang lingkupnya. Segmen tersebut dapat merupakan sebuah *workstation, server, router*, atau *Local Area Network*.

Firewall secara umum diperuntukkan 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 dan sebagainya.

Firewall mempunyai beberapa tugas:

1. Mengimplementasikan kebijakan *security* di jaringan (*site security policy*) : jika aksi tertentu tidak diperbolehkan oleh kebijakan ini, maka *firewall* harus meyakinkan bahwa semua usaha yang mewakili operasi tersebut harus gagal atau digagalkan. Dengan demikian, semua akses ilegal antar jaringan (tidak diotorisasikan) akan ditolak.
2. Melakukan *filtering* : mewajibkan semua *traffic* yang ada untuk dilewatkan melalui *firewall* bagi semua proses pemberian dan pemanfaatan layanan informasi. Dalam konteks ini, aliran paket data dari/menuju *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 Pratama, (2014).

Gambar 2.1. Firewall
Sumber : https://www.google.com/amp/s/www.cbronline.com/what-is/what-is-a-firewall-4900896/amp/

2.2. Shorewall

S*horewall* adalah salah satu *tools firewall* pada *linux* yang berbasiskan *iptables*. Dalam *shorewall* terdapat konsep *"zone"* yang memudahkan kita untuk menentukan *policy firewall*, dari pada kita melakukan konfigurasi secara manual dengan *iptables* Dicky,(2018).

Gambar 2.2. Shorewall

Sumber : http://shorewall.org/

2.3. Pengertian Jaringan Komputer

Jaringan komputer merupakan kumpulan dari beberapa komputer yang dihubungkan satu dengan lainnya untuk berbagi informasi dan perangkat yang ada baik perangkat keras ataupun perangkat lunak. Jaringan ini memerlukan media transmisi tertentu untuk dapat saling berbagi informasi, program dan penggunaan bersama perangkat keras Arifin, (2011).

Mengemukakan bahwa jaringan komputer adalah sekelompok komputer otonom yang saling menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi data, informasi, program aplikasi, dan perangkat keras seperti *printer, scanner*, *CD-Drive* ataupun *harddisk*, serta memungkinkan untuk saling berkomunikasi secara elektronik" Herlambang, (2008). Potensi jaringan komputer antara lain:

a. Mengintegrasikan dan berbagai pakai peralatan
b. Komunikasi jaringan komputer memungkinkan terjadinya komunikasi antar pemakai komputer.
c. Perlindungan data dan informasi jaringan komputer dimanfaatkan pula untuk mendistribusikan proses dan aplikasi sehingga dapat mengurangi terjadinya *bottleneck* atau tumpukan pekerjaan pada satu bagian.

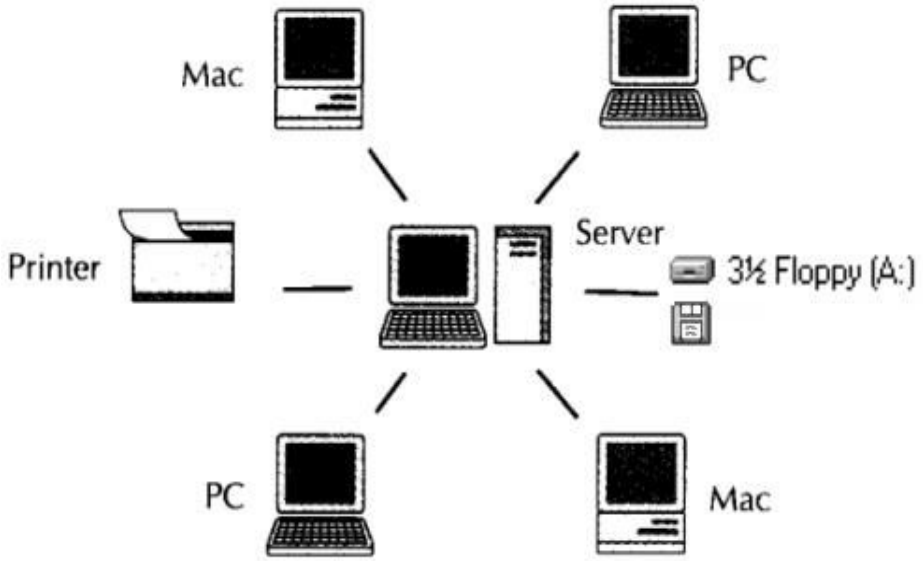
d. Keteraturan aliran infomasi jaringan komputer mampu mengalirkan data-data komputer *client* dengan cepat untuk mengintegrasikan dalam komputer *server*.

Berdasarkan letak geografis jaringan komputer dapat dibagi menjadi tiga jenis yaitu:

a. *Local Area Network* (LAN)
b. *Metropolitan Area Network* (MAN)
c. *Wide Area Network* (WAN)

1. Local Area Network (LAN)

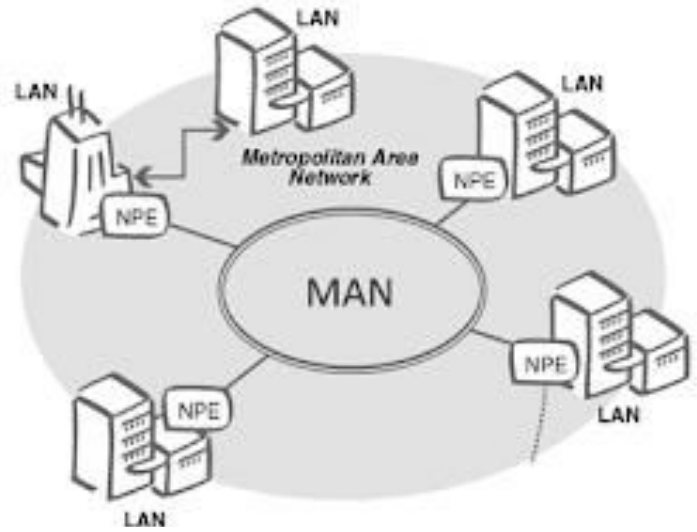
LAN adalah jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan seperti sebuah perkantoran di sebuah gedung atau sebuah sekolah." Secara garis besar terdapat dua tipe jaringan atau LAN, yaitu jaringan *peer to peer* dan jaringan *client-server*. Pada jaringan *peer to peer*, setiap komputer yang terhubung ke jaringan dapat bertindak baik sebagai *client* maupun *server*. Sedang pada jaringan *client-server*, hanya satu komputer yang bertugas sebagai *server* dan komputer lain berperan sebagai *client* Herlambang, (2008).



Gambar 2.3. LAN (Local Area Network)
Sumber : https://www.gueabans.com/2017/11/pengertian-lan-wan-man-dan-internet.html

2. Metropolitan Area Network (MAN)

MAN (*Metropoliton Area Network*) suatu jaringan dalam suatu kota dengan transfer data berkecepatan tinggi, yang menghubungkan berbagai lokasi seperti kampus, perkantoran, pemerintahan, dan sebagainya. Jaringan MAN adalah gabungan dari beberapa LAN. Jangkauan dari MAN ini antara 10 hingga 50 km, MAN ini merupakann jaringan yang tepat *Metropolitan Area Network* atau disingkat dengan MAN (Freeman, 2014).



Gambar 2.4. Metropolitan Area Network
Sumber : https://www.gueabans.com/2017/11/pengertian-lan-wan-man-dan-internet.html

3. Wide Area Network (WAN)

WAN (*Wide Area Network*) adalah suatu jaringan yang digunakan untuk membuat interkoneksi antara jaringa komputer *Local* yang secara fisik tidak berdekatan satu sama lain, yang secara fisik bisa dipisahkan dengan kota, provinsi atau bahkan melintasi batas *geography* – lintas negara dan benua. Ada beberapa teknologi jaringan WAN saat ini yang bis kita gunakan (Ciccarelli, 2016).

Gambar 2.5 Wide Area Network
Sumber : https://www.gueabans.com/2017/11/pengertian-lan-wan-man-dan-internet.html

2.4. Manfaat Jaringan Komputer

Beberapa manfaat yang terdapat pada jaringan komputer sebagai berikut :

1. Pengguna dapat saling berbagi printer dengan kualitas tinggi, dibanding menggunakan printer kualitas rendah dimasing-masing meja kerja. Selain itu, lisensi perangkat lunak jaringan komputer dapat lebih murah dibandingkan lisensi stand-alone terpisah untuk jumlah pengguna sama.
2. Jaringan komputer membantu mempertahankan informasi agar tetap handal dan *up-to-date*. Sistem penyimpanan data terpusat yang dikelola dengan baik memungkinkan banyak pengguna mengakses data dari berbagai lokasi yang berbeda dengan hak akses yang bisa diatur bertingkat.
3. Jaringan Komputer membantu mempercepat proses berbagi data (data *sharing*). *Transfer* data pada jaringan komputer lebih cepat dibandingkan dengan sarana berbagi data lainnya.
4. Jaringan komputer memungkinkan kelompok kerja berkomunikasi dengan lebih efisien. Subtansinya adalah penyampaian pesan secara elektronik

misalnya sistem penjadwalan, pemantauan proyek, konferensi online dan groupware yang bertujuan membantu tim bekerja lebih efektif.

5. Jaringan komputer juga membantu perusahaan dalam melayani pelanggan dengan lebih efektif Sukmaaji, (2008).

2.5. Keamanan Jaringan

Keamanan jaringan komputer sebagai bagian dari sebuah sistem informasi adalah sangat penting untuk menjaga validitas dan integritas data serta menjamin ketersediaan layanan bagi penggunanya. Sistem harus dilindungi dari segala macam serangan dan usaha-usaha penyusupan atau pemindaian oleh pihak yang tidak berhak dimana usaha tersebut bisa dilakukan baik dari dalam maupun dari luar sistem" Aziz, (2012),.

Keamanan jaringan adalah implementasi perangkat keamanan, kebijakan dan proses untuk mencegah akses tanpa izin ke dalam sumber daya jaringan maupun melakukan perubahan atau kerusakan pada sumber daya atau data Bastien, (2004).

Keamanan komputer (computer security) meliputi empat aspek yaitu *privacy and confidentiality*, *integrity*, *authentication* dan *availability* Garfinkel, (2003).

1. Privacy and Confidentiality

Penjelasan dari aspek ini adalah usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. *Privacy* lebih ke arah data-data yang sifatnya privat sedangkan *confidentiality* berhubungan dengan data yang diberikan ke pihak lain untuk keperluan tertentu.

2. Integrity

Aspek ini menekankan bahwa informasi tidak boleh diubah tanpa seizin pemilik informasi tersebut. Contohnya adalah sebuah *email* dapat saja ditangkap (*intercept*) di tengah jalan, diubah isinya, kemudian diteruskan ke alamat yang dituju. Penanggulangannya adalah dengan menggunakan *enkripsi* dan digital *signature*.

3. Authentication

Aspek ini berhubungan dengan metode untuk menyatakan bahwa informasi benar-benar asli, orang yang mengakses atau memberikan informasi adalah benar-benar orang yang dimaksud, atau *server* yang dihubungi benar-benar *server* yang asli.

4. Availability

Aspek ini berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang dapat menghambat atau meniadakan akses ke informasi

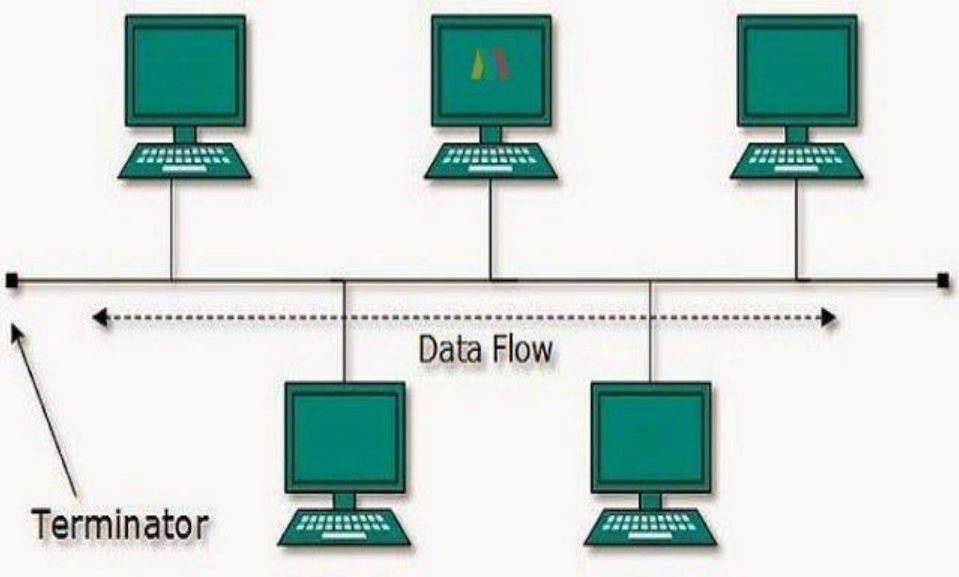
2.6. Topologi Jaringan

Topologi atau arsitektur jaringan merupakan pola hubungan antar terminal dalam suatu sistem jaringan komputer". *Topologi* jaringan adalah istilah yang digunakan untuk menguraikan cara bagaimana komputer terhubung dalam suatu jaringan Herlambang, (2008).

1. Topologi Bus

Topologi bus menggunakan sebuah kabel *backbone* dan semua *host* terhubung secara langung pada kabel tersebut". *Topologi* ini paling banyak dipergunakan pada masa penggunaan kabel *coaxial* menjamur. *Topologi bus* atau linear mempunyai karakteristik sebagai berikut:

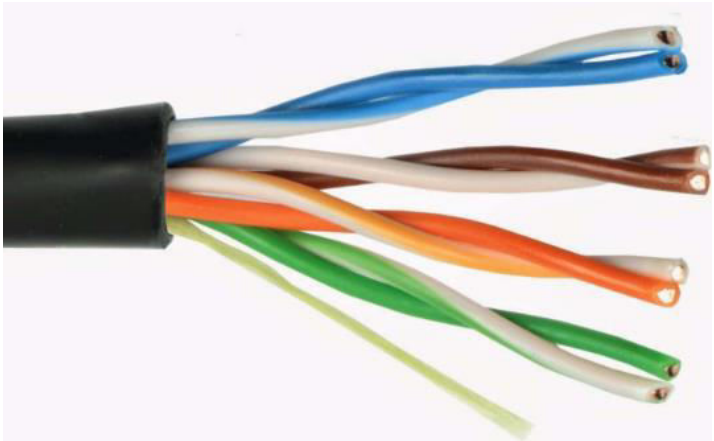
a. Merupakan satu kabel yang kedua ujungnya ditutup dimana sepanjang kabel terdapat *node*.
b. Paling sederhana dalam instalasi.
c. Signal melewati kabel 2 arah sehingga memungkinkan terjadinya *collision*.
d. Masalah terbesar jika salah satu segmen kabel terputus, maka seluruh jaringan akan terhenti.
e. *Topologi bus* adalah jalur transmisi dimana sinyal diterima dan dikirimkan pada setiap alat/*device* yang tersambung pada satu garis lurus (kabel), sinyal hanya akan ditangkap oleh alat yang dituju, sedangkan alat lainnya yang bukan tujuan akan mengabaikan sinyal tersebut Sofana, (2011).



Gambar 2.6. Topologi bus (Sumber : Siti, 2016)
Sumber : https://www.maxmanroe.com/vid/teknologi/komputer/topologi-jaringan.html

2.7. Kabel UTP

UTP (*Unshielded twisted-pair*) adalah sebuah jenis kabel jaringan yang menggunakan bahan dasar tembaga, yang tidak dilengkapi *shield internal*. UTP merupakan jenis kabel yang paling umum digunakan dalam jaringan lokal (LAN) karena harganya yang rendah, fleksibel dan kenerja yang ditunjukannya *relative* bagus Sukmaaji, (2008).



Gambar 2.7. Kabel UTP

Sumber : https://rocketmanajemen.com/pengertian-dan-fungsi-kabel-utp/

2.8. Pengertian Komputer

Menurut Supriyanto, (2007) istilah dari komputer itu sendiri berasal dari bahasa latin camputare yang mengandung arti "menghitung" (*to compute atu reckon*). Defenisi lain misalnya komputer secara umum pernah dipergunakan untuk mengacu pada orang yang melakukan perhitungan aritmatika, dengan atau tanpa mesin pembantu.

Dari uraian diatas maka yang disebut dengan komputer adalah perangkat elektronik yang dapat menerima masukan (*input*), dan selanjutnya melakukan pengolahan (proses) untuk menghasilkan keluaran (*output*) berupa informasi. Bentuk komputer yang dulu cukup besar untuk mengoperasikan sebuah program, sekarang berbentuk kecil dengan kemampuan mengoperasikan program yang beragam.

Perangkat komputer harus bisa difungsikan secara komperehensif (kompak dan bersamasama) dan melaksanakan tugasnya yaitu dalam mengelolah data atau informasi . untuk mewujudkan konsepsi komputer sebagai pengelolah data agar menghasilkan suatu informasi, maka diperlukan sistem komputer (*computer System*) yang elemenya terdiri dari *hardware, software*, dan *brainware*. Ketiga elemen sistem komputer tersebut harus saling berhubungan dan membentuk kesatuan yang saling mendukung untuk bekerja sama. *Hardware* tidak akan berfungsi apabila tanpa *software*, mikian juga sebaliknya. Dan keduanya tidak akan bermanfaat apabila tidak ada manusia (*brainware*) operasikan dan mengendalikanya (sumber : Siswanto, 2013).

A. Perangkat komputer

Sesuai dengan fungsinya, beberapa perangkat komputer yang terdapat dalam suatu jaringan komputer, yaitu :

1. Komputer Server / PC Server

Komputer *server* adalah komputer yang biasanya dikususkan untuk menyimpan data yang akan digunakan bersama, atau berbagai basis data. Selain itu, jika menggunakan sistem operasi berbasis *network* (*Network Operating System*) maka komputer *server* berisi informasi daftar *user* yang diperbolehkan

masuk ke *server* tersebut, berikut otoritasnya yang dapat di-*manage* oleh *administrator* (Sopandi, 2010).

1. Workstation

Workstation adalah komputer yang ditunjukan sebagai *client*, dimana komputer ini sebagai tempat kerja atau pengolahan data yang diakses dari *server* (Sopandi, 2010).

2.9. Pengertian Sistem Operasi

Sistem operasi adalah program terpenting dari program-program yang terdapat dalam sistem komputer. Sistem operasi dapat dianggap sebagai program kontrol yang bertugas untuk menjalankan programprogram lain yang ada di dalam komputer. Dalam hal ini sistem operasi berada di tengah-tengah antara program atau aplikasi dan perangkat keras, dan bertindak sebagai pembagi sumber daya (*resource allocator*) yang mengatur penggunaan sumber daya, seperti siklus CPU, memori, ruang penyimpanan *disk* dan alat-alat *input* dan *output* (Abdul, 2013).

Secara sederhana sistem operasi dapat didefinisikan sebagai antar muka antar *user* dengan *hardware*. Atau dengan kata lain, sistem operasi merupakan *software* yang digunakan untuk mengatur kerja *hardware* serta menyediakan lingkungan dimana seorang *user* dapat menjalankan program aplikasi. Sistem operasi adalah sekumpulan rutin perangkat lunak yang berada diantara program aplikasi dan perangkat keras. Sistem operasi memiliki tugas yaitu mengelola seluruh sumber daya sistem komputer dan sebagai penyedia layanan Ari, (2010).
Tugas dari sistem operasi yaitu :

Berdasarkan kemampuan untuk menangani *user* dan proses yang dijalankan, sistem operasi dapat digolongkan menjadi sistem operasi *standalone* dan sistem operasi *multiuser*. Pada sistem operasi *standalone*, komputer hanya dapat melayani satu *user* pada saat bersamaan, tetapi proses yang ditangani bisa lebih dari satu pada saat yang bersamaan. Sedangkan pada sistem operasi *multiuser*, komputer dapat digunakan untuk melayani proses dari banyak pemakai pada saat yang bersamaan. Semua sistem operasi terdiri atas tiga bagian utama, yaitu :

1. *Kernel*

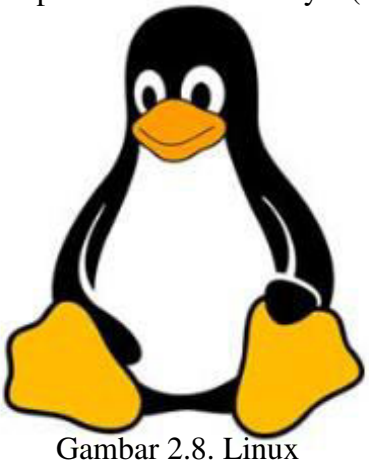
2. *Utilitas Standar*
3. *File database Sistem*

Sistem operasi yang dikenal antara lain :

a. Windows (95, 98, ME, 2000, XP, VISTA, SERVER, Windows7)
b. Linux (*Red Hat, Slackware, Ubuntu, Fedora, Mikrokomputer, Debian, OpenSUSE*)
c. UNIX
d. FreeBSD (*Berkeley Software Distribution*) -
e. SUN (SOLARIS)
f. DOS (MS-DOS)
g. Machintosh (MAC OS, MAC OSX)

2.10. GNU- Linux

Linux adalah sebuah sistem Operasi yang dibuat oleh Linus Torvald dari Finladia pada tahun 1991. Cara kerja dan tampilan dari *linux* mirip seperti sistem operasi *UNIX* yang merupakan sebuah hasil implementasi dari *standar Institute of Electrical and Electronics Engineers* (IEEE) untuk *Operating System* yang bernama *Portable Operating System Interfaces* (POSIX). Kemampuan yang berbasis standar *POSIX* seperti *multitasking, virtual memory, shared libraries, proper memory management, dan multiuser* telah dimiliki oleh *Linux*. Hampir semua *software* gratis (*free software*) yang diorganisasikan oleh GNU dapat berjalan di *Linux*. Bahkan Linux memiliki 7 *performance* yang lebih baik dibandingkan dengan sistem operasi komersial lainya. (Gery, 2011).



Gambar 2.8. Linux
Sumber : http://www.pengertianku.net/2017/08/pengertian-linux-dan-contohnya.html

2.11. Distribusi Debian

Paket *Linux* dapat diperoleh dalam berbagai distribusi. Distribusi atau lebih sering disebut *Distro*, adalah sebutan untuk semua sistem operasi yang menggunakan *kernel Linux*. Salah satu distribusi dari *Linux* yang sangat populer dan juga akan digunakan dalam pengembangan sistem jaringan ini yaitu *distro Debian* GNU/*Linux*. *Debian* memiliki komunitas yang sangat luas, serta melahirkan berbagai macam *distro-distro* turunannya Bowo, (2010).

Berikut ini adalah beberapa hal menarik tentang Debian :

a) 100% gratis

b) Rilis distribusi stabil dan teruji

c) Banyak tersedia paket dan pengaturan paket dengan fasilitas APT

d) Mendukung banyak arsitektur perangkat keras

e) Dukungan komunitas yang sangat luas.

Gambar 2.9. Debian
Sumber : https://erlanggablc.blogspot.com/2017/03/sejarah-dan-pengertian-linux-debian.html

2.12. Protocol

Protokol merupakan prosedur yang mengatur beberapa fungsi yang ada pada setiap komputer. Tugas dari protokol sendiri adalah mengatur hubungan atas komunikasi data dimulai sampai dengan komunikasi data diakhiri. Protokol dapat memperlancar proses transformasi data Nugroho, (2005).

Ada beberapa jenis protokol yang sangat berhubungan dengan jaringan *internet*, di antaranya :

1. UDP (*User Datagram Protocol*)
2. TCP (*Transmission Control Protocol*)
3. FTP (*File Transfer Protocol*)
4. RDP (*Remote Dektop Protocol*)

2.13. TCP / IP

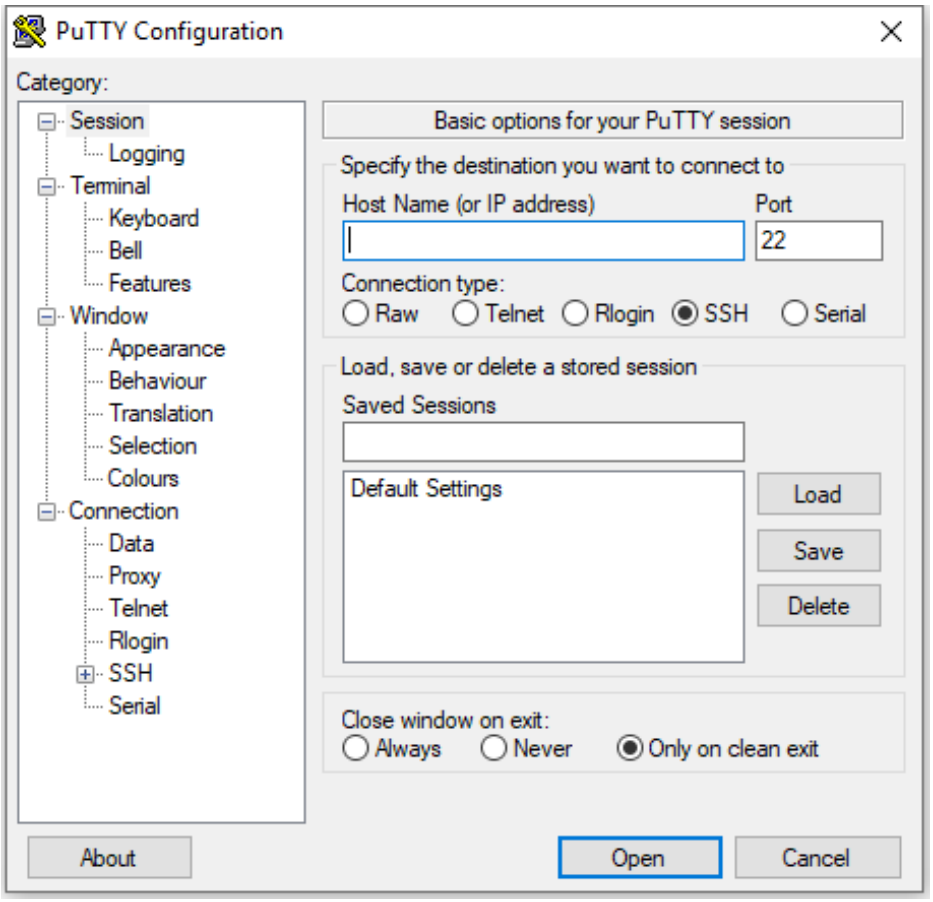
TCP/IP adalah standar komunikasi data yang digunakan dalam proses pertukaran data dari satu komputer ke komputer lainnya dalam suatu jaringan. *Transport layer* bertugas melakukan segmentasi (membagi data) dan menggabungkan kembali data yang tersegementasi (Pradikta, 2013).

2.14. PING

PING (*Packet Internet Groper*) adalah *software* yang berjalan di atas *protocol* ICMP (*Internet Control Message Protocol*) untuk mengecek hubungan antara dua komputer di *internet*. PING beroperasi dengan mengirimkan sebuah *packet* kepada suatu alamat yang dituju dan menunggu respon balik dari *host* yang dituju tersebut. REPLY pada halaman PING merupakan salah satu program yang digunakan untuk mengecek komunikasi antar komputer dalam sebuah jaringan melalui protokol TCP/IP. *Bytes* adalah ukuran *default* paket *Internet Control Message Protocol* (ICMP) PING yaitu 32 *bytes*. *Time* pada halaman PING mengindikasikan ketersediaan *bandwidth* yang disediakan untuk paket PING. TTL adalah nilai waktu termasuk dalam paket yang dikirim melalui TCP / IP. RTO (*Request Time Out*) adalah ketika komputer *server* tidak merespon permintaan koneksi dari *client* setelah beberapa lama (jangka waktu *time out* bervariasi). *Destination Host Unreachable* menginformasikan bahwa *host, port*, dan protokol tertentu tidak dapat dijangkau atau masih mencari (Ardiansyah, 2017).

2.15. Putty

Putty adalah sebuah program *open source* yang dapat Anda gunakan untuk melakukan protokol jaringan SSH, *Telnet* dan *Rlogin*. Aplikasi ini merupakan aplikasi *portable* sehingga tidak perlu di install. Protokol ini dapat digunakan untuk menjalankan sesi *remote* pada sebuah komputer melalui sebuah jaringan, baik itu LAN, maupun *internet*. Program ini banyak digunakan oleh para pengguna komputer tingkat menengah ke atas, yang biasanya digunakan untuk menyambungkan, mensimulasi, atau mencoba berbagai hal yang terkait dengan jaringan. Program ini juga dapat Anda gunakan sebagai *tunnel* di suatu jaringan Andi,(2010).



Gambar 2.10. Putty (sumber : Herlambang, 2016)

2.16. Pengertian SSH

Beberapa pengertian umum tentang SSH, diantaranya :

1. SSH (*secure shell hosting*) adalah protokol atau aplikasi yang memungkinkan pertukaran data antara dua perangkat jaringan yang lebih

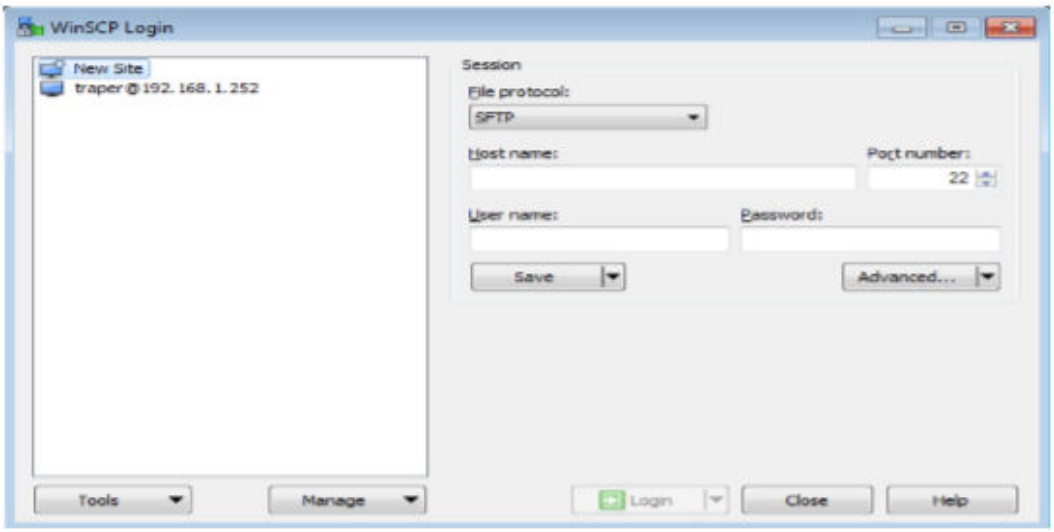
aman dibandingkan dengan *telnet, rsh* dan *rlogin*. SSH banyak digunakan pada sistem berbasis *linux* dan *unix* untuk mengakses akun *shell*. SSH pertama kali dikembangkan oleh *openBSD project* dan kemudian versi rilis p (*port*) di *manage* oleh *team porting* ke sistem operasi lainnya, temasuk *linux*. Dengan SSH semua percakapan antara *server* dan *client* dienkripsi, artinya apabila percakapan tersebut disadap, penyadap tidak akan memahami isinya.

2. SSH (*secure socket shell*) atau yang sering disebut *secure shell* adalah protokol jaringan berbasis *unix* yang memungkinkan *user* untuk mengakses sebuah komputer (*remote*) melalui jaringan secara aman. Dikarenakan SSH menggunakan jaringan yang dienkripsi, maka SSH ini banyak digunakan oleh seorang *admin* jaringan untuk mengontrol sebuah *server web* atau sebuah komputer dari jauh (*remote*).
3. Secure shell atau SSH merupakan protokol *network* yang memungkinkan pengguna untuk membuka jendela akses pada komputer lokal dan terhubung ke komputer *remote/server*, sehingga pengontrol seperti berada di depan *server* target. Dengan demikian SSH menyediakan koneksi aman dari *black hacker* untuk transfer data antar 2 komputer.
4. SSH merupakan paket program yang digunakan sebagai pengganti yang aman untuk *rlogin*, *rsh* dan *rcp* dengan menggunakan *public-key cryptography* untuk mengenkripsi komunikasi antara dua *host*, demikian pula untuk autentikasi pemakai. Protokol ini digunakan untuk *login* secara aman ke *remote host* atau menyalin data antar *host*, sementara mencegah *man in the middle attacks* (pembajakan sesi) dan DNS *spoofing* atau dapat dikatakan *secure shell* adalah program yang melakukan *login* terhadap komputer lain dalam jaringan, mengeksekusi perintah lewat mesin secara *remote*, dan memindahkan *file* dari satu mesin ke mesin lainnya.
5. SSH dirancang untuk menggantikan protokol *telnet* dan FTP. Adapun SSH merupakan produk serbaguna yang dirancang untuk melakukan banyak hal, yang kebanyakan berupa penciptaan *tunnel* antar *host*. Beberapa implementasi SSH tergantung pada SSL libraris karena SSH dan SSL menggunakan banyak menggunakan algoritma enkripsi yang sama

(misalnya Algoritma enkripsi lain yang didukung oleh SSH di antaranya *BlowFish* (*BRUCE SCHNEIER*), IDEA (*The International Data Encryption Algorithm*), dan RSA (*The Rivest-Shamir-Adelman*) Dwi Cahyani, (2010).

2.17. Winscp 533

Winscp adalah aplikasi yg berfungsi untuk *transfer file* atau *copy file* antara *windows* dengan *linux*. Kegunaan dari WinSCP ini adalah sebagai alat untuk *transfer*, atau lebih familiar kita kenal dengan sebutan *upload* dan *download file* melalui protokol ftp dan *secure shell* (SSH), Dengan WinSCP kita juga dapat melakukan editorial seperti mengedit isi *file*, merubah nama *file* menghapus *file* dan lain sebagainya Dhulkifli, (2015).



Gambar 2.11. WinSCP (sumber : Herlambang, 2016)

2.18. FTP

FTP didefinisikan sebagai sebuah protokol untuk mengirim dan menerima *file* antara *host* (dalam ARPANET), dengan fungsi utama dari FTP adalah mengirim dan menerima *file* dengan efisien dan handal antara *host* dan mengijinkan penggunaan yang nyaman dari kemampuan untuk penyimpanan *file* secara *remote*. FTP menggunakan autentikasi dengan *username* dan *password* untuk menambah privasi pada data yang di-*sharing*. Sayangnya, *username* dan *password* yang digunakan terkirim dalam bentuk tidak terenkripsi. Hal tersebut menyebabkan transmisi FTP rentan terhadap penyadapan data. Jika transmisi FTP

berhasil disadap, pesan yang ada didalamnya dapat dengan mudah dibaca oleh pihak luar Ricoh, (2010).

2.19. IP Address

Alamat IP (IP *Address*) adalah alamat yang diberikan pada jaringan komputer dan peralatan jaringan yang menggunakan protokol TCP/IP. TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data komputer di internet menurut Arifin, (2011).

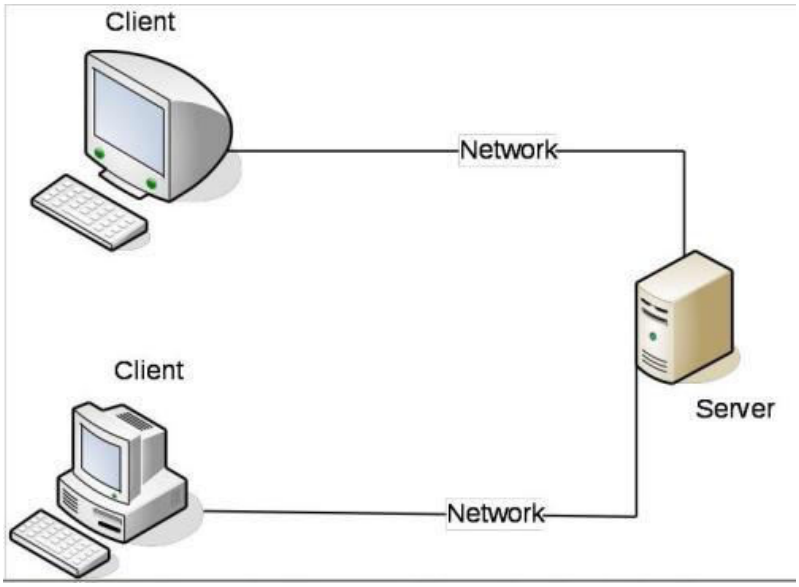
Setiap komputer yang terhubung ke internet setidaknya harus memiliki sebuah alamat IP pada setiap perangkat yang terhubung ke *internet* dan alamat IP itu sendiri harus unik karena tidak boleh ada komputer/*server*/perangkat jaringan lainnya yang menggunakan alamat IP yang sama di internet. Alamat IP versi 4 (IPv4) merupakan deretan bilangan biner sepanjang 32 bit yang digunakan untuk memberikan tanda pengenal pada perangkat di jaringan Winarno, (2010).

2.20. Client – Server

Client-server yaitu jaringan komputer yang didedikasikan khusus sebagai *server*. Sebuah layanan bisa diberikan oleh sebuah komputer atau lebih. *Server* adalah komputer yang dapat memberikan layanan ke *client*, sedangkan *client* adalah komputer yang mengakses beberapa layanan yang ada di *server*. Ketika *client* membutuhkan suatu *service* yang ada di *server*, dia akan mengirim *request* kepada *server* lewat jaringan. Jika *request* tersebut dapat dilaksanakan, maka *server* akan mengirim balasan berupa *service* yang dibutuhkan untuk saling berhubungan.

Client-server merupakan teknologi pendistribusian kerja aplikasi anatara dua komputer atau lebih, yang dihubungkan oleh jaringan komunikasi, dimana yang satu akan bertindak sebagai *client* ataupun *server*. Baik *client* ataupun *server* memiliki pemrosesan jaringan yang digunakan, bisa berupa jaringan lokal (LAN) ataupun jaringan yang lebih luas lagi (WAN). Sebuah jaringan dan sistem operasi bekerjasama membentuk sebuah unit. Biasanya sebuah mesin *server* akan di-install sistem operasi *server* seperti MS Windows NT Server, Windows 2000 Server, Linux, Unix, dll Rizki, (2010).

Dalam model *client-server*, sebuah aplikasi dibagi menjadi dua bagian yang terpisah, akan tetapi masih merupakan sebuah kesatuan yakni komponen *client* dan komponen *server*. Komponen *client* juga sering disebut sebagai *front-end*, sementara komponen *server* disebut sebagai *back-end*. Komponen *client* dari aplikasi tersebut dijalankan dalam sebuah *workstation* dan menerima masukan data dari pengguna. Komponen *client* tersebut akan menyiapkan data yang dimasukkan oleh pengguna dengan menggunakan 28 teknologi pemrosesan tertentu dan mengirimkannya kepada komponen *server* yang dijalankan di atas mesin *server*, umumnya dalam bentuk *request* terhadap beberapa layanan yang dimiliki oleh *server*. Komponen *server* akan menerima *request* dari *client*, dan langsung memprosesnya dan mengembalikan hasil pemrosesan tersebut kepada *client*. *Client* pun menerima informasi hasil pemrosesan data yang dilakukan *server* dan menampilkannya kepada pengguna,dengan menggunakan aplikasi yang berinteraksi dengan pengguna Yudianto, (2007).



Gambar 2.12. Client-Server
Sumber : https://alyscompt.wordpress.com/2012/11/21/pemrograman-client-server/

BAB III
METODELOGI PENELITIAN

3.1. Metode Penelitian

Penelitian ini menggunakan metode penelitian observasi karena dalam pengumpulan data yang diperoleh dilakukan dengan cara mengadakan pengamatan langsung terhadap objek yang diteliti. Pada penelitian ini akan membangun *firewall* untuk keamanan jaringan komputer menggunakan *shorewall* pada *Linux Debian* 10.

3.2. Waktu dan Tempat Penelitian

Penelitian ini di lakukan pada kantor PT. Indofarma Global Medika Jl. Ir. Mohd Thair No. 5A-5D Gp. Lamdom Kec. Lueng Bata – Banda Aceh. Penelitian di laksanakan mulai September 2019 sampai dengan selesai.

3.3. Alat dan Bahan

Perangkat keras (*Hardware*) yang digunakan dengan spesifikasi yang cukup untuk penelitian ini. Adapun alat yang digunakan sebagai berikut :

1. Perangkat Keras
    a. Komputer dengan spesifikasi Intel Core i3
    b. RAM 4 GB
    c. HDD 500 GB
2. Perangkat Lunak
    a. Sistem Operasi Windows 10 32-bit
    b. Sistem Operasi Linux Debian 10 (Buster) 64-bit
    c. Microsoft Office Word
    d. Terminal

3.4. Jadwal Penelitian

Tabel 3.1. Jadwal Penelitian

No	Uraian	2019-2020																							
		September				Oktober				November				Desember				Januari				Februari			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Pengajuan Judul Proposal			■																					
2.	Mengumpulkan Data dan Referensi					■	■																		
3.	Rancangan Sistem							■	■																
4.	Analisis Sistem									■															
5.	Implementasi Rancangan											■	■												
6.	Penyusunan Proposal													■	■										
7.	Pengujian Sistem																	■	■						
8.	Penyusunan Skripsi																						■	■	■

3.5. Pembahasan *Firewall* dan *Shorewall*

3.5.1. Metode Filtering *Firewall*

Pada umumnya *firewall* menggunakan beberapa metode, pada penelitian ini metode yang penulis gunakan yaitu *packet filtering* alias penyaringan paket. *Packet filtering* ini melakukan pemeriksaan sederhana paket data. Kemudian, setelah itu informasi yang didapatkan seperti nomor *port*, alamat *ip* tujuan dan asal, juga informasi tingkat permukaan lainnya diperiksa tanpa membuka paket untuk memeriksa isinya. Langkah akhir adalah jika paket informasi tersebut setelah diperiksa tidak lulus inspeksi, maka paketnya akan dibuang.

3.5.2. *File-file shorewall*

Berikut adalah *file-file* penting yang harus *disetting* dalam mengkonfigurasi *Shorewall*, yaitu :

1. */etc/shorewall/zone*

File ini untuk mendefinisikan zona asal trafik pada jaringan. *Server* tempat *shorewall* diinstal dikenal sebagai zona yang disebut fw.Pada *file* ini, *local* yang merupakan *interface* yang terhubung dengan jaringan *local* dan *net* merupakan *interface* yang terhubung dengan jaringan *network*.

 # ZONE DISPLAY COMMENTS

fw firewall

net Internet

loc local networks

2. */etc/shorewall/policy*

File ini berisi aturan untuk semua trafik yang lewat pada *firewall* diatur pada */etc/shorewall/rules*, jika tidak terdefinisikan pada *file* tersebut maka akan dicek pada */etc/shorewall/policy*.

#SOURCE ZONE DESTINATION POLICY LOG

loc net ACCEPT

3. */etc/shorewall/interface*

File ini untuk menentukan *interface* yang akan terhubung dengan suatu zona, pada *file* ini, *eth0* terhubung dengan jaringan *internet* dan *eth1* terhubung dengan jaringan lokal.

#ZONE INTERFACE BROADCAST OPTIONS

net eth0

loc eth1

4. */etc/shorewall/rules*

File ini berisi aturan-aturan dari semua trafik yang melewati *firewall*. Berikut contok konfigurasinya :

#*Rule* dari *local* ke mesin firewall

ACCEPT loc fw tcp 23

ACCEPT loc fw tcp 80

6. */etc/shorewall/shorewall.conf*

Digunakan untuk mengaktifkan *shorewall* agar dapat di *load* pada saat *startup*.

Startup_ENABLE = Yes

3.5.3. Kebijakan-Kebijakan *Shorewall*

1. *Accept*

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*.

ACCEPT loc fw tcp 80

2. *Drop*

Berbeda dengan *REJECT*, bila *firewall* menemukan paket yang di-*DROP, firewall* akan langsung ”membuang” setiap paket yang memiliki target ini tanpa mengirim pesan *error* kepada pengirim paket tersebut.

DROP loc fw tcp 23

3. *Reject*

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

all all REJECT

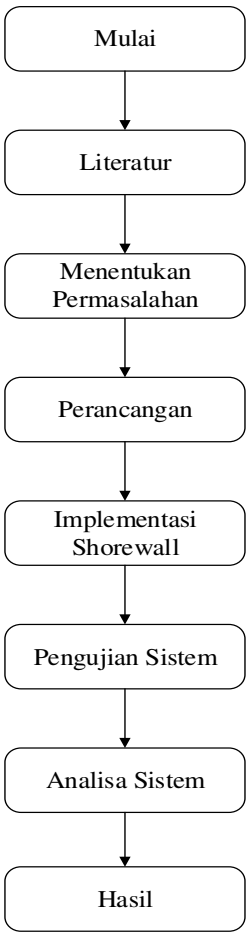
3.5.4. Perintah-Perintah *Shorewall*

Tabel 3.2 Perintah-perintah *Shorewall*

Perintah Shorewall	Keterangan
Shorewall start	Perintah untuk memulai firewall dengan shorewall
Shorewall stop	Perintah untuk menghentikan firewall dengan shorewall
Shorewall restart	Perintah untuk kembali ke konfigurasi awal
Shorewall clear	Perintah untuk menghapus semua aturan – aturan shorewall

Sumber : http://jurnal.stmikmethodistbinjai.ac.id/

3.6. Alur Penelitian



Gambar 3.1. Alur penelitian

Adapun tahapan-tahapannya sebagai berikut :

1. Studi Literatur

   Studi literatur adalah sebuah kegiatan mengumpulkan, mempelajari beberapa buku atau jurnal tentang *firewall* dan *shorewall* yang akan di jadikan penunjang penelitian.

2. Menentukan Permasalahan

   Dari hasil studi literatur sebelumnya, maka ditemukan permasalahan terkait dengan keamanan jaringan komputer pada PT.IGM yaitu memberikan batasan izin akses jaringan yang masuk pada *server*.

3. Perancangan

   Dalam pembuatan tugas akhir ini penulis merancang sebuah komputer *server* menjadi *firewall* dengan menggunakkan *Shorewall* yang merupakan

perangkat bawaan sistem operasi *Linux Debian*. pertama yang dilakukan adalah dengan menginstal sistem operasi *linux debian* di sebuah komputer *server*, kemudian install dan aktifkan *Shorewall*.

4. Implementasi *Shorewall*

   Implementasi sistem adalah sebuah kegiatan membangun hasil rancangan menjadi sebuah perangkat lunak. Pada tahap ini pula akan di bangun *firewall* menggunakan *shorewall*, baik itu proses instalasinya maupun konfigurasinya pada sistem operasi *Linux Debian* 10.

5. Pengujian

   Dalam pengujian *firewall* dengan *shorewall* ini penulis menggunakan 3 (tiga) macam pengujian yaitu dengan pengiriman paket *ping,* program *remote dekstop SSH* menggunakan *putty* dan penggunaan *transfer file (FTP)* melalui *WinSCP*.

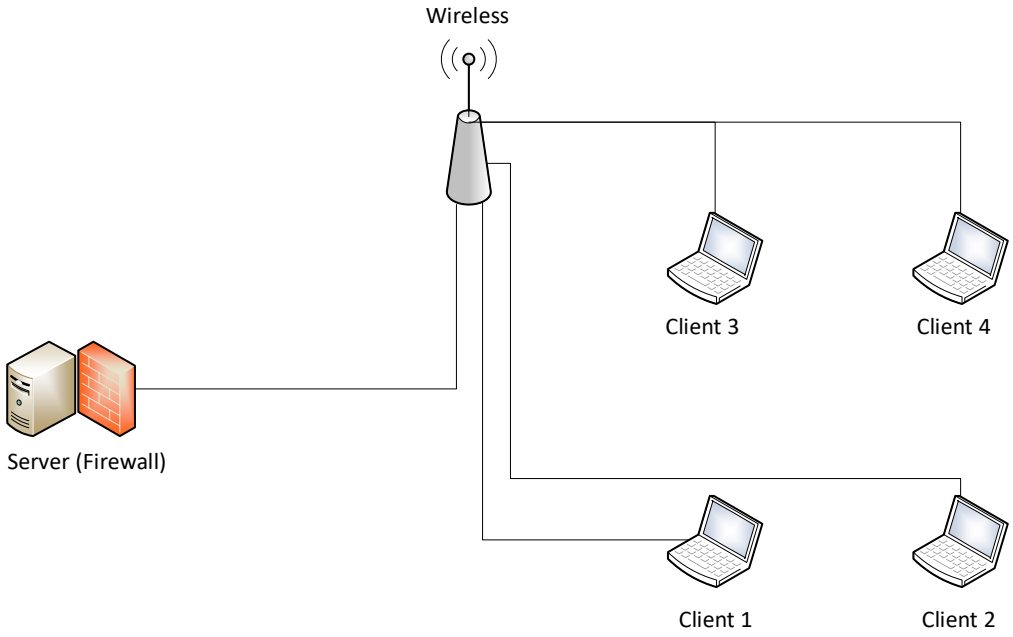
6. Analisa Sistem

   Pada tahap ini adalah dilakukan analisa terhadap sistem yang telah di buat.

7. Hasil

   Pada tahap ini adalah melakukan laporan penelitian yang meliputi hasil pengujian dan di akhiri dengan kesimpulan.

3.7. Rancangan Sistem



Gambar 3.2. Rancangan sistem

Pada gambar 3.2, terdapat 4 komputer yaitu client 1, client 2, client 3, client 4 dan 1 komputer *server*. Disini penulis akan membangun sebuah *firewall* menggunakan *shorewall* yang terdapat pada komputer *server*. Untuk perancangan *firewall* ini *Operating System* (OS) yang digunakan adalah *Linux Debian Server* 10, *shorewall* di *Debian Server* 10 belum terinstall saat penulis melakukan instalasi di komputer, jadi penulis perlu melakukan installasi *shorewall* secara tersendiri.

3.8. Perencanaan Pengujian

1. Pengujian Pengiriman *Packet Ping*

Pada pengujian mengirim *packet ping* ini, masing-masing komputer client akan mengirimkan *packet ping* ke komputer *server* dengan *ip address* 192.168.43.10. Kebijakan-kebijakaan yang akan di gunakan pada pengujian ini yaitu *ACCEPT, DROP*, dan *REJECT* yang akan di konfigurasi pada *file rules shorewall*.

Pada saat sebelum *shorewall* dikonfigurasi pengiriman paket *ping* dalam keadaan lancar-lancar saja. Namun setelah dilakukan konfigurasi pada *rules shorewall* yaitu pada bagian *Ping/ACCEPT* loc:192.168.43.77 $FW maka yang boleh melakukan pengiriman paket *ping* ke komputer *server* hanya komputer *client* yang memiliki *ip* tersebut dan pada konfigurasi ini dalam jaringan *local* saja, sedangkan pada jaringan *internet* tidak bisa dilakukan.

2. *Remote Desktop SSH* Dengan *Putty*

Pada pengujian *remote dekstop SSH* menggunkan *Putty* ini, masing-masing komputer *client* akan meremote komputer *server* secara bersamaan. SSH *Putty* biasa digunakan untuk *meremote* atau masuk ke dalam *resource operating* sistem yang berbasis *unix*. Dalam keadaan normal atau sebelum *rules shorewall* dikonfigurasi semua jenis jaringan bisa melakukan *remote* ke komputer *server* melalui program *Putty* ini.

Kemudian setelah dilakukan konfigurasi pada *rules shorewall* yaitu pada bagian *SSH/ACCEPT loc:192.168.43.158 $FW* maka tidak semua jenis jaringan atau nomor *IP* yang bisa meremotnya, hanya nomor *IP* 192.168.43.158 saja yang bisa melakukannya. Kebijakan yang akan digunakan pada pengujian ini yaitu *ACCEPT*, dan *DROP*.

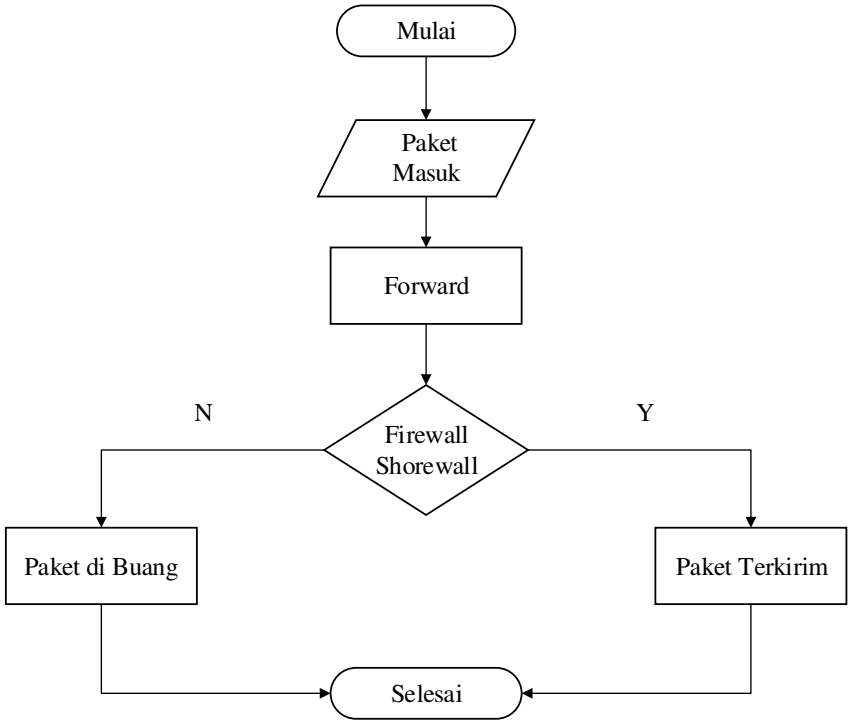
3. *Transfer File* (*FTP*) Menggunakan *WinSCP*

Pada pengujian *transfer file* (*FTP*) mengggunakan *WinSCP* ini, masing-masing komputer *client* akan melakukan *transfer file* pada komputer *server*. Penggunaan *FTP* disini *client* menggunakan aplikasi *WinSCP* untuk masuk ke dalam *resource* komputer *server*. Dalam keadaan normal atau sebelum *rules shorewall* di konfigurasi semua komputer *client* bisa melakukan *transfer file* ke komputer *server*.

Kemudian setelah dilakukan konfigurasi pada *rules shorewall* yaitu pada bagian *FTP/ACCEPT loc:192.168.43.73 $FW* maka tidak semua komputer *client* yang bisa *mentransfer file*, hanya *client* yang memiliki nomor *IP* 192.168.43.73 saja yang bisa melakukannya. Kebijakan yang digunakan pada pengujian ini yaitu *ACCEPT*, dan *REJECT*.

3.9. *Flowchart*

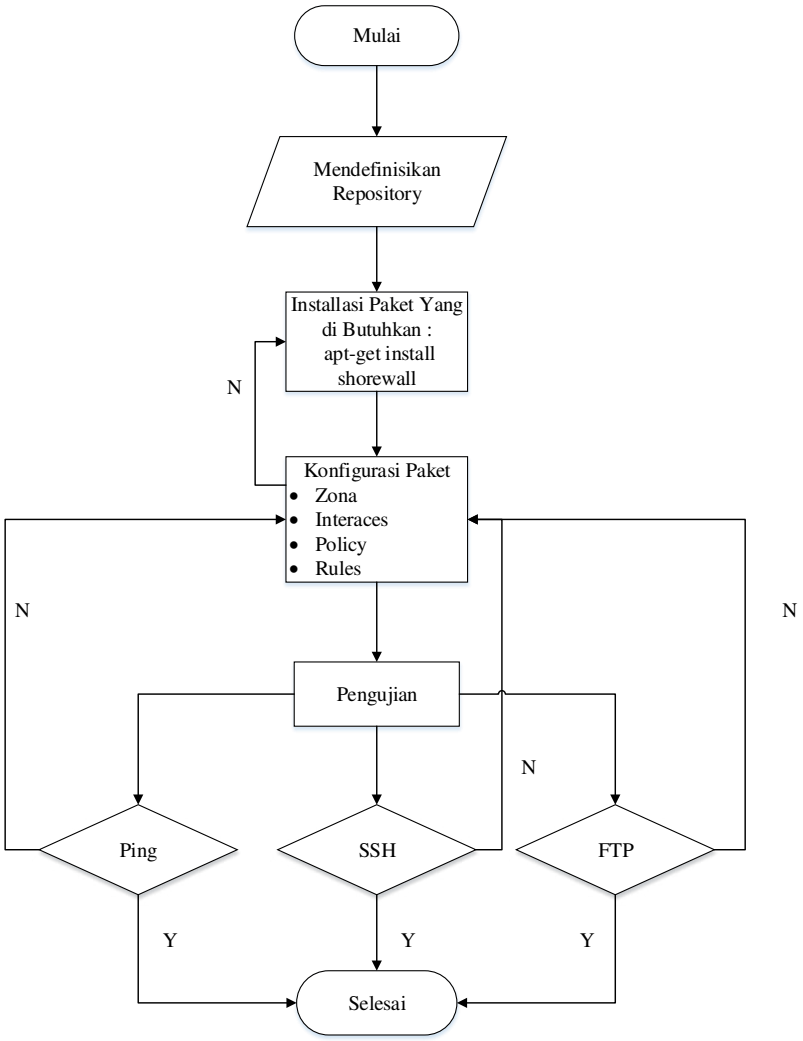
3.9.1. *Flowchart Front-end System*



Gambar 3.3 Flowchart front-end system

Pada gambar 3.3, merupakan *flowchart front-end system* dengan penjelasaan sebagai berikut, paket data yang masuk akan di *forward* ke sistem *firewall* (*shorewall*) dengan pilihan izin akses dengan status paket terkirim atau paket terbuang.

3.9.2. *Flowchart Back-end System*

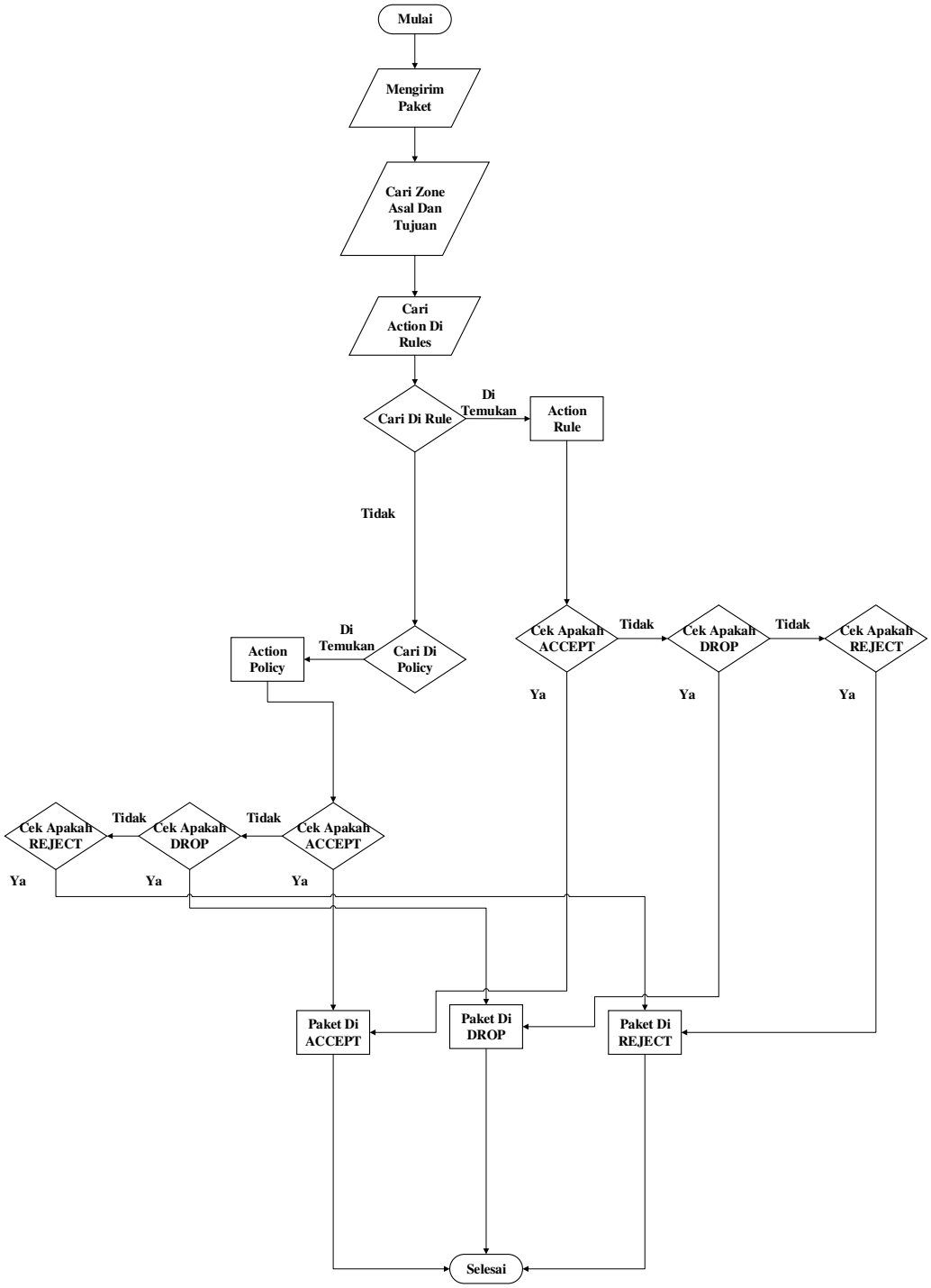


Gambar 3.4 Flowchart back-end system

Pada gambar 3.4, merupakan *flowchart back-end system* dengan penjelasan sebagai berikut, di awali dengan mendefinisikan *repository* guna untuk memasukkan paket-paket instalasi yang di butuhkan pada perancangan *firewall* menggunakan *shorewall* ini. Disini penulis melakukan atau memasukkan *repository* nya dengan cara *online*. Selanjutnya melakukan instalasi paket yang di butuhkan yaitu paket *shorewall* dengan menggunakan perintah ***apt-get install shorewall***. Kemudian untuk konfigurasi paket yang terdapat pada *shorewall* itu ada 4 *file* yang harus di konfigurasi, di antaranya yaitu *file zone, file interfaces, file policy, file rules*. Selanjutnya proses pengujian, disini penulis menggunakan 3 pengujian yaitu pengiriman paket *ping (ICMP)*, *remote dekstop SSH* dan *transfer*

file (FTP), jika kondisi paket tesebut dijalankan dan diberi batas akses (*firerwall mode on*) maka ketiga paket tersebut tidak akan berjalan, sebaliknya jika tidak di batasi akses (*firewall mode off*) paket tersebut pasti bisa di jalankan sesuai fungsinya.

3.5.3. *Flowchart Filtering* Paket



Gambar 3.5. Flowchart filtering packet

Berdasarkan gambar 3.5 maka penjelasannya sebagai berikut :

Pada saat *client* mengirim paket data ke komputer *server(firewall)*, sistem *firewall* akan mencari komputer asal dan komputer tujuan, kemudian mencari aturan di *rules*, di cek *action*nya apakah *ACCEPT*, *DROP*, dan *REJECT*, lalu di tampilkan sesuai *action* yang ditemukan. Apabila tidak di temukan di *rules* maka sistem akan melakukan pencarian di *policy*.

BAB IV
HASIL DAN PEMBAHASAN

4.1. Installasi dan Konfigurasi *Firewall* Menggunakan *Shorewall*

Pada perancangan *firewall* dengan *shorewall* mengenai bagaimana cara mengimplementasikan dan mengkonfigurasi *shorewall* untuk mengatur keluar dan masuknya paket data serta kebijakan-kebijakan yang perlu dilakukan dalam pengamanan komputer yang terhubung ke jaringan.

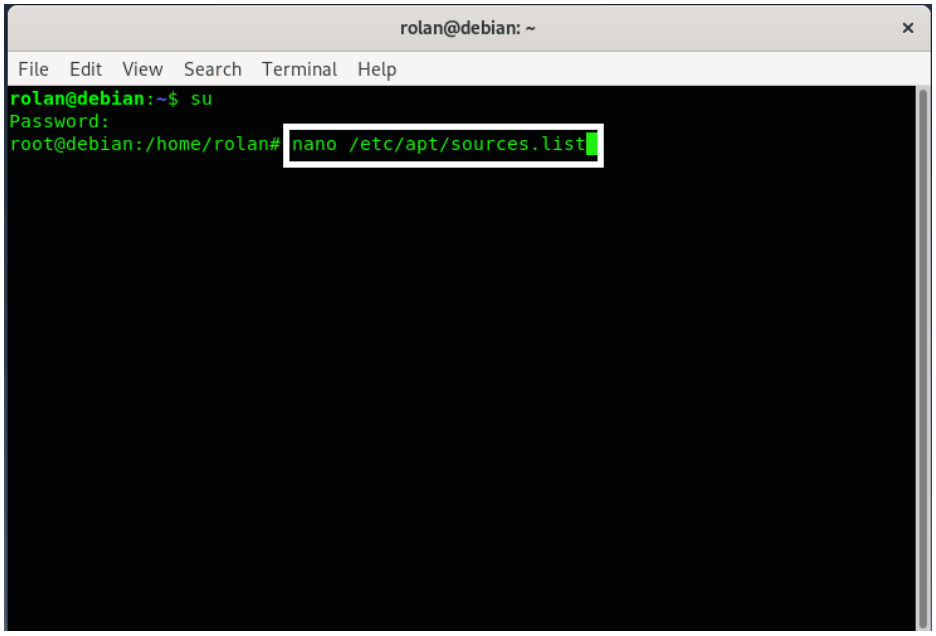
Sebelum melakukan installasi dan konfigurasi *firewall* menggunakan *shorewall*, ada beberapa tahapan yang harus dilakukan terlebih dahulu, yaitu *repository* dan installasi paket *SSH* dan *FTP*.

4.1.1. *Repository*

Repository merupakan sekumpulan paket-paket aplikasi atau program untuk sebuah sistem operasi *linux* yang digunakan untuk menunjang kinerja dari sebuah aplikasi, program, dan sebagainya yang didapatkan dari *server mirror* atau cd/dvd atau media penyimpanan lainnya.

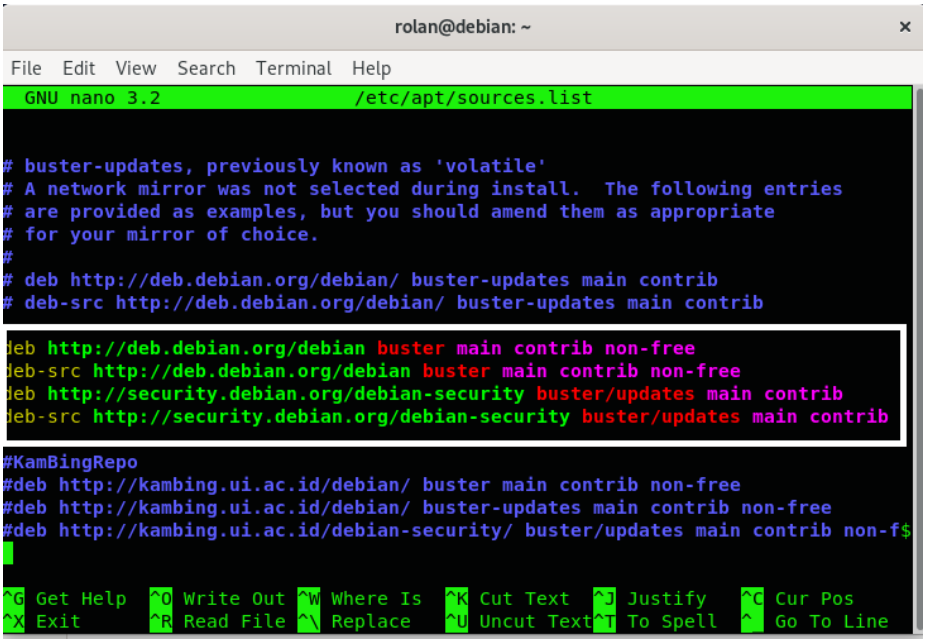
Repo yang penulis gunakan ada dua yaitu *repo local* akses dan *internet* akses. Berikut langkah-langkah konfigurasi *repository*, yaitu :

1. Buka *terminal Linux* kemudian *login* menggunakan *user root* atau *super user* dengan menggunakan perintah **su** pada *terminal Linux*.
2. Buka *file* konfigurasi *repository* menggunakan perintah ***nano /etc/apt/source.list***. Seperti pada gambar 4.1.



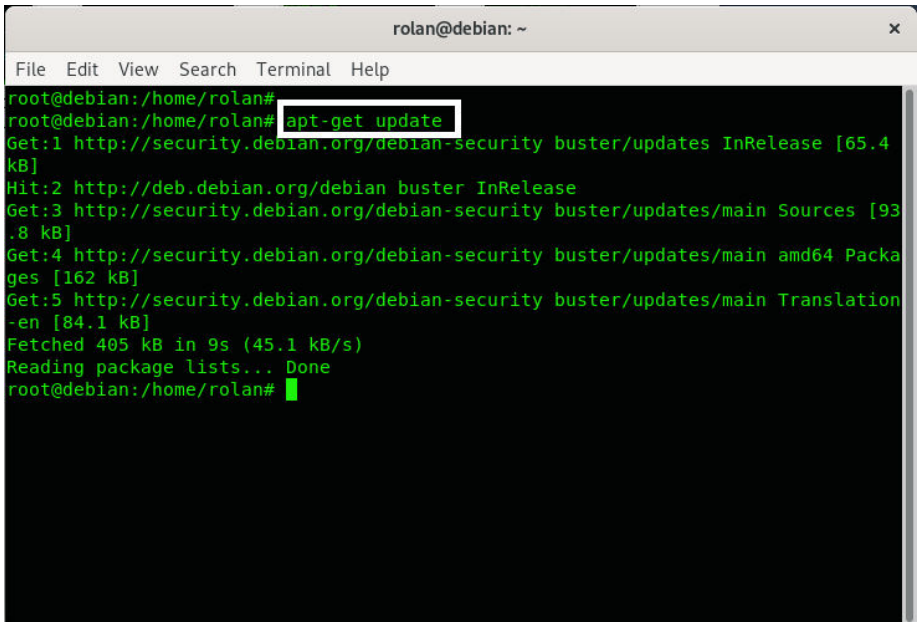
Gambar 4.1 Perintah masuk ke source.list repo

3. Teks yang di awali dengan tanda # dianggap sebagai komentar, tidak di eksekusi. Ubah isi *source.list* menjadi seperti pada gambar 4.2.



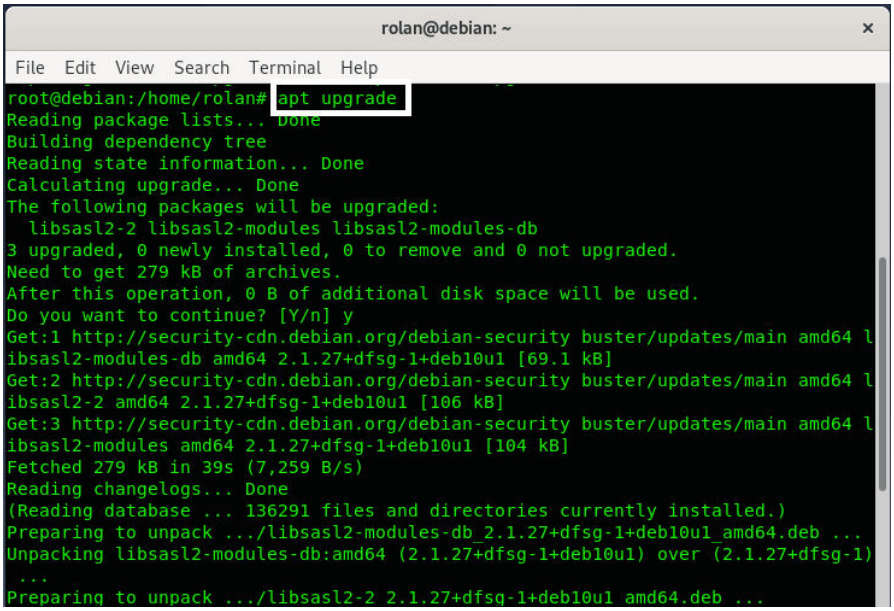
Gambar 4.2 File source.list

4. Untuk menyimpan *repository* yang telah di konfigurasi sebelumnya itu bisa dilakukan dengan tekan kombinasi **CTRL+X** kemudian tekan tombol **Y** setelah itu ***enter***.
5. Selanjutnya lakukan *update* dengan perintah ***apt-get update*** kemudian tekan ***enter***. Seperti pada gambar 4.3.



Gambar 4.3 Proses update

6. Setelah *update*, selanjutnya melakukan *upgrade* dengan perintah ***apt-get upgrade***. Proses tersebut akan menampilkan nama paket yang di *upgrade*, ukuran *file* paket yang harus di *download*, dan setelah *upgrade* berapa besar *disk space* yang akan terpakai. Jawab **Y** untuk melanjutkan proses *upgrade***,** dan proses *upgrade* bisa di lihat pada gambar 4.4.
7. Jika sudah maka proses *upgrade* telah selesai.

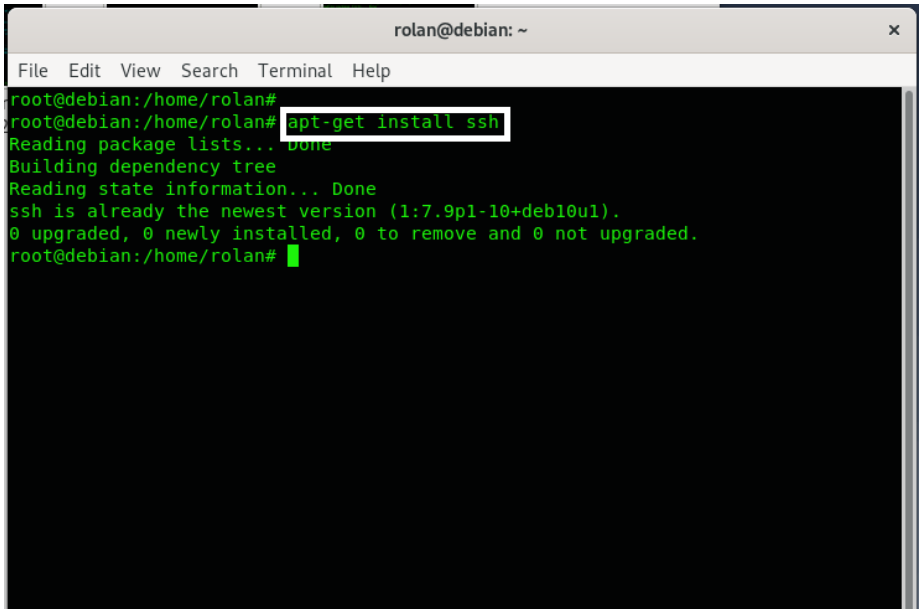


Gambar 4.4 Proses upgrade

4.1.2. Installasi Paket *SSH* dan *FTP Server*

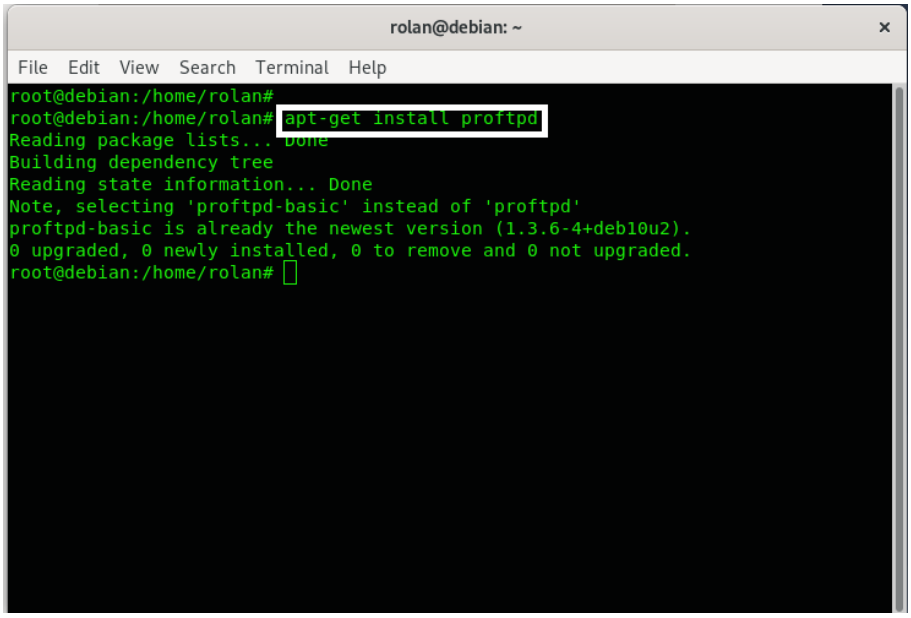
Pada perencanaan pengujian yang sebelumnya penulis paparkan yaitu menggunakan *Remote dekstop SSH* dan *File Trasnfer Protocol* (*FTP*), maka perlu di lakukan installasi paket *SSH* dan *FTP* dengan cara sebagai berikut.

1. Perintah untuk melakukan installasi *SSH* yaitu ***apt-get install ssh*** setelah itu tekan **Y** untuk melanjutkan installasi kemudian ***enter***. Seperti pada gambar 4.5.



Gambar 4.5 Proses installasi ssh

2. Jika sudah selanjutnya melakukan installasi *FTP* menggunakan perintah ***apt-get install proftpd*** setelah itu tekan **Y** untuk melanjutkan installasi kemudian ***enter***. Seperti pada gambar 4.6.

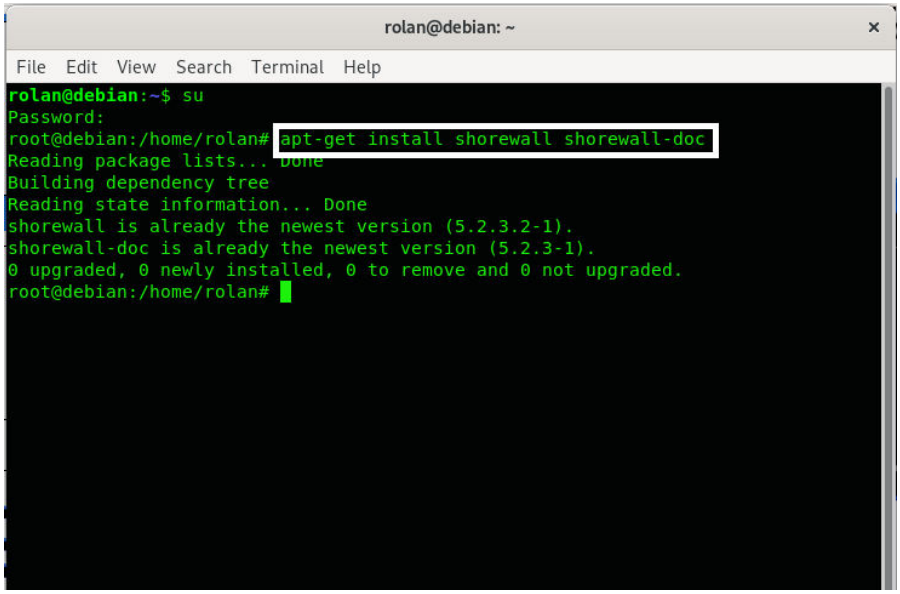


Gambar 4.6 Proses intsallasi ftp

4.1.3. Installasi *Shorewall*

Pada perancangan *firewall* ini *Operating System* (OS) yang digunakan adalah *Linux Debian Server* 10, *Shorewall* di *Debian Server* 10 belum terinstall saat kita melakukan instalasi di komputer kita, jadi kita perlu melakukan installasi *Shorewall* secara tersendiri. Di sini penulis melakukan installasi *Shorewall* langsung menggunakan *internet* dan proses installasi *shorewall* bisa di lihat pada gambar 4.7. Untuk perintah installasinya sebagai berikut :

apt-get install shorewall shorewall-doc



Gambar 4.7 Installasi shorewall

4.1.4. Konfigurasi *Firewall* Menggunakan *Shorewall*

1. *Copy file* konfigurasi *shorewall* dari *template* yang ada

   File konfigurasi tersebut terdapat pada direktori ***/usr/share/doc/shorewall/examples/three-interfaces/.*** Untuk masuk ke di rektori ini gunakan perintah sebagai berikut :

   root@debian:/home/rolan# cd /usr/share/doc/shorewall/examples/three-interfaces

2. Konfigurasi zona

   Langkah pertama *copy file* konfigurasi *zones* pada ***/usr/share/doc/shorewall/examples/three-interfaces/*** ke di rektori ***/etc/shorewall/*** dengan perintah sebagai berikut :

   root@debian:/home/rolan/usr/share/doc/shorewall/examples/three-interfaces#cp zones /etc/shorewall/

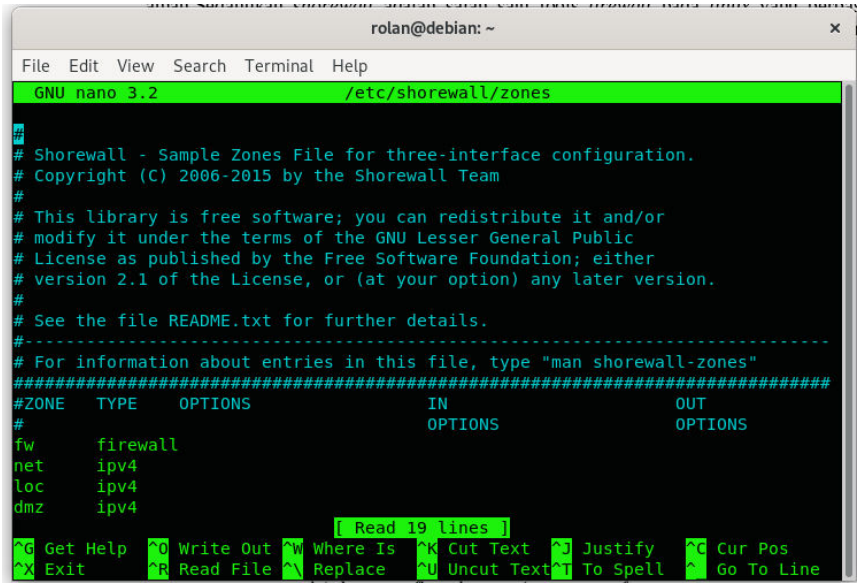
   Selanjutnya melakukan konfigurasi dengan masuk ke *file zones* menggunakan perintah sebagai berikut :

   root@debian:/home/rolan# nano /etc/shorewall/zones
   fw firewall

net ipv4

loc ipv4

dmz ipv4



Gambar 4.8 File zones

Pada gambar 4.8 merupakan konfigurasi *file zones*, *server* tempat *shorewall* diinstal dikenal sebagai zona yang disebut *fw*. Pada *file* ini, *loc* merupakan *interface* yang terhubung dengan jaringan *local* dan *net* merupakan *interface* yang terhubung dengan jaringan *network*.

3. Konfigurasi *interfaces*

Copy file konfigurasi *interfaces* pada ***/usr/share/doc/shorewall/examples/three-interfaces/*** ke direktori ***/etc/shorewall/*** dengan perintah sebagai berikut :

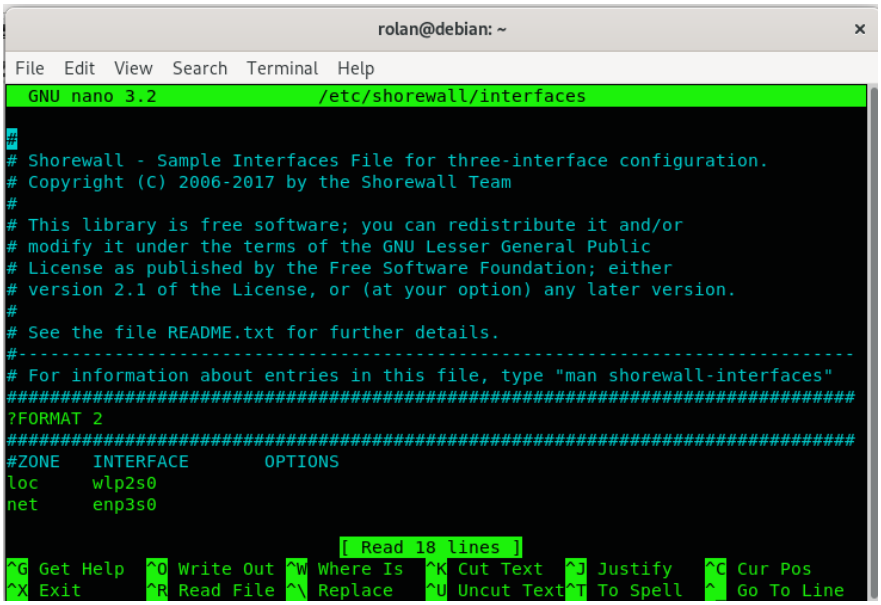
root@debian:/home/rolan/usr/share/doc/shorewall/examples/three-interfaces#cp interfaces /etc/shorewall/

Selanjutnya melakukan konfigurasi dengan masuk ke *file interfaces* menggunakan perintah sebagai berikut :

root@debian:/home/rolan# nano /etc/shorewall/interfaces

loc wlp2s0

net enp3s0



Gambar 4.9 File Interfaces

Pada gambar 4.9 merupakan konfigurasi *interfaces*, wlp2s0 merupakan *interfaces wireless* sedangkan enp3s0 adalah *interfaces ethernet*.

4. Konfigurasi *Policy*

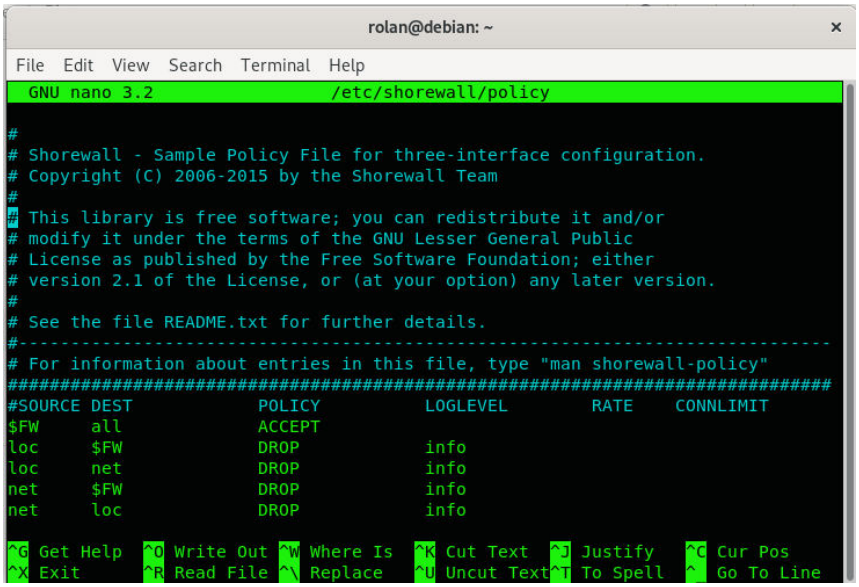
Copy file konfigurasi *policy* pada ***/usr/share/doc/shorewall/examples/three-interfaces/*** ke direktori ***/etc/shorewall/*** dengan perintah sebagai berikut :

root@debian:/home/rolan/usr/share/doc/shorewall/examples/three-interfaces#cp policy /etc/shorewall/

Selanjutnya melakukan konfigurasi dengan masuk ke *file policy* menggunakan perintah sebagai berikut :

root@debian:/home/rolan# nano /etc/shorewall/policy

$FW	***all***	***ACCEPT***	
loc	***$FW***	***DROP***	***info***
loc	***net***	***DROP***	***info***
net	***$FW***	***DROP***	***info***
net	***loc***	***DROP***	***info***



Gambar 4.10 File policy

Pada gambar 4.10 merupakan konfigurasi *file policy*, pada konfigurasi *file* ini dapat di nyatakan bahwa :

1) Semua koneksi dari zona *$FW* menuju *net* dan *loc* di terima *(ACCEPT)*
2) Semua koneksi dari zona *loc* menuju *$FW* di tolak *(DROP)*
3) Semua koneksi dari zona *loc* menuju *net* di tolak *(DROP)*
4) Semua koneksi dari zona *net* menuju *$FW* di tolak *(DROP)*
5) Semua koneksi dari zona *net* menuju *loc* di tolak *(DROP)*
6) Semua koneksi apapun selain yang didefinisikan di atas akan di tolak *(DROP)*

5. Konfigurasi *Rules*

Pada *file rules* merupakan prinsip *ACCEPT, DROP* dan *REJECT* yang ada di *shorewall*, dan untuk konfigurasinya sama seperti sebelumnya yaitu *copy file* konfigurasi *rules* pada ***/usr/share/doc/shorewall/examples/three-interfaces/*** ke direktori ***/etc/shorewall/*** dengan perintah sebagai berikut :

root@debian:/home/rolan/usr/share/doc/shorewall/examples/three-interfaces#cp rules /etc/shorewall/

Selanjutnya melakukan konfigurasi dengan masuk ke *file rules* menggunakan perintah sebagai berikut :

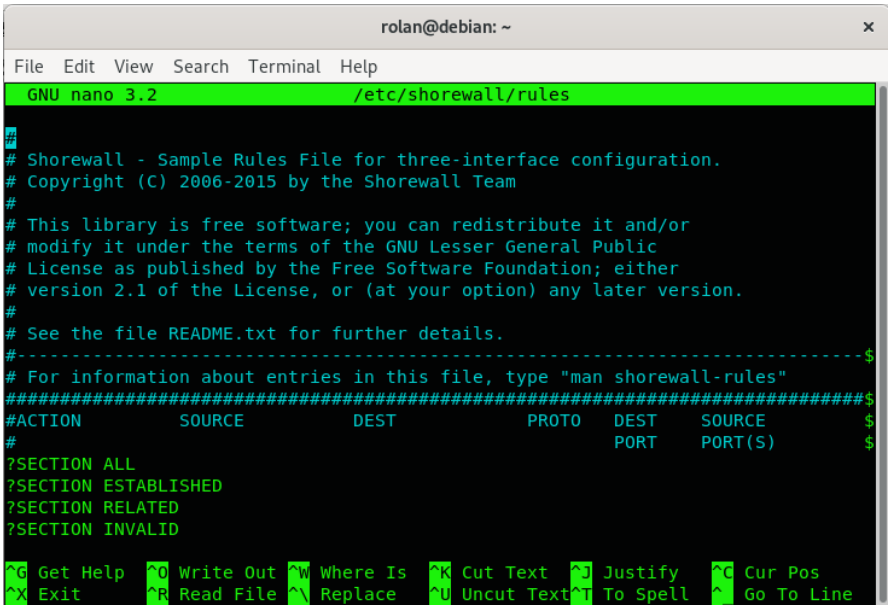
root@debian:/home/rolan# nano /etc/shorewall/rules

Ping/ACCEPT loc $FW

SSH/ACCEPT loc $FW

FTP/ACCEPT loc $FW

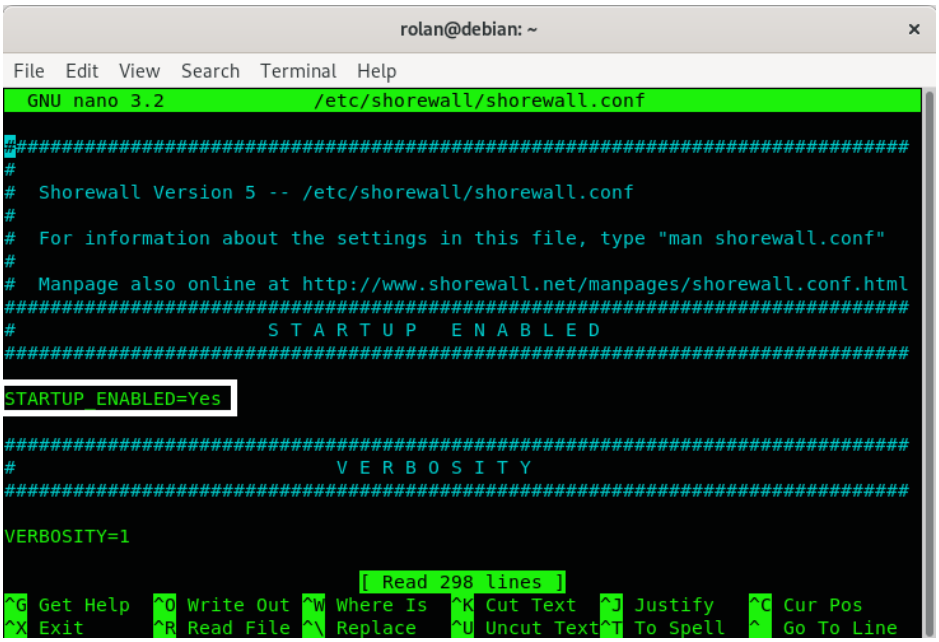
Berdasarkan konfigurasi *rules* di atas, pada *loc* ke *fw* untuk koneksi *Ping*, *SSH*, dan *FTP* di buat dalam kebijakan *ACCEPT*.



Gambar 4.11 File rules

6. Selanjutnya ubah ***Startup_ENABLE = No*** menjadi ***Startup_ENABLE = Yes*** agar dapat di *load* pada saat *startup*. Proses ini bisa di lihat pada gambar 4.12 dan untuk memasuki *file startup_enable* ini menggunakan perintah sebagai berikut.

root@debian:/home/rolan#nano /etc/shorewall/shorewall.conf

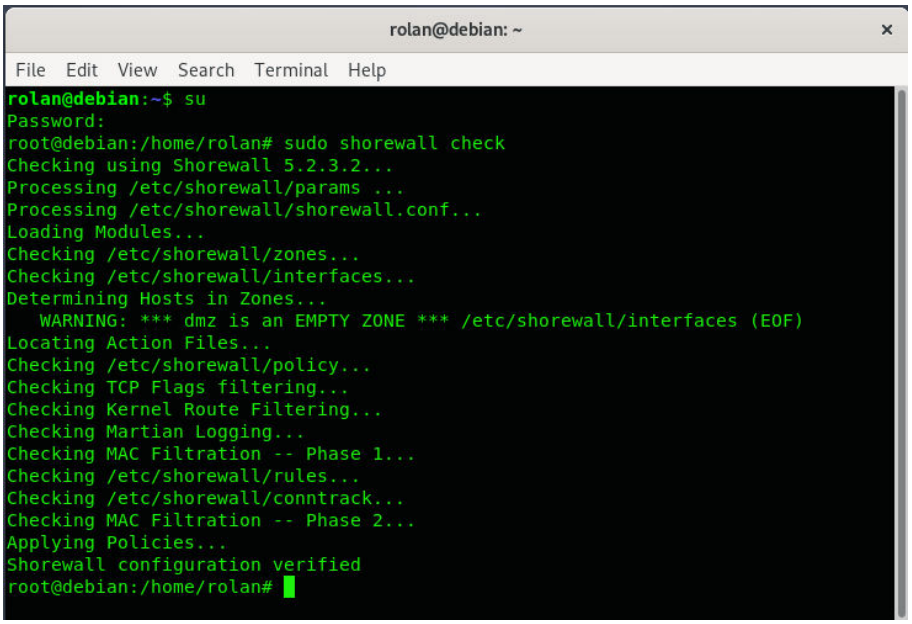


Gambar 4.12 Enable shorewall

7. Sebelum menjalankan *shorewall*, testing konfigurasi *shorewall* terlebih dahulu guna untuk mengecek konfigurasi yang telah di lakukan sebelumnya dengan perintah sebagai berikut.

root@debian:/home/rolan# sudo shorewall check

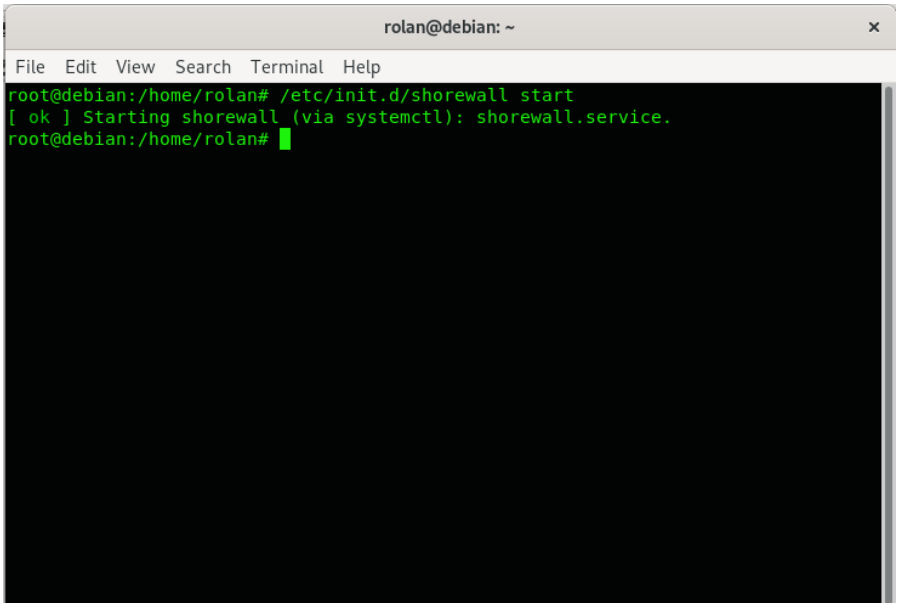
8. Untuk proses testing konfigurasi dapat di lihat pada gambar 4.13.



Gambar 4.13 Check konfigurasi

9. Untuk menjalankan atau mengaktifkan *shorewall,* gunakan perintah sebagai berikut dan untuk proses menjalankannya bisa di lihat pada gambar 4.14.

root@debian:/home/rolan# /etc/init.d/shorewall start



Gambar 4.14 Start shorewall

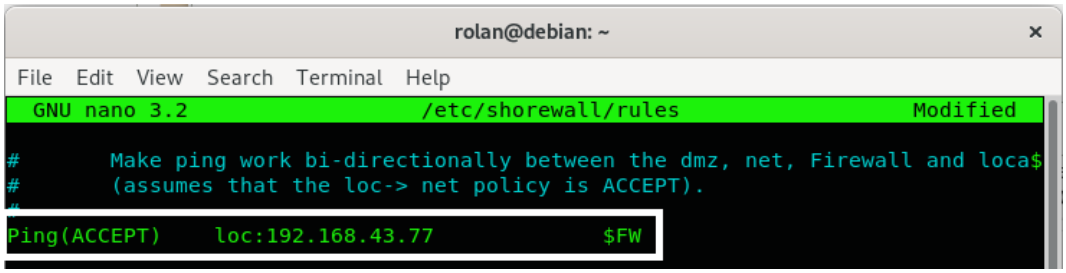
4.2. Hasil Pengujian

Dalam pengujian *firewall* dengan *shorewall* ini penulis menggunakan 3 (tiga) macam pengujian yaitu dengan pengiriman paket *ping, remote dekstop* SSH yaitu *putty* dan *File Transer Protocol (FTP)* menggunakan *WinSCP*.

4.2.1. Pengiriman *Packet Ping*

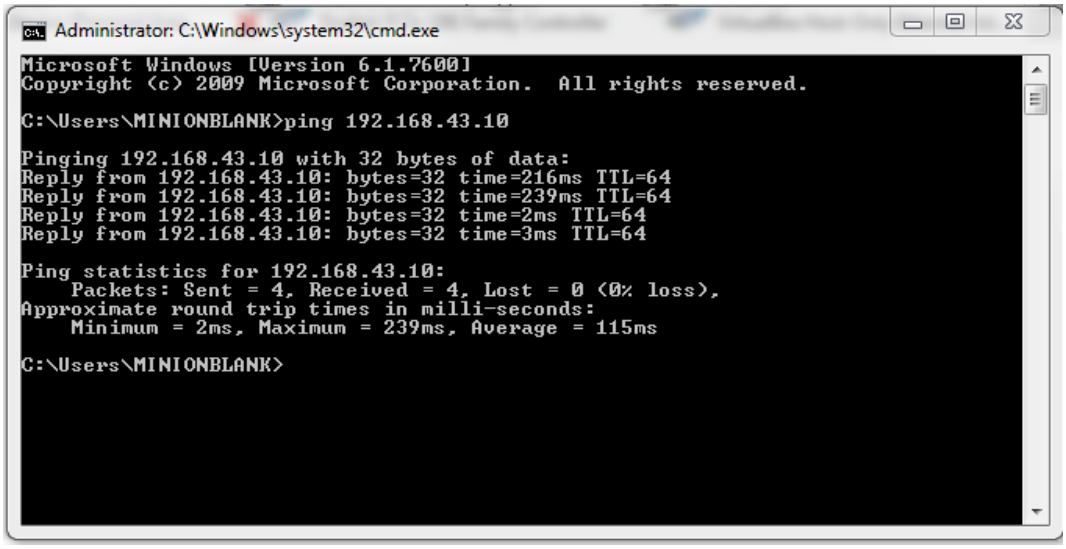
1. Pengujian Menggunakan Kebijakan *ACCEPT*

Pada kebijakan ini, semua komputer *client* bisa melakukan pengiriman paket *ping* ke *server,* tapi setelah di konfigurasi pada *file rules* yaitu pada bagian ***Ping(ACCEPT) loc:192.168.43.77 $FW*** maka hanya komputer *client* yang memiliki *ip address* 192.168.43.77 yang bisa mengirim paket *ping* ke *server* (*firewall*).



Gambar 4.15 Script ping accept

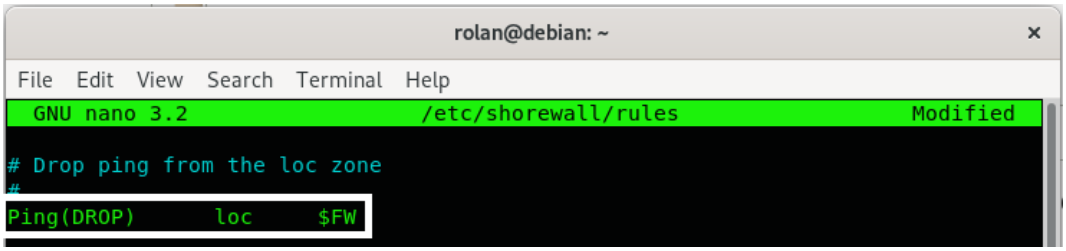
Pada gambar 4.15 menunjukkan kebijakan *ACCEPT*, yaitu pada kebijakan ini komputer *client* yang memiliki *ip address* 192.168.43.77 bisa mengirimkan paket *ping* ke *server* (*firewall*) dan untuk menambah komputer *client* yang ingin diberi izin untuk mengirim paket *ping*, tinggal menambahkan di baris bawah konfigurasi sebelumnya pada *file rules* yaitu pada bagian ***Ping(ACCEPT) loc: $FW*****,*** untuk bagian *loc* masukkan *ip address* komputer *client* yang ingin di beri izin akses atau melakukan pengiriman paket *ping*. Hasilnya bisa di lihat pada gambar 4.16.



Gambar 4.16 Hasil pengiriman paket ping yang di terima

2. Pengujian menggunakan kebijakan *DROP*

Pada kebijakan ini, semua komputer *client* tidak bisa melakukan pengiriman paket *ping* ke *server*, karena *firewall* akan langsung ”membuang” setiap paket yang memiliki target ini tanpa mengirim pesan *error* kepada pengirim paket tersebut.



Gambar 4.17 Script ping drop

Pada gambar 4.17 menunjukkan kebijakan *DROP*, yaitu pada kebijakan ini semua komputer *client* tidak bisa mengirimkan paket *ping* ke *server* (*feriwall*). dan hasilnya bisa di lihat pada gambar 4.18.

```
C:\Users\DELL>ping 192.168.43.10

Pinging 192.168.43.10 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.43.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\DELL>
```

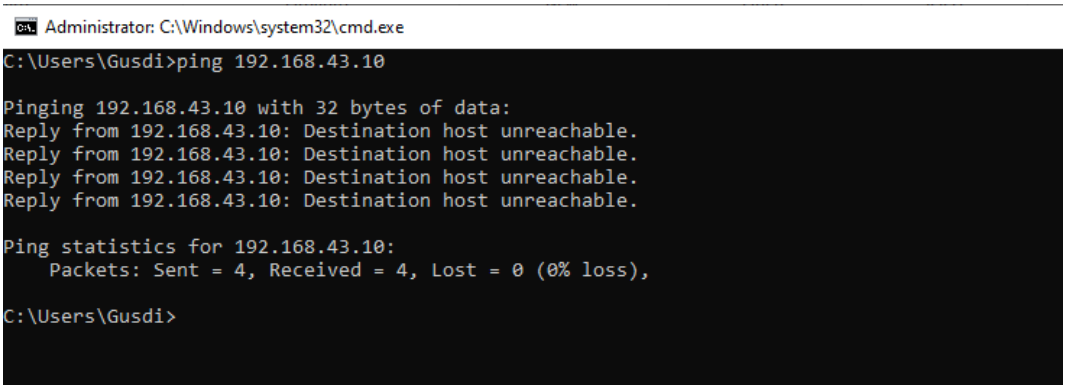
Gambar 4.18 Hasil pengiriman paket ping yang di tolak

3. Pengujian menggunakan kebijakan *REJECT*

Sama dengan *DROP*, semua komputer *client* juga tidak bisa melakukan pengiriman paket *ping* ke *server* tetapi, setiap paket yang memiliki kebijakan *reject* ini *firewall* akan mengirimkan pesan *ICMP error* kepada si pengirim paket. Untuk melakukan konfigurasi *file rules* nya seperti pada gambar 4.19.

Gambar 4.19 Script ping reject

Pada gambar 4.19 menunjukkan kebijakan *REJECT*, yaitu sama seperti *DROP* pada kebijakan ini komputer *client* tidak bisa mengirimkan paket *ping* ke *server* (*firewall*), bedanya kalau kebijakan ini akan di kirimkan pesan *error* kepada komputer *client.* Secara *default, firewall* akan mengirimkan pesan *ICMP* berupa *port-unreachable*. Hasilnya bisa di lihat pada gambar 4.20

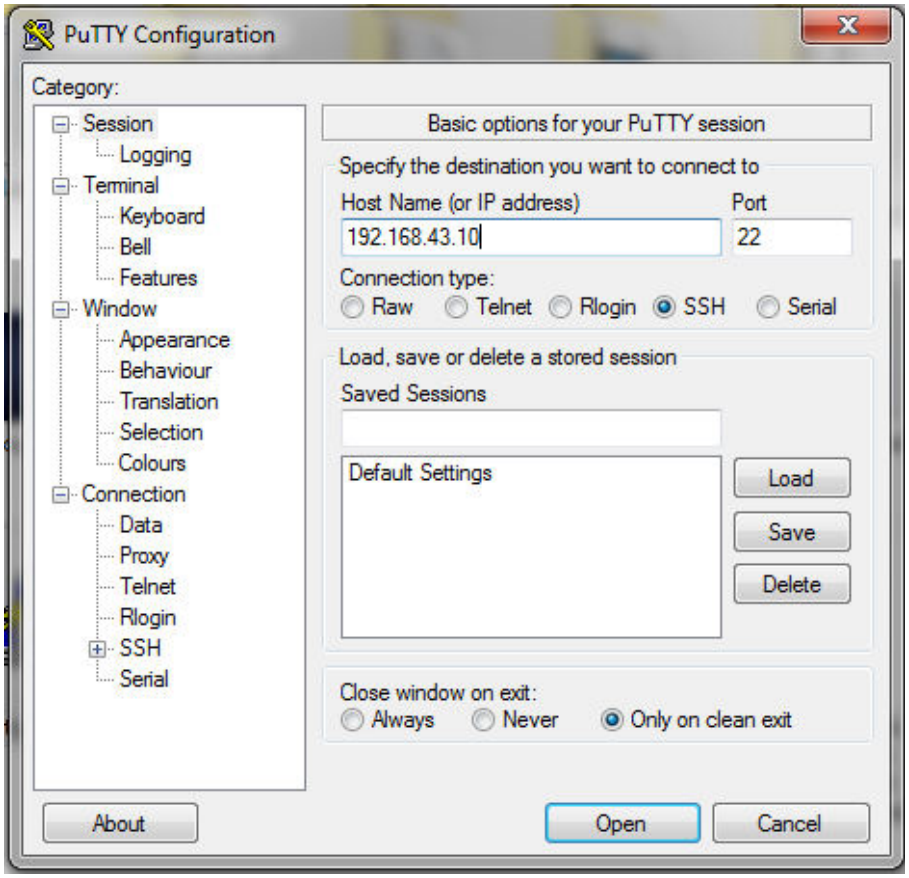


Gambar 4.20 Hasil pengiriman paket ping yang di reject

Tabel 4.1 Hasil Pengujian Pengiriman Paket *Ping*

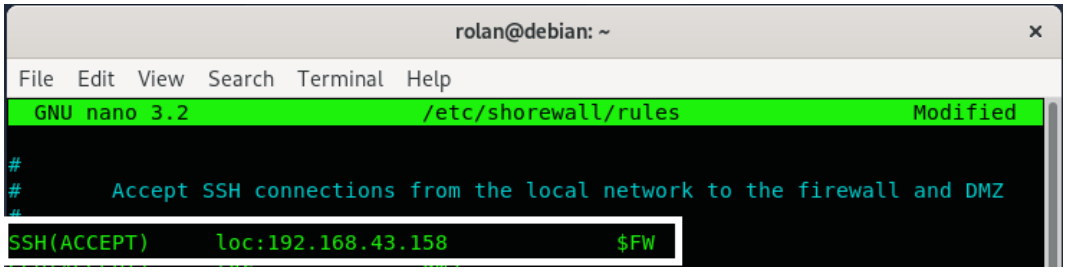
Komputer	*IP Address*	Pengujian yang di lakukan	Kebijakan	Keterangan
Server (Firewall)	192.168.43.10	_____	_____	_____
Client	192.168.43.77	Pengujian pengiriman paket *ping* ke komputer *server*	*ACCEPT*	Berhasil (*reply*)
Client	192.168.43.73			
Client	192.168.43.158	Pengujian pengiriman *ping* ke *server*	*DROP*	Gagal (*koneksi time out*)
Client	192.168.43.118	Pengujian pengiriman *ping* ke *server*	*REJECT*	Gagal (*destination host unreachable*)

4.2.2. *Remote Dekstop SSH* Menggunakan *Putty*



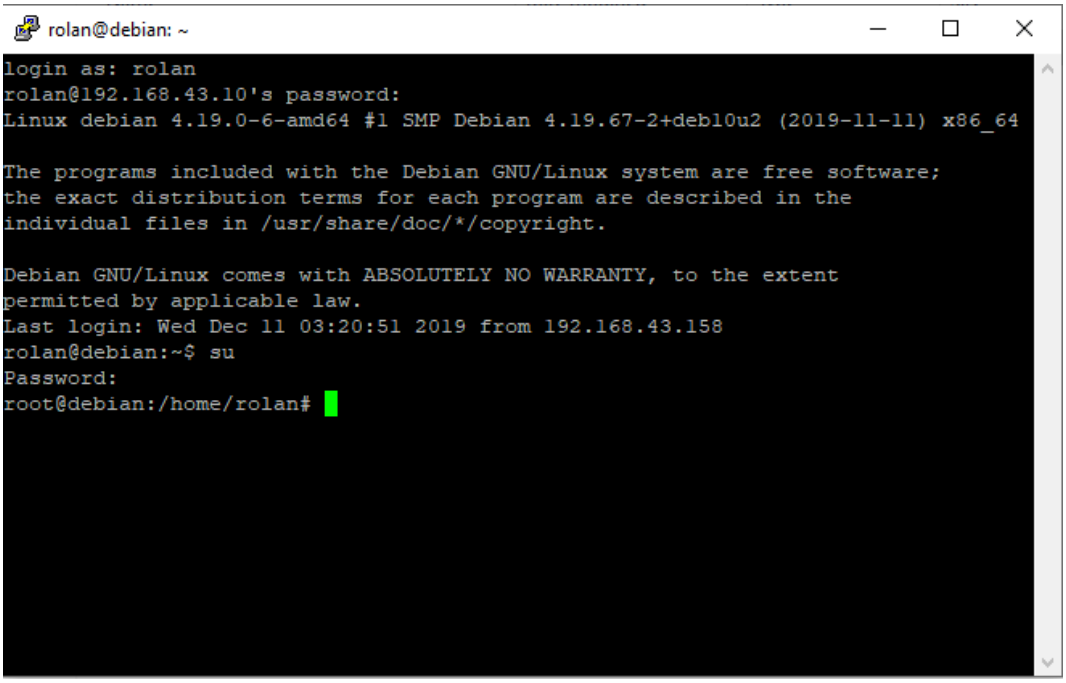
Gambar 4.21 Remote dekstop dengan SSH Putty

Pada pengujian ini, kebijakan yang akan di gunakan yaitu *ACCEPT* dan *DROP* dengan konfigurasi pada *file rules* seperti pada gambar 4.22.



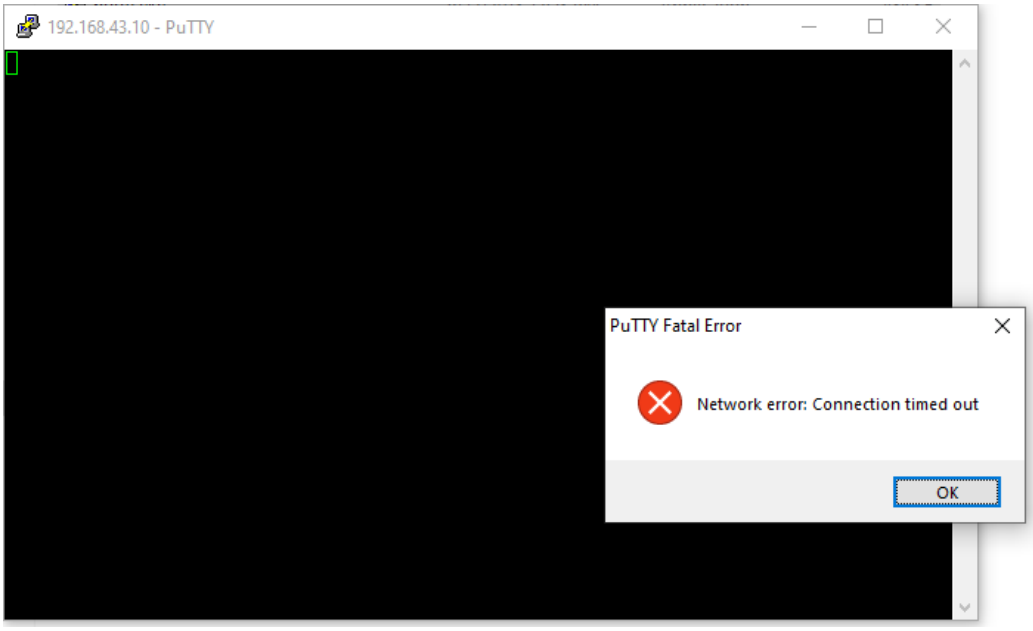
Gambar 4.22 Script ssh accept

Pada gambar 4.22 merupakan konfigurasi *SSH*, pada konfigurasi ini hanya komputer *client* yang memiliki *ip address* 192.168.43.158 yang bisa melakukan *remote dekstop* ke *server*, selain dari *ip addreess* itu tidak bisa melakukan *remote dekstop* tersebut.



Gambar 4.23 Proses remote dekstop yang di terima

Selanjutnya adalah komputer *client* yang di *DROP* oleh *shorewall* pada gambar 4.24.

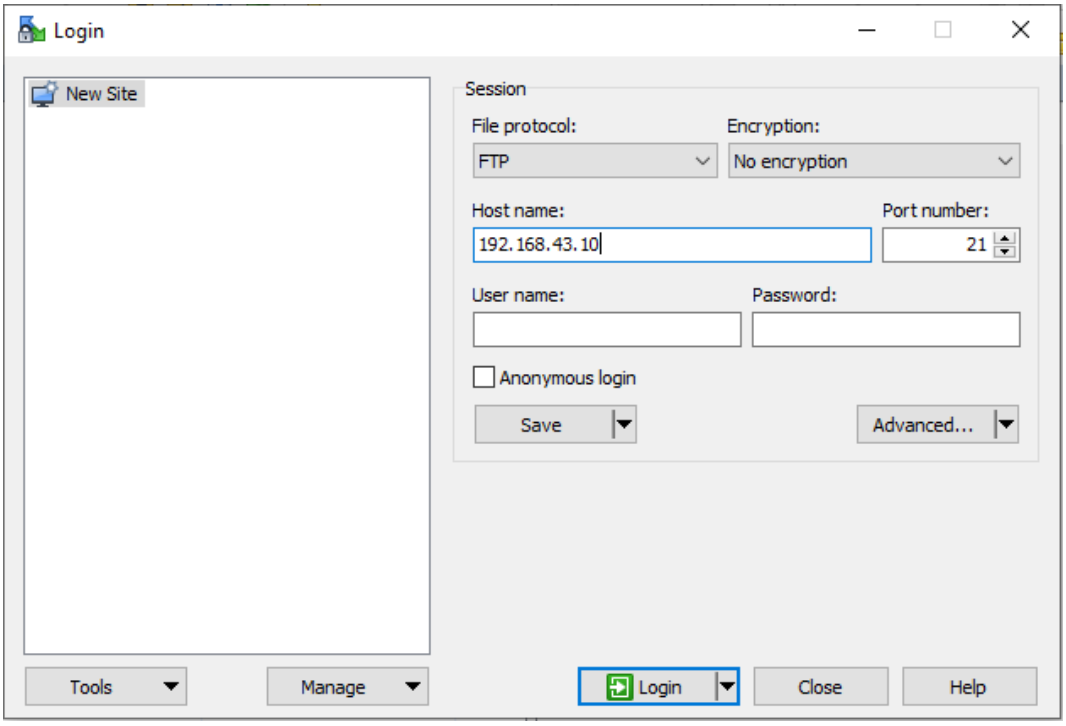


Gambar 4.24 Proses remote dekstop yang di tolak

Tabel 4.2 Hasil pengujian *remote dekstop ssh* menggunakan *putty*

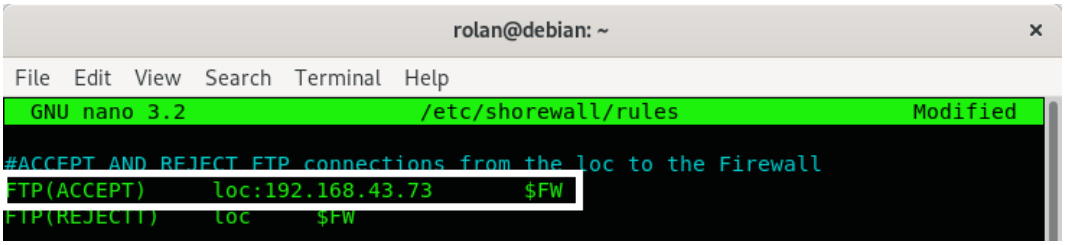
Komputer	*IP Address*	Pengujian yang di lakukan	Kebijakan	Keterangan
Server (Firewall)	192.168.43.10	_____	_____	_____
Client	192.168.43.158	Pengujian *remote dekstop ssh* ke komputer *server*	*ACCEPT*	Berhasil (dapat melakukan *remote dekstop*)
Client	192.168.43.77	Pengujian *remote dekstop ssh* ke komputer *server*	*DROP*	Gagal (tidak bisa melakukan *remote dekstop*)

4.2.3. *File Transfer Protocol* Menggunakan *WinSCP*



Gambar 4.25 Pengujian ftp menggunakan winscp

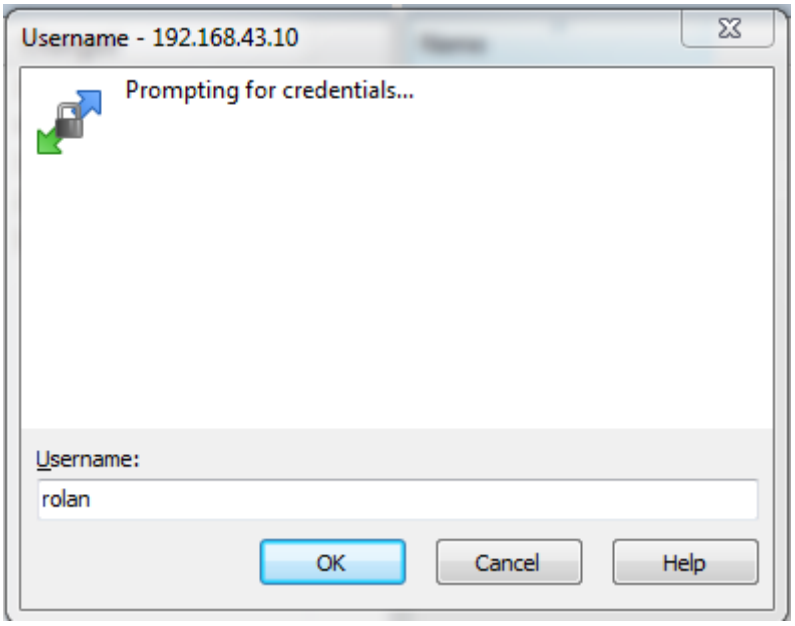
Pada pengujian ini, kebijakan yang akan di gunakan yaitu *ACCEPT* dan *REJECT* dengan konfigurasi pada *file rules* yaitu di bagian ***FTP(ACCEPT) loc: $FW*** seperti pada gambar 4.26.



Gambar 4.26 Script ftp accept dan reject

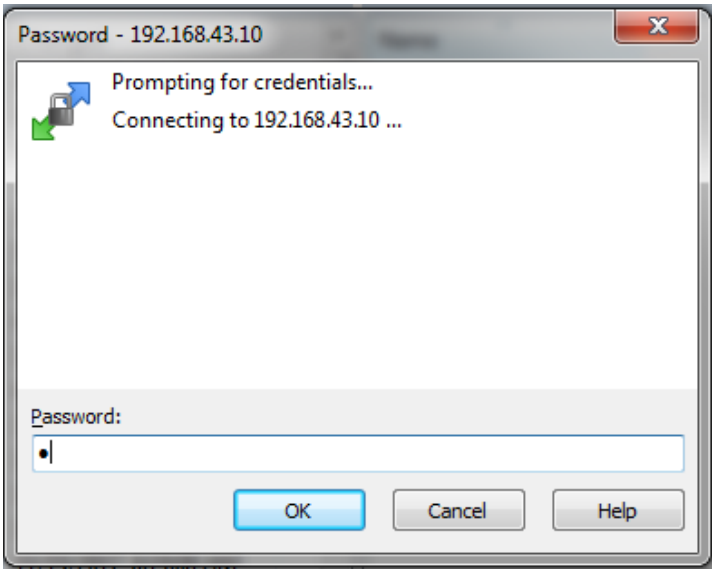
Pada gambar 4.26 merupakan konfigurasi *FTP*, pada konfigurasi ini hanya komputer *client* yang memiliki *ip address* 192.168.43.73 yang bisa melakukan *transfer file* ke *server*, selain dari *ip address* itu tidak bisa melakukan *transfer file* tersebut.

Memasukkan nama *user* komputer *server(firewall)*. Proses tersebut bisa di lihat pada gambar 4.27.



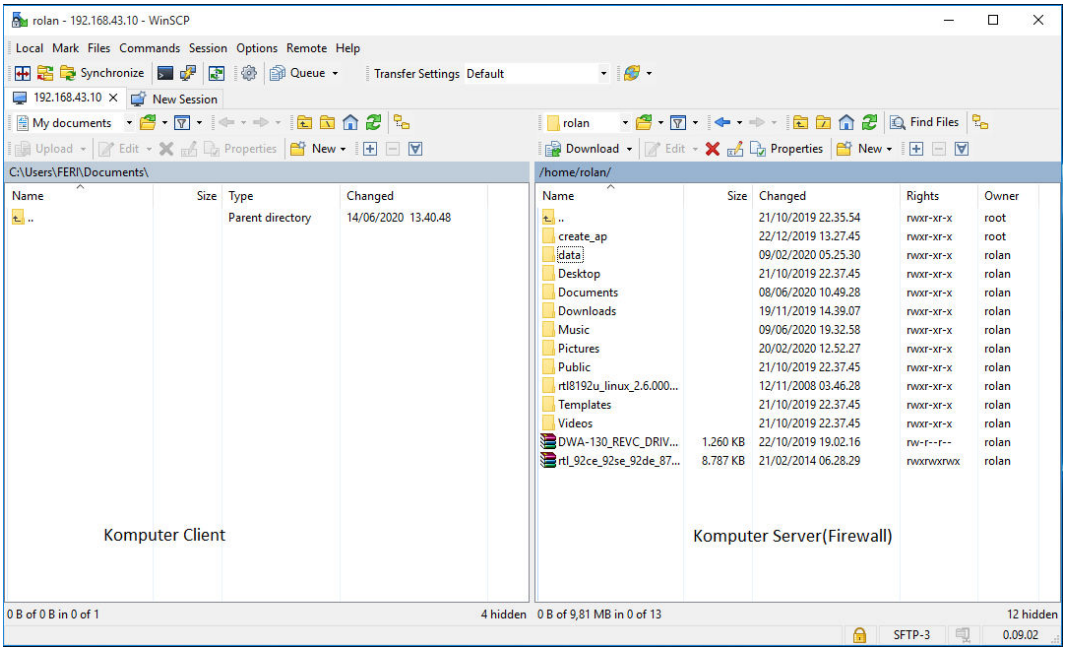
4.27 Proses memasukkan username komputer server

Selanjutnya memasukkan *password* komputer *server(firewall)*. Proses tersebut bisa di lihat pada gambar 4.28.



Gambar 4.28 Proses memasukkan password komputer server

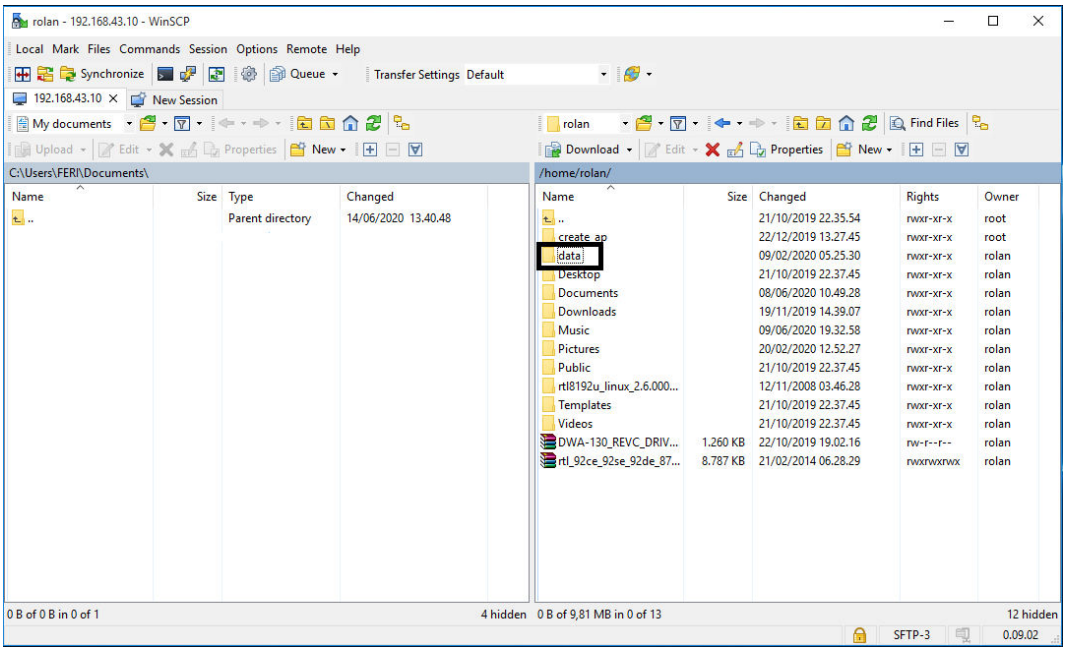
Berikut adalah proses ketika *client* berhasil *login* ke komputer *server(firewall)* melalui program *WinSCP*. Proses tersebut bisa di lihat pada gambar 4.29.



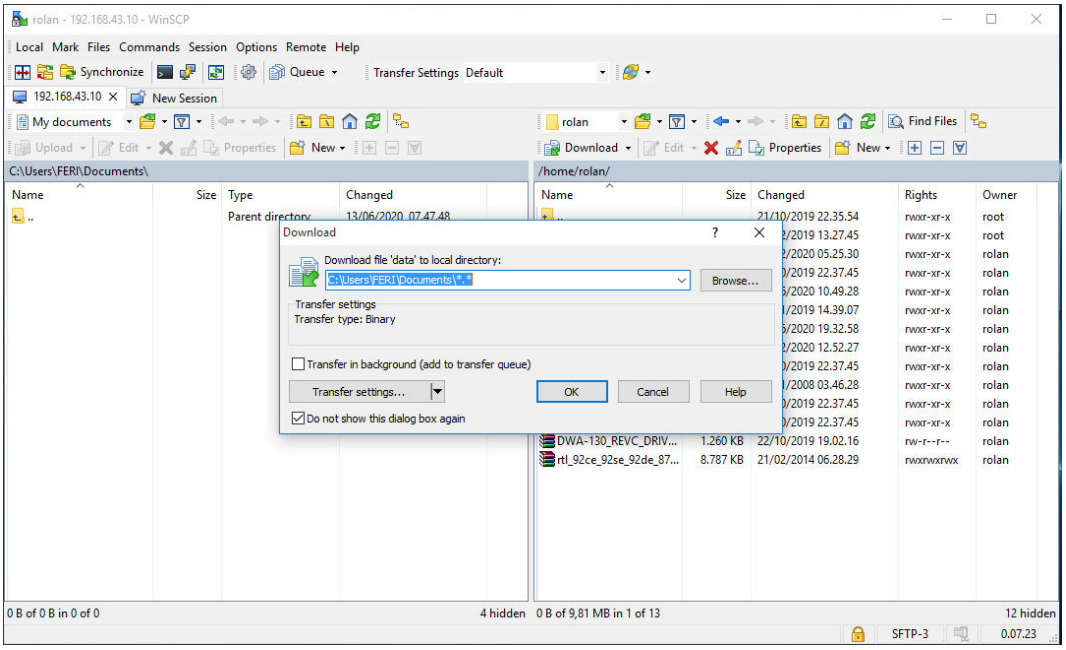
Gambar 4.29 Ftp yang di terima

Kemudian, komputer *client* akan melakukan *transfer file* dari komputer *server(firewall)*, *file* yang akan di *transfer* yaitu *folder* data yang terdapat pada

komputer *server(firewall)*. Proses tersebut bisa di lihat pada gambar 4.30 dan 4.31.

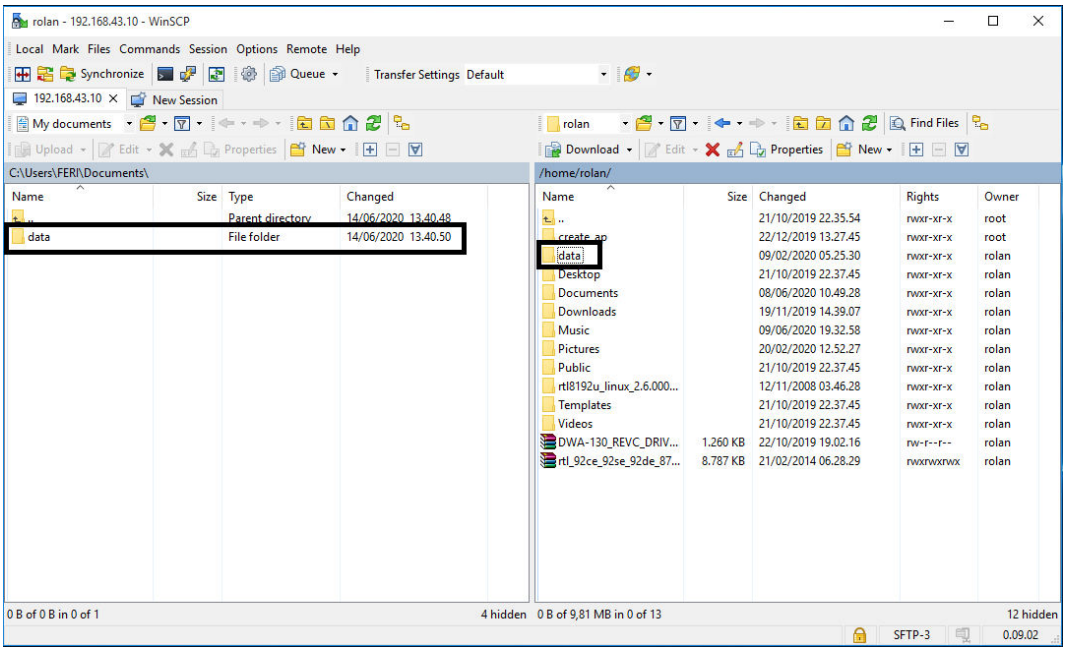


Gambar 4.30 File yang di transfer



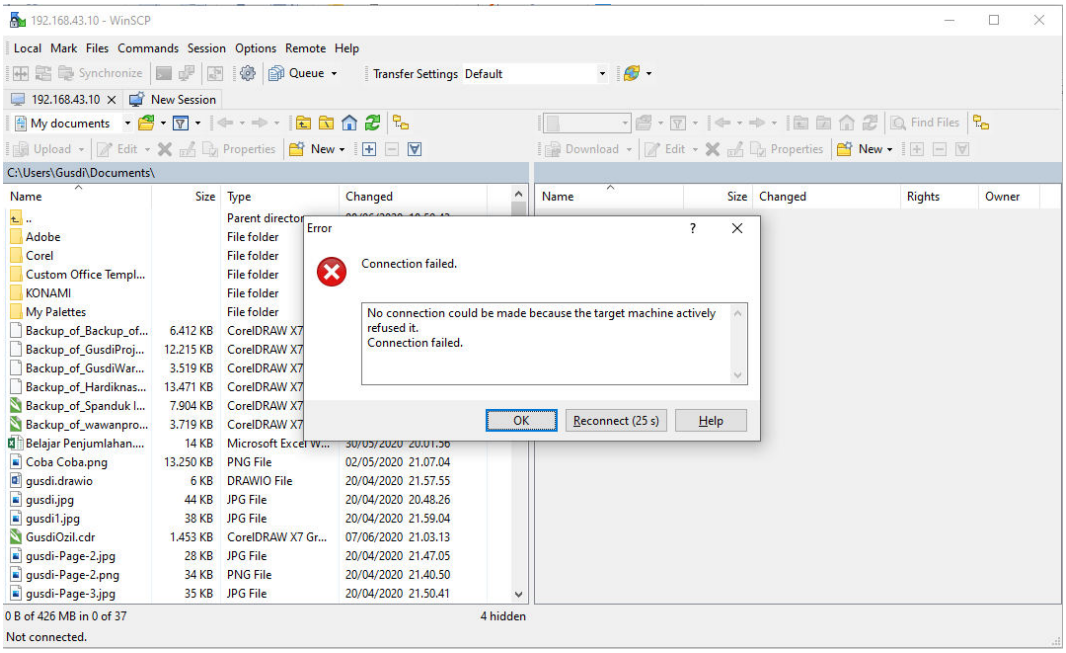
Gambar 4.31 Proses transfer file

Berikut adalah hasil *transfer file* yang berhasil di lakukan oleh komputer *client*. Hasil tersebut bisa di lihat pada gambar 4.32.



Gambar 4.32 Proses ftp yang berhasil di lakukan

Selanjutnya adalah komputer *client* yang di *REJECT* oleh *shorewall* seperti pada gambar 4.33.

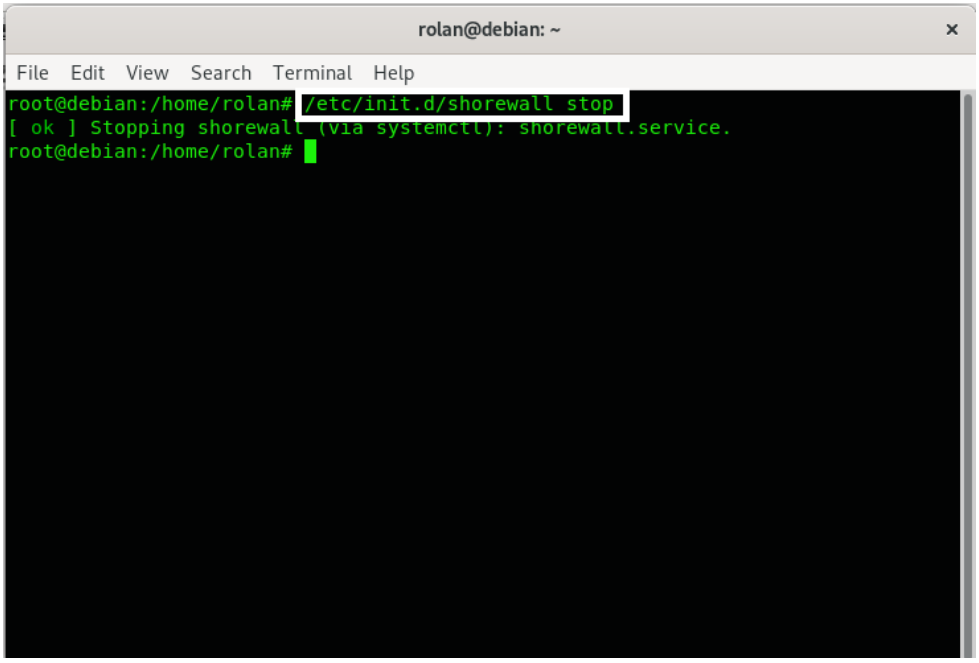


Gambar 4.33 Proses ftp yang di tolak

Tabel 4.3 Hasil pengujian *file transfer protocol* menggunakan *WinSCP*

Komputer	*IP Address*	Pengujian yang dilakukan	Kebijakan	Keterangan
Server (Firewall)	192.168.43.10	———————	———	———————
Client	192.168.43.73	Pengujian *transfer file* ke komputer *server*	*ACCEPT*	Berhasil (melakukan *transfer file*)
Client	192.168.43.118	Pengujian *transfer file* ke komputer *server*	*REJECT*	Gagal (tidak dapat melakukan *transfer file*)

Jika ingin menonaktifkan *firewall* (*shorewall*) ini gunakan perintah ***/etc/init.d/shorewall stop*** pada *terminal*. Seperti pada gambar 4.34.



Gambar 4.34 Menonaktifkan shorewall

BAB V
PENUTUP

5.1. Kesimpulan

Dari hasil dan pembahasan serta perancangan yang telah dibuat maka kesimpulannya sebagai berikut :

1. Pengujian *firewall* menggunakan *shorewall* di lakukan dalam ruang lingkup jaringan lokal.
2. Ada tiga pengujian yang dilakukan dalam penelitian ini, yaitu pengiriman paket *ping (ICMP)*, *remote dekstop SSH* menggunakan *Putty* dan *File Transfer Protocol (FTP)* menggunakan *WinSCP*.
3. Pada *firewall* (*shorewall*) ini memiliki perintah/*source code* yang perlu di jalankan untuk melakukan pengamanan terhadap komputer yang terhubung ke jaringan. Perintah tersebut di antaranya yaitu *ACCEPT*, *DROP*, dan *REJECT*.
4. Berdasarkan pengujian beberapa layanan *service Ping (ICMP), SSH* dan *FTP* tidak berfungsi jika *firewall* di aktifkan (tergantung dari konfigurasi sistem).
5. Sistem dasar yg digunakan oleh *shorewall* ini bersifat *open source*, dimana semua *code* untuk membangun sebuah sistem *firewall* bisa di modifikasi sesuai kebutuhan. Sedangkan jenis *firewall* yang lain belum tentu bisa di ubah *base code*nya. Seperti fungsi *firewall* di *Windows*.
6. *Firewall* menggunakan *Shorewall* dapat menjaga keamanan data di *Server* , *client* yang terhubung pada jaringan komputer.
7. Kelebihan dari *software shorewall* yaitu *shorewall* lebih mudah di konfigurasi dari pada jenis *firewall* lainnya yang ada di *Linux*.

5.2. Saran

Dari hasil penelitian dan pengamatan sebelumnya, maka penulis mengemukakan beberapa pendapat antara lain :

1. Sesudah melakukan konfigurasi pada *file rules*, *shorewall* harus di *restart* terlebih dahulu agar konfigurasi tersebut dapat berjalan dengan baik sesuai dengan kebutuhan sistem.
2. Perancangan *firewall* menggunakan *shorewall* ini semoga dapat terus di manfaatkan dan di kembangkan sehingga membawa hasil yang lebih baik kedepan.

DAFTAR PUSTAKA

Ardiansyah, Rino, 2017, Pengertian Ping (Packet Internet Groper).

Arifin, Hasnul. 2011. Kitab Suci Jaringan Komputer dan Koneksi Internet. Yogyakarta. Mediakom.

Aziz, Saiful dan Bambang Eka Purnama. 2012. Sistem Keamanan Jaringan Komputer Dengan Firewall dan Intrusion Detection System (IDS). Jurnal Speed 13, Volume 9. No. 2, Agustus 2012: 1-6.

Bastien, G; Degu, C.A. 2004. CCSP Secure Exam Certification Guide. Cisco Press. Indianapolis, USA.

Bowo, Eri. 2010. Ubuntu From Zero. Jasakom. Jakarta.

Ciccarelli, dan Faulkher. 2016. Wide Area Network. Bandung: CV Alfabeta

Dicky, Mhd. Syahputra Lubis dan Allwine. 2018. Membangun PCrouter Dengan Ubuntu Server dan Keamanan Jaringan Dengan Shorewall. STMIK Methodist Binjai Jl. Jend. Sudirman No. 136 Binjai 061-88742021 Teknik Informatika. Vol 2. No 1.

Dwi Cahyani, Ika.2010."Sistem Keamanan Enkripsi SSH untuk Keamananan", Yogyakarta: Library UGM.

Erlangga, Gery. 2011. The Primary Domain Controller In Networking Webrother. 22 Agustus 2015. http://library.gunadarma.ac.id.

Freeman. 2014. Jaringan Komputer. Yogyakarta: Pustaka Pelajar.

Garfinkel, S; Spafford, G; Schwartz, A. 2003. Practical UNIX and Internet.

Herlambang, Moch Linto, dan Aziz Catur L. 2008. Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik OS.Jakarta: Andi Publisher.

Herlambang, M. Dicki. 2016. “Rancang Bangun Keamanan Loker Dengan Autentifikasi Wajah Dan Password Berbasis Raspberry Pi Dengan

Notifikasi Keamanan Via Android". Jurusan Teknik Komputer Politeknik Negeri Sriwijaya Palembang.

Kadir, Abdul. (2013). Pengantar Teknologi Informasi. Yogyakarta: Andi Offset.

Nugroho, Bunafit. (2005). Instalasi & Konfigurasi Jaringan Windows dan Linux, Yogyakarta: ANDI

Nur Khasanah, Siti. 2016. Keamanan Jaringan Dengan Packet Filtering Firewall (Studi Kasus: Pt. Sukses Berkat Mandiri Jakarta). STMIK Nusa Mandiri Jakarta. Vol IV. No. 2.

Pradikta Reza, Achmad Affandi, Eko Setijadi. Rancang bangun aplikasi monitoring jaringan dengan menggunakan Simple Network Management Protocol. Jurnal Teknik Pomits. Volume 2. 2013.

Pratama, Putu Agus E. 2014. Handbook Jaringan Komputer. Bandung: Infomatika.

Prasetyo, Rizki Aris, 2010. "Analisis dan Perancangan Site to Site Virtual Private Netwok (VPN) Berbasis IP Security Menggunakan Mikrotik Router Operating System", Bandung : Universitas Komputer Indonesia.

Riadi, Imam. 2011. Optimalisasi Keamanan Jaringan Menggunakan Pemfilteran Aplikasi Berbasis Mikrotik. JUSI Volume 1, No. 1, Februari 2011: 71-80.

RICOH. 2010. Network Security White Paper.

Sopandi, Dede. 2010. Instalasi dan Konfigurasi Jaringan Komputer. Bandung : Informatika

Sujarwo, Ari. 2010. Sistem Operasi, Laboratorium Sistem dan Jaringan Komputer UII, Yogyakarta.

Sukmaaji, Anjik, Rianto. 2008, Jaringan Komputer : Konsep Dasar Pengembangan Jaringan Dan Keamanan Jaringan. Yogyakarta : Andi

Toibah Umi Kalsum, Siswanto dan Eko Prasetyo Rohmawan. 2013. Sistem Pengendalian Parkir Menggunakan Sensor Switch. Jurnal Media Infotama, Vol.9, No.2, September 2013.

Winarno, Edi, Ali Zaki dan SmitDev Community. 2010. Web Programming dengan Visual Basic 2010. Jakarta. PT. Elex Media Komputindo.

Yudianto, M. Jafar Noor. 2007. Jaringan Komputer dan Pengertiannya. Ilmukomputer.com.