BAB II
LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1. Keamanan Jaringan

Menurut Ariyus (2007) yang dikutip dari jurnal penetitian yang berjudul perancangan keamanan jaringan komputer menggunakan Snort dengan notifikasi SMS yang ditulis oleh Teguh Wahyudi dan Rissal Efendi (2015) menjelaskan bahwa keamanan jaringan secara umum komputer yang terhubung dalam sebuah jaringan komputer memiliki ancaman keamanan lebih besar daripada komputer yang tidak terhubung dalam sebuah jaringan. Keamanan jaringan menjadi tantangan tersendiri dalam sebuah jaringan komputer karena *network security* bertolak belakang dengan *network access*, dimana *network access* semakin mudah maka *network security* akan semakin rawan, dan bila *network security* semakin baik maka *network access* semakin sulit.

Dalam keamanan jaringan terdapat berbagai bentuk ancaman baik fisik maupun *logic* yang secara langsung maupun tidak langsung mengganggu kegiatan yang sedang berlangsung di dalam jaringan. Resiko dalam jaringan komputer disebabkan oleh beberapa faktor, yaitu :

1) Kelemahan manusia.
2) Kelemahan perangkat komputer.
3) Kelemahan sistem operasi jaringan.
4) Kelemahan sistem jaringan komunikasi

Sedangkan tujuan khusus membuat keamanan yang lebih baik di antaranya:

1) *Confidentiality*

   Adanya data-data penting yang tidak dapat diakses oleh semua *user*, maka dilakukan usaha untuk menjaga informasi dari *user* yang tidak mempunyai akses tersebut. Biasanya *confidentiality*, ini berhubungan dengan informasi yang diberikan kepada pihak lain.

2) *Integrity*

Pesan yang dikirim dengan pesan yang diterima masih orisinil yang tidak diragukan keasliannya, tidak dimodifikasi selama perjalanan dari sumber kepada penerima.

3) *Availability*

Dimana user diberikan hak akses tepat pada waktunya, biasanya ini berhubungan dengan ketersediaan infomasi atau data ketika dibutuhkan, tidak hanya melindungi jaringan tetapi dapat bertindak apabila terjadi serangan yang ada di dalam jaringan. Salah satu metode tersebut yaitu *Intrusion Detection System* (IDS).

Metode tersebut membutuhkan suatu pemahaman untuk menentukan kebijakan keamanan (*security policy*) dalam keamanan jaringan. Jika ingin menentukan apa saja yang harus dilindungi maka harus mempunyai perencanaan keamanan yang matang dan baik berdasarkan pada prosedur dan kebijakan keamanan jaringan, karena apabila tidak direncanakan maka tidak akan sesuai dengan yang diharapkan dalam keamanan jaringan.

Kesimpulannya dari keamanan jaringan adalah sebuah aktivitas untuk melakukan pengamanan dari sebuah penyalahgunaan atau gangguan, serta meakukan perlindungan di sebuah jaringan.

2.1.2. *Intrusion Detection System* (IDS)

Konsep dasar Intrusion Detection System (IDS) adalah cara mendeteksi penyusup atau pengganggu melalui jaringan dan masuk ke sistem jaringan yang bersifat perusak (*attacker*) atau seorang pengguna yang sah tetapi menyalahgunakan hak akses. Sehingga IDS ini mampu mengawasi jika terjadi penyerangan sistem jaringan melalui *traffic* yang terjadi pada jaringan dengan mendeteksi perubahan grafik ataupun besaran paket data yang lewat dari alamat IP pengguna. Jika ditemukan, sebuah peringatan akan dicatat dan respon yang diberikan berdasarkan data yang telah dicatat (Purbo, 2010).

IDS adalah metode perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. IDS

dapat melakukan inspeksi terhadap lalu-lintas *inbound* dan *outbound* dalam sebuah sistem atau jaringan, melakukan analisis dan mencari bukti dari percobaan *intrusion* (Ariyus,2007).

Jenis IDS dapat dibedakan menjadi 2 jenis, yaitu *Host-based Intrusion Detection System* (HIDS) dan *Network-based Intrusion Detection System* (NIDS). *Intrusion Detection System* (IDS) terkenal dan dipergunakan secara luas sebagai perangkat keamanan yang digunakan untuk mendeteksi serangan dan aktivitas mencurigakan di dalam sebuah jaringan. *Intrusion Detection System* (IDS) merupakan sebuah elemen penting pada keamanan jaringan. Terdapat 2 macam teknik untuk mendeteksi serangan, yaitu *signature-based detection* dan *anomaly-based detection*. Kedua teknik tersebut memiliki keunggulan dan kekurang masing-masing. Arsitektur dari *Intrusion Detection System* (IDS) dan teknik yang dipakai berdampak besar pada hasil kerja dari *Intrusion Detection System* (IDS) itu sendiri. Program yang dipergunakan biasanya disebut sebagai *Intrusion Detection System* (IDS).

IDS juga dapat digunakan untuk memonitori lalu lintas jaringan, sehingga mendeteksi jika sistem sedang ditargetkan oleh serangan jaringan

Menurut Darapareddy dan Gummadi (2012) terdapat dua jenis dasar deteksi intrusi yaitu berbasis *host* (HIDS) dan berbasis *network* (NIDS) sedangkan tedapat dua jenis metode analisis *event* pada IDS yaitu *Signature Based* dan *Anomaly Based.*

2.1.2.1. Jenis-jenis IDS

2.1.2.1.1. *Host-based Intrusion Detection System* (HIDS)

HIDS terletak di sistem dan dapat mengamati semua aktivitas dari *host*. HIDS akan mencatat semua aktivitas yang ditemukan dan melakukan pengecekan atau menganalisa aktivitas pada komputer tertentu dan mencari tanda-tanda serangan pada komputer tersebut. HIDS melakukan pengawasan terhadap paket-paket atau aktivitas sebuah *host* apakah terjadi percobaan serangan atau penyusupan dalam jaringan atau tidak.

2.1.2.1.2. *Network-based Intrusion Detection System* (NIDS)

Network-based Intrusion Detection System (NIDS) merupakan jenis IDS yang paling umum dan sering digunakan dalam sebuah jaringan. Mekanisme ini mendeteksi serangan dengan menangkap dan menganalisa paket-paket jaringan. NIDS biasanya ditempatkan pada sebuah titik pusat atau tempat yang strategis di dalam sebuah jaringan untuk melakukan pengawasan terhadap *traffic* yang menuju dan berasal dari semua perangkat (*device*) dalam jaringan.

2.1.2.2. Metode Analisis *Event* IDS

2.1.2.2.1. *Signature Based*

Signature based menggunakan pendekatan dengan cara pencocokan kejadian (*event*) dengan jenis serangan yang telah dikenal pada *database* IDS. Teknik ini sangat efektif dan merupakan metode utama yang digunakan pada beberapa perangkat atau produk IDS untuk mendeteksi serangan.

2.1.2.2.2. *Anomaly Based*

Anomaly based menggunakan pendekatan dengan cara mengidentifikasi perilaku atau aktivitas yang tidak biasa yang terjadi pada suatu *host* atau jaringan. *Anomaly based* membentuk perilaku dasar pada sebuah kondisi jaringan normal dengan profil pengguna tertentu kemudian mengukur dan membandingkannya ketika aktivitas jaringan berjalan tidak normal.

2.1.2.3. Cara Kerja IDS

Intrusion detection system dapat berupa perangkat lunak atau perangkat keras yang melakukan otomatisasi proses monitoring kejadian yang terjadi pada sebuah jaringan. IDS dibuat bukan untuk menggantikan fungsi *firewall* karena memiliki tugas yang berbeda. IDS adalah pemberi sinyal pertama jika terjadi serangan atau adanya penysup dalam jaringan.

Kesimpulan dari *Intrusion Detecsion System* (IDS) adalah IDS merupakan sebuah bentuk atau metode pengamanan jaringan yang menggunakan teknik pendeteksian dari serangan atau gangguan pada sebuah jaringan.

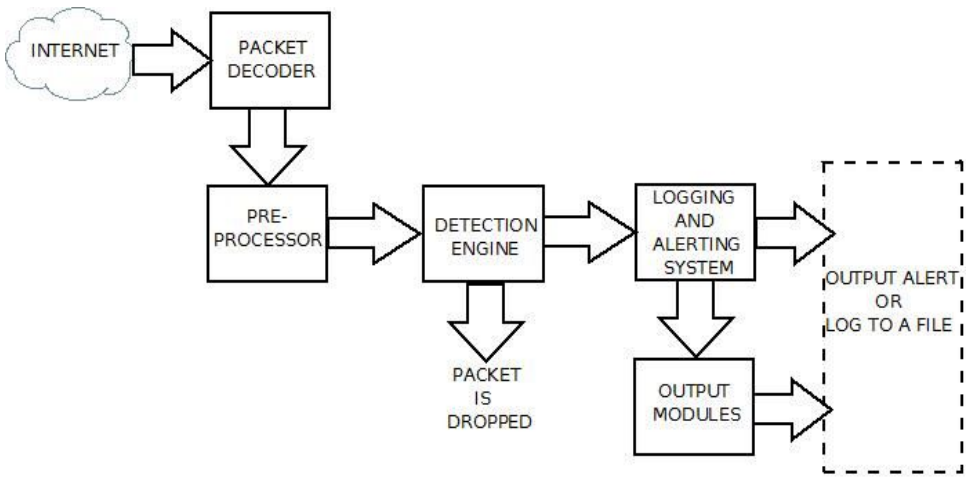
2.1.3 Snort

Snort (www.snort.org) adalah salah satu *tool* atau aplikasi *open source* dari *Intrusion Detection System* (IDS) yang terbaik yang tersedia dan dikembangkan hingga saat ini. Snortdirancang untuk beroperasi berbasis *command line* dan telah diintegrasikan ke beberapa aplikasi pihak ketiga dan mendukung *cross platform*. Snortmenganalisis semua lalu lintas jaringan untuk mengendus (*sniff*) dan mencari beberapa jenis penyusupan dalam sebuah jaringan.

Snortmerupakan *software* yang masih berbasis *command-line*, sehingga cukup merepotkan bagi pengguna yang sudah terbiasa dalam lingkungan *Graphical UserInterface* (GUI). Oleh karena itu, ada beberapa *software* pihak ketiga yang memberikan GUI untuk Snort, 4 misalnya *IDS Center* untuk *Microsoft Windows*, dan *Acid* yang berbasis PHP sehingga bisa diakses melalui *web browser* (Pentrani dan Dwiarso, 2014).

2.1.3.1 Komponen Snort

Snortdibagi ke dalam beberapa komponen yang memiliki kegunaan yang sangat penting. Komponen ini bekerjasama untuk mendeteksi serangan yang berbeda dan untuk menghasilkan keluaran pada *tipe* yang diinginkan pada sistem deteksi. IDS berbasiskan Snortumumnya terdiri dari komponen beberapa komponen (Ariyus, 2007:146), lihat pada Gambar 2.1.



Gambar 2.1 Komponen Snort

1) Dekoder paket

Dekoder paket mengambil paket dari beberapa jenis jaringan yang berbeda dari antarmuka jaringan dan mempersiapkan paket untuk di proses atau di kirim menuju *detection engine*. Antarmukanya dapat berupa *Ethernet*, SLIP, PPP dan yang lain.

2) Preprocessor

Preprocessor merupakan komponen atau *plug-ins* yang dapat digunakan pada Snortuntuk menyusun atau mengubah paket data sebelum *detection engine* melakukan beberapa operasi untuk mencari tahu jika paket digunakan oleh penyusup. *Preprocessor* pada Snortdapat men*dekode*-kan URL HTTP, melakukan *defragmentasi* paket, menggabungkan kembali aliran TCP dan yang lain.

3) *Detection Engine*

Detection engine adalah bagian terpenting dari Snort. Tugasnya adalah untuk mendeteksi jika terjadi aktifitas penyusup pada paket. *Detection engine* mempekerjakan *rules* Snortuntuk tujuan ini. *Rules* dibaca ke dalam struktur atau rantai data *internal* kemudian dicocokkan dengan paket yang ada. Jika paket sesuai dengan *rules* yang ada, tindakan akan diambil, jika tidak paket akan dibuang. Tindakan yang diambil dapat berupa *logging* paket atau mengaktifkan *alert*.

Detection engine merupakan bagian dari Snortyang sangat bergantung pada waktu tanggap. Waktu tanggap merupakan waktu yang dibutuhkan untuk merespon paket, hal ini bergantung pada seberapa bagus *server* yang ada dan seberapa banyak rules yang telah didefinisikan.

Berikut yang mempengaruhi waktu tanggap pada *detection engine* :

a) Jumlah *rules.*
b) Kekuatan mesin pada sistem.
c) Kecepatan *bus internal server* Snort.
d) Beban pada jaringan.

Ketika mendesain sistem keamanan komputer berbasiskan NIDS (*Network Intrusion Detection System*), faktor ini harus dipertimbangkan.

Catatan bahwa sistem deteksi dapat membelah paket dan menerapkan *rules* pada bagian paket yang berbeda. Bagian tersebut diantaranya:

a) *Packet payload*. Hal ini berarti dapat diciptakan *rules* yang digunakan oleh *detection engine* untuk mencari *string* di dalam data yang ada pada paket.

b) *Header layer transport*. *Header* ini dapat berupa TCP, UDP atau *header layer ransport* lainnya. Hal ini juga dapat bekerja pada *header* ICMP.

c) *Header level layer* aplikasi. *Header* layer aplikasi termasuk *header* DNS, FTP, SNMP dan SMTP, tidak dibatasi hanya *header* ini saja.

Detection engine bekerja dengan cara yang berbeda tergantung versi dari Snort. Pada versi 1.x *detection engine* akan berhenti bekerja jika paket telah sesuai dengan *rules*. *Detection engine* mengambil tindakan yang berbeda dengan me-*log* paket atau mengaktifkan *alert*, tergantung pada *rule*s. Hal ini paket harus sesuai dengan kriteria beberapa *rules*, hanya *rules* pertama yang akan diterapkan pada paket dan *rules* yang lain tidak akan dihiraukan. Penanganan jenis ini baik, kecuali pada satu permasalahan. *Rules* dengan prioritas rendah mengaktifkan *alert* dengan prioritas rendah, bahkan dengan *rules* berprioritas tinggi akan menghasilkan *alert* berprioritas tinggi akan ditempatkan kemudian pada rantai *rules*. Permasalahan ini kemudian diralat pada Snortversi 2 dimana semua *rules* telah dicocokkan dengan paket sebelum mengaktifkan *alert*. Setelah mencocokkan semua *rules*, *rules* berprioritas tertinggi memilih untuk mengaktifkan *alert*.

4) Sistem *Log* dan *Alert*

Berdasarkan apa yang ditemukan *detection engine* pada paket, paket dapat digunakan untuk me-*log* kegiatan atau mengaktifkan *alert*, bergantung pada apa yang ditemukan *detection engine* pada paket. *Log* di simpan pada format file teks sederhana, file berjenis *tcpdump* atau bentuk yang lain. File *log* disimpan di direktori /var/log/Snortsecara *default*. Perintah Snort-l pada *command line* dapat digunakan untuk memodifikasi lokasi dari *log* dan *alert* yang dihasilkan.

5) Modul *Output*

Modul *Output* dapat melakukan beberapa operasi berbeda tergantung bagaimana cara penyimpanan keluaran yang dihasilkan sistem *log* dan *alert* dari Snort. Pada dasarnya modul ini mengatur jenis keluaran yang dihasilkan oleh sistem *log* dan *alert*. Berdasarkan konfigurasi, keluaran *modul* dapat melakukan hal-hal berikut :

a) *Logging* ke dalam file /var/log/Snort/alerts atau file lainnya.
b) Mengirimkan *traps* SNMP.
c) Mengirimkan pesan ke *syslog.*
d) *Logging* ke dalam *database* seperti *MySQL* atau *Oracle.*
e) Menghasilkan *output extensible Markup Language* (XML).
f) Mengubah konfigurasi pada *router* atau *firewall*.
g) Mengirimkan pesan *Server Messager Block* (SMB) kepada mesin berbasiskan *Microsoft Windows.*

Snortmemiliki 3 buah mode, yaitu :

1) *Sniffer mode*, untuk melihat paket yang lewat dijaringan.
2) *Packet logger mode*, untuk mencatat semua paket yang lewat di jaringan untuk dianalisa dikemudian hari.
3) *Intrusion Detection Mode*, pada mode ini Snortakan berfungsi untuk mendeteksi serangan yang dilakukan melalui jaringan komputer

2.1.3.2. *Sniffer Mode*

Untuk menjalankan Snortpada *sniffer mode* dengan contoh perintah *sniffer mode* pada Tabel 2.1.

Tabel 2.1 Perintah *Sniffer Mode*

No	Perintah	Fungsi
1	#Snort–v	Melihat *header* TCP/IP paket yang lewat
2	#Snort–vd	Melihat isi paket
3	#Snort–vde	Melihat header link layer paket seperti *ethernet header*
4	#Snort–v –d –e	Melihat header TCP/IP,isi paket dan *header link* secara bersamaan

2.1.3.3. *Packet Logger Mode*

Tentunya cukup melelahkan untuk melihat paket yang lewat sedemikian cepat di layar terutama jika kita menggunakan *ethernet* berkecepatan 100Mbps, layar akan *scrolling* dengan cepat sekali susah untuk melihat paket yang di inginkan. Cara paling sederhana untuk mengatasi hal ini adalah menyimpan dulu semua paket yang lewat ke sebuah *file* untuk di lihat. Beberapa perintah *Packet Logger Mode* yang mungkin dapat digunakan untuk mencatat seperti Tabel 2.2.

Tabel 2.2 *Packet Logger Mode*

No	Perintah	Fungsi
1	./Snort–dev –l ./log	Me-*log* paket yang lewat
2	./Snort–dev –l ./log –h 192.168.0.0/24	Me-log host 192.168.0.0/24
3	./Snort–dev –l ./log –b	File yang di log dalam *format binary*
4	./Snort–dv –r packet.log	Membaca file yang sudah di-*log*

Perintah yang paling penting untuk me-*log* paket yang lewat adalah -l ./log yang menentukan bahwa paket yang lewat akan di *log* / di catat ke file ./log. Beberapa perintah tambahan dapat digunakan seperti –h 192.168.0.0/24 yang menunjukan bahwa yang di catat hanya paket dari *host* mana saja, dan –b yang memberitahukan agar *file* yang di *log* dalam *format binary*, bukan ASCII, Untuk membaca file *log* dapat dilakukan dengan menjalankan Snortdengan di tambahkan perintah –r nama file *log*, seperti pada Tabel 2.2.

2.1.3.4. *Intrusion Detection Mode*

Intrusion detection mode adalah proses mendeteksi penggunaan yang tidak sah, atau serangan terhadap suatu jaringan komputer. *Intrusion Detection System* (IDS) dirancang dan digunakan untuk membantu dalam menghalangi atau mengurangi ancaman, kerusakan yang dapat ditimbulkan dari aktivitas *hacking*. IDS merupakan kombinasi perangkat lunak atau perangkat keras yang dapat melakukan deteksi penyusupan pada sebuah jaringan. IDS dapat mendeteksi

adanya upaya yang membahayakan menyangkut kerahasiaan, keaslian, dan ketersediaan data pada sebuah jaringan komputer. Serangan bisa berasal dari luar sistem, orang dalam yang menyalahgunakan hak akses yang diberikan, dan orang tidak berwenang yang mencoba mendapatkan hak akses. IDS tidak bisa digunakan secara terpisah, tetapi harus menjadi bagian dari perencanaan dan kerangka langkah-langkah keamanan IT (Rian, 2014).

Mode operasi Snortyang paling rumit adalah sebagai pendeteksi penyusup (*intrusion detection)* di jaringan yang kita gunakan. Ciri khas mode operasi untuk pendeteksi penyusup adaah dengan menambahkan perintah ke Snortuntuk membaca file konfigurasi –c nama-file-konfigurasi.conf. Isi file konfigurasi ini sudah cukup banyak, tetapi sebagian besar telah di set secara baik dalam contoh Snort.conf yang dibawa oleh *source* Snort.

Intrusion Detection System (IDS) adalah sebuah aplikasi *software* atau *hardware* yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah jaringan. IDS dapat melakukan *inspeksi* terhadap lalu lintas *inbound* dan *outbound* dalam sebuah sistem atau jaringan, melakukan analisis dan mencari bukti dari percobaan *intrusion* (penyusupan) seperti perintah pada Tabel 2.3.

Tabel 2.3 Pencarian Penyusup

No	Perintah	Fungsi
1	./Snort–dev –l ./log –h 192.168.0.0/24 –c Snort.conf	Melakukan analisis dan mencari bukti dari percobaan intrusi (penyusupan) yang sudah di-*log* saat melewati *host* 192.168.0.0/24
2	./Snort–d –h 192.168.0.0/24 –l ./log –c Snort.conf	Melakukan analisis dan mencari bukti dari percobaan intrusi (penyusupan) yang melalui host 192.168.0.0/24

Untuk melakukan deteksi penyusup secara prinsip Snortharus melakukan *logging* paket yang lewat dapat menggunakan perintah –l nama-file-*logging*, atau membiarkan Snortmenggunakan *default* file *logging*-nya di *directory* /var/log/Snort. Kemudian menganalisa catatan / *logging* paket yang ada sesuai dengan isi perintah Snort.conf. Untuk mengirimkan *alert* ke *syslog* UNIX kita bisa menambahkan switch –s, seperti Tabel 2.4.

Tabel 2.4 Pengiriman *Alert* ke *Syslog*

No	Perintah	Fungsi
1	./Snort–c Snort.conf –l ./log –s –h 192.168.0.0/24	Pemberitahuan ke *syslog* ke host 192.168.0.0/24
2	./Snort–c Snort.conf –s –h 192.168.0.0/24	Pemberitahuan ke *syslog* ke host 192.168.0.0/24

Ada beberapa tambahan perintah yang akan membuat proses deteksi menjadi lebih effisien, mekanisme pemberitahuan *alert* di *Linux* dapat melakukan *seting* dengan perintah –A sebagai berikut, -A fast, mode *alert* yang cepat berisi waktu, berita, IP & *port* tujuan dengan keterangan berikut ini:

1) -A *full* untuk mode *alert* dengan informasi lengkap.
2) -A *unsock* untuk mode *alert* ke *unix socket*.
3) -A *none* untuk mematikan mode *alert*.

Agar Snortberoperasi secara langsung setiap kali *workstation* / *server* di *boot*, kita dapat menambahkan ke file /etc/rc.d/rc.local perintah di bawah ini /usr/local/bin/Snort–d –h 192.168.0.0/24 –c /root/Snort/Snort.conf –A full –s –D atau /usr/local/bin/Snort–d –c /root/Snort/Snort.conf –A full –s –D dimana –D adalah *switch* yang mengatur agar Snortbekerja sebagai *Daemon* (bekerja di belakang layar).

Kesimpulan dari Snortini adalah Snortmerupakan aplikasi pendeteksian serangan atau gangguan jaringan yang berbasis *Intrusion Detection System* (IDS) yang berfungsi untuk melakukan pengamanan jaringan.

2.1.4 *Firewall*

Firewall adalah alat yang digunakan untuk mencegah orang luar memperoleh akses kesuatu jaringan. *Firewall* merupakan suatu kombinasi dari perangkat lunak dan perangkat keras dan *firewall* biasanya menerapkan pengeluaran rencana atau perintah untuk memilih alamat yang tak dikehendaki dan diinginkan, *firewall* mempunyai cara kerja dalam pengamanan jaringan yaitu *firewall* bekerja dengan mengamati paket IP yang melewatinya (Purbo, 2000).

Fungsi-fungsi umum *firewall* adalah sebagai berikut:

1) *Static packet filtering* untuk penyaringan paket secara statis.
2) *Dynamic packet filtering* untuk penyaringan paket secara dinamis.
3) *Stateful filtering* untuk penyaringan paket berdasarkan status.
4) *Proxy*.

Berdasarkan konfigurasi dari *firewall*, akses dapat diatur berdasarkan IP *address*, *port*, dan arah informasi, sehingga untuk memahami *firewalls* bekerja kita perlu mengetahui pengalamatan IP Statis dan IP dinamis.

1) IP alamat statis adalah alamat yang permanen, yang merupakan alamat dari suatu mesin yang selalu dihubungkan ke internet.
2) IP address dinamis adalah alamat IP yang selalu berubah-ubah yang berfungsi sebagai koneksi ke jaringan.

Selain itu *firewall* juga mempunyai karakteristik dalam pengamanan jaringan diantaranya :

1) Segala lalu lintas jaringan, baik dari dalam atau dari luar harus melalui *firewall*, hal tersebut akan terhalangi oleh *firewall* dari semua akses dalam bentuk apapun kecuali *firewall*.
2) Kebijakan keamanan hanya akan memberikan ijin untuk memasuki *server* atau jaringan komputer yang memenuhi syarat tertentu.
3) *Firewall* sendiri bebas terhadap penetrasi, yang menandakan bahwa suatu sistem dapat dipercaya dan menjamin keamanan dari sistem operasi.

Ada beberapa macam perbedaan dari *firewall*, masing-masing tipe memiliki keuntungan dan kerugian, secara umum, tipe dari *firewall* ada 3 macam. Yaitu :

1) Paket *filter router*

   Paket *filter router* menggunakan ketantuan untuk paket IP, mana yang boleh masuk dan mana yang harus ditolak. Informasi yang disaring dari suatu paket yang melewati jaringan, diantarannya:

   a) Sumber IP *address*: alamat asli dari IP paket (contoh: 192.186.1.2).
   b) Tujuan IP *address*: alamat IP yang akan menerima IP paket (contoh: 192.186.1.3).

c) Tujuan dan sumber *transport-level address* merupakan level *transport* dari *port number* (seperti TCP dan UDP).
d) IP *protocol,* yang berfungsi sebagai *transport protocol.*
e) *Interface*: untuk *router* dengan tiga atau lebih *port*, dari *interface router* mana paket datang atau bertujuan.

2) *Application Level Gateway*

Application level gateway juga dikenal dengan *application-proxy firewall*. Pada tipe ini *user* harus melakukan kontak dengan *gateway* yang menggunakan aplikasi komponen *proxy* diantaranya adalah :

a) *Telnet*
b) FTP
c) *Rlogin*
d) *Sendmail*
e) HTTP
f) *The x window system*

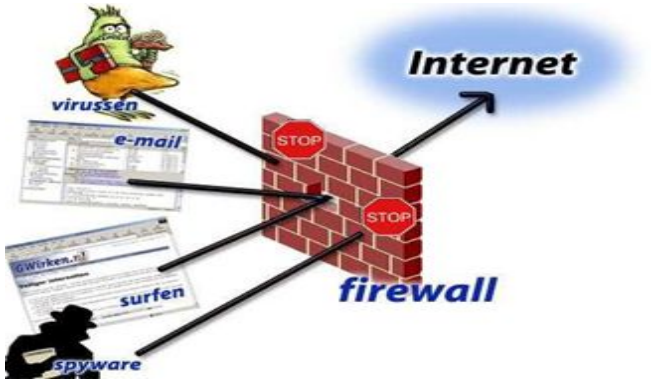
3) *Circuit Level Gateway*

Circuit level gateway merupakan sistem *proxy server* yang secara statis menggambarkan jaringan lalu lintas yang akan disampaikan. *Circuit proxy* selalu mengizinkan paket yang berisi *port* alamat-alamat yang diizinkan oleh aturan *policy* (kebijakan).

Terdapat beberapa tujuan penggunaan *firewall*, antara lain:

a) *Firewall* biasanya digunakan untuk mencegah atau mengendalikan aliran data tertentu. Artinya, setiap paket yang masuk atau keluar akan diperiksa, apakah tepat atau tidak dengan criteria yang ada pada standar keamanan yang didefinisikan dalam *firewall*.
b) Untuk melindungi dengan menyaring, membatasi atau bahkan menolak suatu atau semua hubungan atau kegiatan suatu komponen pada jaringan pribadi dengan jaringan luar yang bukan merupakan ruang lingkupnya.
c) Penggunaan *firewall* yang dapat mencegah upaya berbagai *trojan horses*, *virus*, *phishin*, *spyware* untuk memasuki sistem yang dituju dengan cara mencegah hubungan dari luar, kecuali yang diperuntukan bagi komputer

dan *port* tertentu seperti pada Gambar 2.2.



Gambar 2.2 Desain Fungsi *Firewall*

d) *Firewall* akan memeriksa serta menyaring *traffic* yang melintasi perbatasan antara jaringan luar maupun dalam.

Dengan demikian bahwa *firewall* adalah sebuah alat atau aturan berfungsi sebagai pembatas antara jaringan satu dengan jaringan lain sehingga dapat dikatakan sebagai pembatas antara jaringan *internal* dan jaringan *external*.

2.1.5 Jenis-jenis Serangan

Menurut Purbo (2010) ada beberapa jenis dan teknik serangan yang dapat mengganggu keamanan jaringan komputer, antara lain:

1) *Traffic Flooding*

Membanjiri *traffic* atau lalu lintas jaringan dengan banyaknya data-data sehingga lalu lintas jaringan yang datang dari pengguna yang terdaftar menjadi tidak dapat masuk ke dalam sistem jaringan.

2) *Request Flooding*

Membanjiri jaringan dengan cara melakukan *request* sebanyak-banyaknya terhadap sebuah layanan jaringan yang disediakan oleh sebuah *client* sehingga *request* yang datang dari para pengguna terdaftar tidak dapat dilayani oleh layanan tersebut.

3) *Port Scanning*

Merupakan suatu proses untuk mencari *port* pada suatu jaringan komputer yang terbuka dan dapat dilakukan serangan. Hasil *scanning*

tersebut akan didapatkan letak kelemahan sistem tersebut. Pada dasarnya sistem *port scanning* mudah untuk di deteksi, namun penyerang akan menggunakan berbagai metode untuk menyembunyikan serangan tersebut.

4) *IP-Spoofing*

IP-Spoofing juga dikenal sebagai *Source address spoofing,* yaitu pemalsuan alamat IP penyerang sehingga sasaran menganggap alamat IP penyerang adalah alamat IP dari *host* di dalam *network*.

Maka dalam sebuah pengamanan jaringan harus memperhatikan 3 aspek yang mungkin terjadi, yaitu:

1) Risiko dan tingkat bahaya (*risk*) yaitu menyatakan seberapa besar kemungkinan dimana penyusup (*intruder*) berhasil mengakses komputer dalam suatu jaringan.
2) Ancaman (*threat)* yaitu menyatakan sebuah ancaman yang datang dari seseorang yang mempunyai keinginan untuk memperoleh akses *illegal* ke dalam suatu jaringan komputer seolah-olah mempunyai otoritas terhadap jaringan tersebut.
3) Kerapuhan sistem (*vulnerability*) yaitu menyatakan seberapa kuat sistem keamanan suatu jaringan komputer yang dimiliki dari seseorang dari luar sistem yang berusaha memperoleh akses *illegal* terhadap jaringan komputer tersebut.

2.2. Kerangka Pemikiran

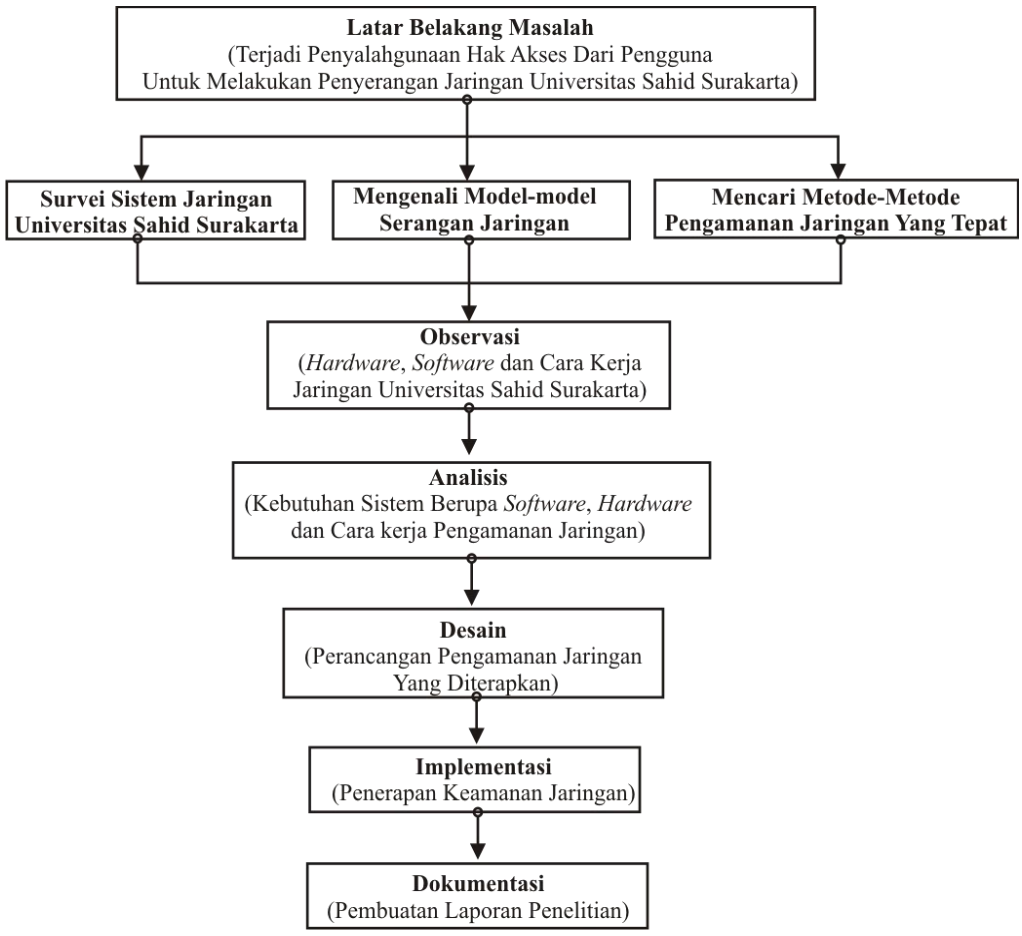
Kerangka pemikiran adalah proses dimana penelitian dimulai dari mendapatkan permasalahan sampai dengan dokementasi penelitian. Isi dalam kerangka pemikiran dalam penelitian keamanan jaringan ini yaitu

1) Latar belakang permasalahan adalah tahap utama mengenali permasalahan yang akan dijadikan objek penelitian.
2) Survei sistem jaringan yang ada, mengenali jenis-jenis serangan jaringan dan mencari metode pengamnan yang tepat.
3) Observasi adalah tahap melakukan mendalami hasil survei yang telah

dilakukan yang berkaitan dengan jaringan Universitas Sahid Surakarta.

4) Perancangan sistem adalah tahap mendesain pengamanan jaringan yang akan diterapkan.
5) Implementasi adalah tahap melakukan penerapan pengamanan jaringan untuk mengatasi permasalahan yang ada.
6) Dokumentasi adalah tahap membuat peloran penelitian atau pembukuan yang bertujuan untuk mengetahui hasil penelitian dikemudian hari.

Keterangan kerangka berfikir lebih lanjut dapat dilihat pada Gambar 2.3.



Gambar 2.3 Kerangka Pemikiran