UNIVERSITAS INDONESIA

ANALISA MEKANISME PERTAHANAN DOS DAN DDOS (*DISTRIBUTED DENIAL OF SERVICE*) PADA *VIRTUAL MACHINE* DENGAN MENGGUNAKAN IDS CENTER

SKRIPSI

MUHAMAD ZAMRUDI AH

0606078443

FAKULTAS TEKNIK

PROGRAM TEKNIK KOMPUTER

DEPOK

DESEMBER 2009

UNIVERSITAS INDONESIA

ANALISA MEKANISME PERTAHANAN DOS DAN DDOS (*DISTRIBUTED DENIAL OF SERVICE*) PADA *VIRTUAL MACHINE* DENGAN MENGGUNAKAN IDS CENTER

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar

Sarjana Teknik

MUHAMAD ZAMRUDI AH

0606078443

FAKULTAS TEKNIK

PROGRAM TEKNIK KOMPUTER

DEPOK

DESEMBER 2009

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama	**: Muhamad Zamrudi AH**
Npm	**: 0606078443**
Tanda tangan	**:** 
Tanggal	**: 15 Desember 2009**

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama	: Muhamad Zamrudi
NPM	: 0606078443
Program Studi	: Teknik Komputer
Judul Skripsi	: Analisa Mekanisme Pertahanan DoS dan DDoS (*Distributed Denial of Service*) pada *Virtual Machine* dengan menggunakan IDS Center

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Eektro Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Muhamad Salman ST. MT (.....)

Penguji : Dr. Abdul Muis S.T., M.Eng (.....)

Penguji : Prof. Dr.-Ing. Kalamullah Ramli M.Eng (.....)

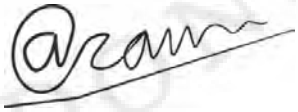
KATA PENGANTAR

Alhamdulillah, segala puji dan syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya saya dapat menyelesaikan Tugas akhir ini. Penulisan Tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi saya untuk menyelesaikan tugas akhir ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Muhammad Salman, ST, MIT, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan tugas akhir ini.
2. Kedua Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
3. Teman dan sahabat yang telah banyak membantu saya dalam menyelesaikan tugas akhir ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga tugas akhir ini membawa manfaat bagi masyarakat dan pengembangan ilmu pengetahuan.

Depok, Desember 2009



Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Zamrudi Al Hadiq

NPM : 0606078443

Program Studi : Teknik Komputer

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis karya : Tugas akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** *(Non-exclusive Royalty-Free Right)* atas karya ilmiah saya yang berjudul :

ANALISA MEKANISME PERTAHANAN DOS DAN DDOS (*DISTRIBUTED DENIAL OF SERVICE*) PADA *VIRTUAL MACHINE* DENGAN MENGGUNAKAN IDS CENTER

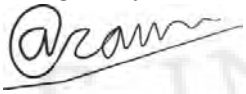
beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia / formatkan, mengelola dalam bentuk pangkalan data *(database)*, merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 15 Desember 2009

Yang menyatakan



(Muhammad Zamrudi Al Hadiq)

ABSTRAK

Nama : Muhamad Zamrudi Al Hadiq

Program Studi : Teknik Komputer

Judul : Analsia Mekanisme Pertahanan *DoS dan DdoS* (*Distributed Denial of Sevice*) pada *Virtual Machine* menggunakan *IDS Center*

Skripsi ini adalah sebuah analisa pada mekanisme pertahanan terhadap serangan *Denial of Service dan Distributed Denial of Service*. *DoS* dan *DdoS* adalah serangan yang sering terjadi dijaringan *internet* maupun jaringan lokal. Serangan ini berdampak negatif yang cukup merugikan korbannya. Serangan ini terbagi menjadi 2 yaitu *logical* dan *flooding. Flooding* merupakan jenis serangan yang sering terjadi. Maka dari itu akan perlu sebuah mekanisme pertahanan yang mampu menangani jenis serangan ini.

Pada skripsi ini dijelaskan tentang mekanisme pertahanan yang telah dibuat sedemikian rupa sehingga mampu diimplementasikan pada applikasi *IDS center 1.1* dan *Snort 2.8*. Ujicoba dilakukan pada jaringan *virtual* yang dibuat menggunakan aplikasi *VirtualBox*. Aplikasi ini mampu membuat *virtual machine* yang dapat di-install O*perating System*. Dengan menggunakan jaringan virtual analisa dapat dilakukan lebih mudah. Simulasi serangan dilakukan menggunakan applikasi *WinArpAttacker*. *WinArpAttacker* adalah applikasi yang mampu membanjiri jaringan dengan paket tcp, arp maupun udp.

Analisa yang dilakukan dengan membandingkan parameter-parameter. Parameter tersebut antara lain adalah *reliability* dan *response time*. Parameter *reliability* mewakili tingkat keberhasilan mekanisme pertahanan pada percobaan yang dilakukan. Parameter *response time* mewakili tingkat *delay* dan waktu respon terhadap jaringan internet ketika terjadi serangan. Dari percobaan tersebut menghasilkan tingkat *reliability* ketika terjadi serangan *DoS* adalah 92% sedangkan pada *DDoS* adalah 82%. Nilai rata-rata *response time* terhadap jaringan internet per satuan waktu adalah 2086.75 ms.

Kata kunci:

Keamanan Jaringan, Virtual Machine, Pertahanan, DoS, DDoS

ABSTRACT

Name : Muhamad Zamrudi Al Hadiq

Study Program : Computer Engineering

Title : DoS and DDoS (Distributed Denial of Service) Defense Mechanism Analisys on The Virtual Machine using IDS Center

This thesis is an analysis of the defense mechanisms against Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. DoS and DDoS attacks often happen in Internetwork or local network. This attack is quite a negative impact against the victim. This attack is divided into 2 kinds which is logical and flooding. Flooding is a type of frequent attacks. Thus it would need a defense mechanism that can handle this type of attack.

In this thesis described the defense mechanisms that have been made in such a way that can be implemented in the application IDS 1.1 center and Snort 2.8. Test performed on a virtual network created using VirtualBox application. This application is able to create a virtual machine that can be installed operating system. By using a virtual network, analysis can be done more easily. Attack simulation conducted using WinArpAttacker 3.50 application. WinArpAttacker is an application that can flood the network with TCP, ARP or UDP packet.

The analysis carried out by comparing the parameters. These parameters include reliability and response time. Reliability parameter represents the success rate of defense mechanisms in the experiments conducted. This parameter represents the level of response time delay and response time of the Internet network during the attack. From the experimental results in the level of reliability when there is a DoS attack at 92% while DDoS is 82%. The average value of response time on the internet network per unit time was 2086.75 ms.

Key Word:

Network Security, Virtual Machine, Defense, DoS and DDoS

DAFTAR ISI

HALAMAN JUDUL ii
PERNYATAAN ORISINALITAS iii
PENGESAHAN iv
KATA PENGANTAR v
PERNYATAAN PUBLIKASI vi
ABSTRAK vii
DAFTAR ISI ix
DAFTAR GAMBAR xi
DAFTAR TABEL xii
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Tujuan Penulisan 5
1.3 Batasan Masalah 5
1.4 Metodologi Penulisan 5
1.5 Sistematika Penulisan 6
BAB 2 DASAR TEORI 7
2.1 Jaringan Komputer 7
2.1.1 Lapisan Fisik 9
2.1.2 Lapisan Datalink 13
2.1.3 Lapisan Network 14
2.1.4 Lapisan Transport 15
2.1.5 Lapisan Aplikasi 17
2.2 *Firewall* 18
2.2.1 Personal *Firewall* 20
2.2.2 Network *Firewall* 20
2.2.3 Proses Kerja *Firewall* pada *Router* 21

2.2.4 Manajemen Akses pada *Firewall* 23
2.3 IDS dan IPS 26
2.3.1 IPS (*Intrusion Prevention System*) 27
2.3.2 IDS (*Intrusion Detection System*)..... 28
2.4 *Virtual Machine* 30
2.5 *Network Security* 32

BAB 3 PERANCANGAN TOPOLOGI JARINGAN 34
3.1 DoS dan DDoS 34
3.1.1 DoS (*Denial of Service*) 35
3.1.2 DDoS (*Distributed Denial of Service*)..... 36
3.2 Peralatan yang Dibutuhkan..... 38
3.2.1 *Personal Computer* 38
3.2.2 Pantech WWAN Controller Modem Broadband 38
3.2.3 *VirtualBox* 38
3.2.4 IDS Center dan SNORT..... 38
3.2.5 *WinArpAttacker* 38
3.3 Instalasi Jaringan Menggunakan *Virtual Box* 39
3.4 Topologi Jaringan Utama 43
3.5 Skenario Diagram 45
3.5.1 Skenario DoS..... 46
3.5.2 Skenario DDoS..... 47
3.6 Hipotesa Mekanisme Pertahanan 48
3.6.1 *Prevention phase* 49
3.6.2 *Termination Phase* 50

BAB 4 ANALISA DAN PEMBAHSAN 52
4.1 Penentuan Parameter Pengukuran..... 52
4.2 Penentuan Skenario Serangan..... 53

4.3 Perhitungan dan Analisa 54
4.3.1 Perhitungan Skenario 1 54
4.3.2 Perhitungan Skenario 2 59
4.4 Analisa dan Pembahasan 63
BAB 5 KESIMPULAN 64
DAFTAR REFERENSI 65

DAFTAR GAMBAR

Gambar 1.1. Jumlah pengguna *internet* pada benua Asia	2
Gambar 1.2. Tingkat Penetrasi di Asia tahun 2009	3
Gambar 1.3 Tingkat Kerugian yang dihasilkan dari Intrusi	4
Gambar 2.1 Diagram Alir Protokol	8
Gambar 2.2 Penampang Kabel UTP	10
Gambar 2.3 Penampang Kabel Koaxial	11
Gambar 2.4 Penampang Kabel Fiber Optik	13
Gambar 2.5 Hubungan antara paket dengan Frame	13
Gambar 2.6 Proses *3-way Handshake*	16
Gambar 2.7 Perlindungan *Firewall*	19
Gambar 2.8 Diagram Jenis-jenis *Firewall*	19
Gambar 2.9 Diagram bagaimana nat bekerja	22
Gambar 2.10 Gambaran *protocol telnet vs SSH*	25
Gambar 2.11 Gambar Sun-xvm VirtualBox	31
Gambar 3.1 Tipe Serangan DoS dan DDoS	36
Gambar 3.2 Skema Serangan DDoS	37
Gambar 3.3 Instalasi *VirtualBox*	39
Gambar 3.4 Proses Instalasi *Virtual Machine*	40
Gambar 3.5 Proses Instalasi Virtual HDD	40
Gambar 3.6 Konfigurasi Ethernet Adapter	41
Gambar 3.7 Diagram Alir pembuatan Topologi Jaringan	42
Gambar 3.8 Topologi Jaringan Utama	44
Gambar 3.9 Tampilan Host dan Virtual OS	45
Gambar 3.10 Skenario Simulasi DoS	46
Gambar 3.11 Skenario Simulasi DDoS	47
Gambar 3.12 Skema mekanisme Pertahanan DoS dan DDoS	49

Gambar 4.1 Tabel Serangan DoS dan DDoS 54
Gambar 4.2 Kondisi Sebelum serangan 55
Gambar 4.3 Kondisi Saat Serangan Sebelum dipasang IDS 55
Gambar 4.4 Konfigurasi IDS center 56
Gambar 4.5 Hasil Response Time pada Skenario 1(*prevention phase*) 57
Gambar 4.6 Data Response time pada Skenario 1(*Termination Phase*) 59
Gambar 4.7 Topologi Serangan DDoS pada scenario 2 60
Gambar 4.8 Kondisi agent2 yang tidak dapat melakukan koneksi 60
Gambar 4.9 Nilai response time pada scenario 2 (*prevention phase*) 61
Gambar 4.10 Nilai response time pada scenario 2 (*Termination Phase*) 62

DAFTAR TABEL

Tabel 4.1 Tingkat *reliability* pada scenario 1 (*Prevention phase*) 57

Table 4.2 Tingkat *reliability* pada scenario 1 (*Termination Phase*) 58

Tabel 4.3 Tingkat *reliability* pada Skenario 2 (*Prevention phase*) 61

Tabel 4.4 Tingkat *reliability* pada scenario 2 (*Termination Phase*) 62

Tabel 4.5 Nilai rata-rata Response time berdasarkan scenario 63

Tabel 4.6 Nilai Response Time terhadap tingkat serangan DoS dan DDoS 64

BAB 1

PENDAHULUAN

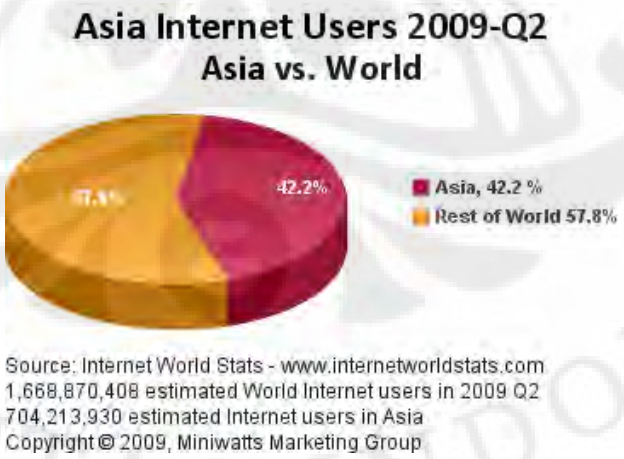
1.1. Latar Belakang

Pada era globalisasi yang semakin cepat dunia semakin terasa dekat karena pemanfaatan teknologi informasi yang semakin baik. Pada era tahun 70-an manusia masih menggunakan surat dan kantor pos untuk berkomunikasi dengan orang lain yang berjarak jauh, itu pun membutuhkan waktu lama untuk dapat berkomunikasi menggunakan surat. Saat ini kita dapat berkomunikasi dengan manusia di belahan dunia yang lain melalui *internet*, kita juga dapat mengirim data/file kedalam suatu database sehingga semua orang bisa melihat atau mendapatkan *file* itu dari *internet*, bahkan kita juga dapat melihat informasi berita yang *real-time* melalui teknologi jaringan *internet* ini. Teknologi informasi ini juga diimplementasikan di berbagai bidang seperti industri, transportasi, telekomunikasi, dan lain sebagainya. Hampir semua bidang memanfaatkan teknologi informasi ini, tujuannya tidak lain tidak bukan adalah untuk meningkatkan kinerja manusia yang terbatas. Akan tetapi belum semua masyarakat dunia bisa merasakan manfaat dari teknologi informasi dan komunikasi ini.

Sebagian orang belum dapat menggunakan teknologi informasi dan komputer dengan baik karena keterbatasan pengetahuan mereka terhadap dunia IT. Hal tersebut medorong para pakar IT (*information technology*) untuk membuat aplikasi yang mudah digunakan oleh semua orang. *User interface, icon dan pointing device* merupakan beberapa cara untuk membuat suaatu aplikasi mudah digunakan. Tetapi kemudahan dalam menggunakan teknologi tersebut membuat orang awam tidak mengerti sistem yang digunakan, sehingga orang-orang yang mempunyai I'tikad buruk dapat memanfaatkan celah ini untuk melakukan kejahatan.

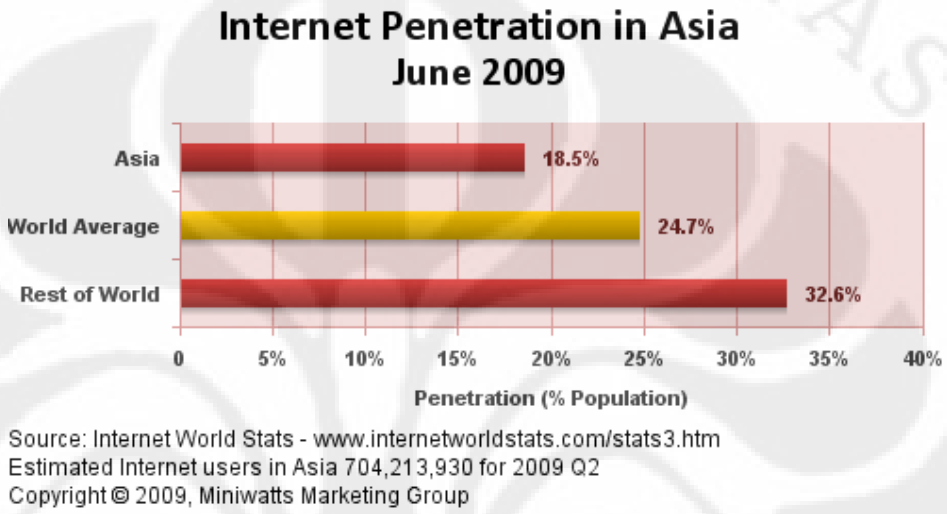
Suatu ciptaan manusia hakikatnya tidak ada yang sempurna, ciptaan tersebut pasti memiliki keuntungan dan kerugian. Tidak terkecuali teknologi komputer yang sampai saat ini sangat membantu mempermudah kehidupan manusia. Dengan semakin mudahnya orang menggunakan *internet*, semakin banyak orang yang menggantungkan pekerjaannya pada teknologi ini. Akan tetapi ada sebagian orang memanfaatkan *internet* ini sebagai media untuk melakukan tindak kejahatan. Sebagai contoh: Terkadang kita tidak menyadari ada seseorang yang dapat masuk jaringan kita dan dia mencuri data atau informasi yang kita miliki. Dalam dunia *internet* sekarang ini hal tersebut sangat mungkin terjadi. Efek yang timbul dari kejahatan tersebut ternyata sangat merugikan bagi orang yang menjadi korban. Mulai dari perusakan, pencurian, penipuan bahkan penghancuran. Maka dari itu keamanan jaringan komputer menjadi hal yang harus diperhatikan kepada para pengguna jaringan tersebut.

Saat ini pertumbuhan pengguna *internet* semakin banyak seiring berjalannya waktu. Tercatat pada tahun 2000 pengguna *internet* di asia mencapai 114.000.000 dan sekarang pengguna *internet* di asia naik hingga 3,808,070,503 [1]. Pertumbuhan yang terjadi dari data tersebut memperlihatkan bahwasanya pengguna *internet* meningkat begitu pesat. Begitu juga tingkat kejahatan yang terjadi di dunia maya ini. Tercatat *penetrasi* yang terjadi di asia pada tahun 2009 mencapai 18,5 % [1].



Gambar 1.1. Jumlah pengguna *internet* pada benua Asia dibandingkan dengan dunia [1]

Berdasarkan data—data tersebut banyak perusahaan membentuk sistem keamanan jaringan komputer yang berfungsi untuk melindungi jaringan kita dari ancaman bahaya. Keamanan jaringan sekarang ini menjadi bagian penting bagi sistem komputer dalam suatu perusahaan institusi bagi orang – orang yang merasa prihatin atas tidak criminal yang terjadi di *internet* ini.

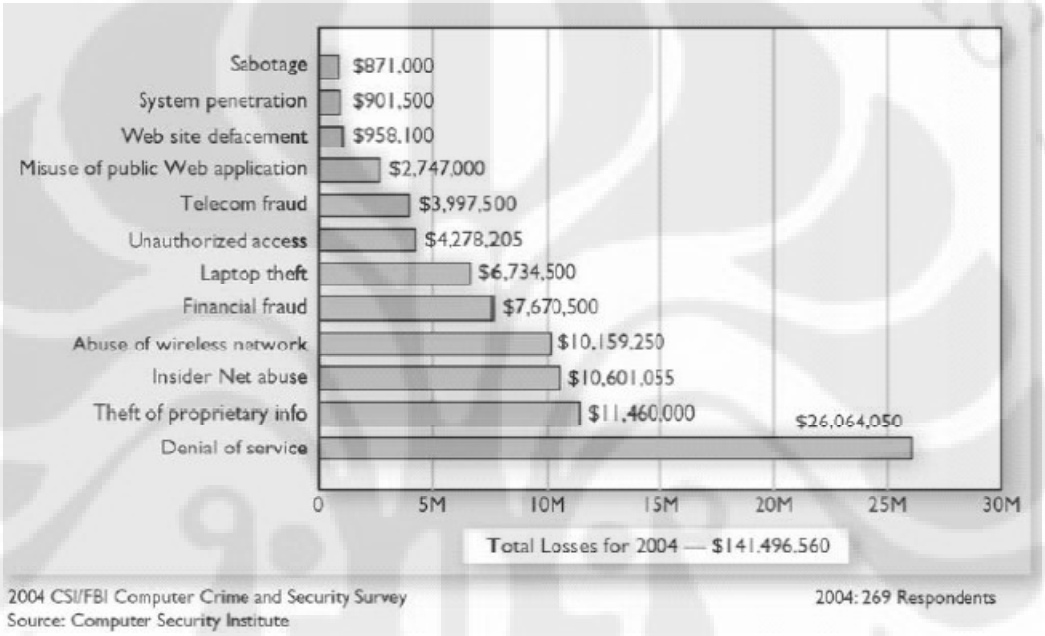


Gambar 1.2. Tingkat penetrasi di Asia pada tahun 2009 [1]

Saat ini begitu banyak cara untuk melakukan serangan terhadap suatu sistem jaringan. Cara-cara ini terus berkembang dari zaman dahulu sampai sekarang. Dahulu untuk melakukan suatu serangan membutuhkan pengetahuan dan pemahaman teknis IT yang tinggi, akan tetapi saat ini sangat mudah untuk melakukan serangan, bukan hanya orang yang mempunyai skill yang tinggi. Metode dan alat-alat yang dipakai semakin banyak dan mudah digunakan, bahkan untuk orang awam. maka dari itu semakin tinggi pula tingkat serangn yang terjadi terhadap sistem keamanan jaringan. Berikut ini merupakkan gambaran perkembangan metode-metode serangan yang digunakan dari tahun ke tahun.

Salah satu jenis serangan yang masih sering digunakan adalah DoS dan DDoS. DoS merupakan kependekan dari *Denial of Service*. Dos ini merupakan serangan yang mengakibatkan sistem yang diserang mengalami gangguan. Ganguan tersebut bisa berupa kegagalan sistem, *halt, error request* bahkan kerusakan *hardware server* tersebut [2].

DoS dan DDoS merupakan serangan yang berbahaya karena akibat yang dihasilkan dari serangan berdampak luas. Serangan ini mudah dilaksanakan dengan *tools* yang minimum atau pengetahuan *scripting* yang tidak terlalu tinggi. DoS dan DDoS bisa terjadi dimana saja dan kapan saja. Motif serangannya pun berbeda-beda sehingga mengakibatkan susahnya melacak orang-orang yang terlibat dalam serangan ini.



Gambar 1.3. Tabel tingkat kerugian yang dihasilkan dari beberapa intrusi yang terjadi. [3]

Tabel tersebut merupakan dampak ekonomi yang dihasilkan dari beberapa serangan pada jaringan *internet*. *DoS dan DDoS* merupakan serangan yang mengakibatkan pengeluaran biaya terbesar.

Maka dari itu diperlukan sebuah solusi untuk menyelesaikan permasalahan tersebut. Dengan tulisan ilmiah ini saya mencoba memberikan solusi yaitu mekanisme untuk mempertahankan jaringan dari serangan DoS maupun DDoS.

1.2. Tujuan

Tujuan dari penulisan skripsi ini adalah:

1. Menganalsis sistem keamanan pada suatu jaringan komputer agar dapat mendeteksi dan mencegah terjadinya serangan yang membahayakan jaringan komputer tersebut.
2. Dapat memanfaatkan perangkat *software* seperti *SNORT, IDS Center* dan *VirtualBox* yang mampu memberikan keamanan terhadap jaringan komputer.
3. Memberikan solusi mekanisme pertahanan *DoS dan DDoS* sesuai dengan kondisi jaringan tersebut.

1.3. Batasan Masalah

Penulisan skripsi ini dibatasi pada hal-hal berikut ini:

1. Analisa dilakukan pada sistem jaringan keamanan lokal dengan menggunakan *software simulator VirtualBox*.
2. Pecobaan dilakukan pada satu topolgi jaringan dengan beberapa scenario.
3. Pengukuran dilakukan pada 2 skenario yang berbeda berdasarkan parameter yang telah ditentukan yaitu: *reliability* dan *response time.*

1.4. Metologi Penulisan

Metode penulisan yang dipakai pada laporan tugas akhir/skripsi ini adalah:

a. Studi *literature*, yaitu dengan membaca dari buku-buku dan *ebook* yang digunakan sebagai referensi.
b. Konsultasi dengan pembimbing skripsi maupun dengan enggineer lainya yang berpengalaman dalam hal yang bersangkutan.
c. Teknik Simulasi, yaitu melakukan analisa terhadap simulasi topologi jaringan telah dirancang pada Bab 3.

1.5. Sistematika Penulisan

Laporan ini terdiri dari lima bab, dimana masing-masing bab akan menjelaskan sebagai berikut:

a. Bab 1: Pendahuluan

   Pada bab ini, akan dijelasan mengenai Latar Belakang, Tujuan, Pembatasan Masalah, Metodologi Penulisan, dan Sistematika penulisan.

b. Bab 2: Dasar Teori

   Pada bab ini, akan dijelaskan mengenai dasar-dasar teori yang digunakan antara lain sebagai berikut: Jaringan komputer, *Firewall*, IDS dan IPS, *Virtual OS, Network Secuirty*

c. Bab 3: Perancangan Topologi Jaringan

   Pada bab ini, akan dijelaskan mengenai perancangan topologi yang digunakan pada skripsi ini. Pada bagian ini juga dirancang hipotesa mekanisme pertahanan DoS dan DDoS.

d. Bab 4: Pengujian dan Pengukuran Mekanisme Pertahanan DoS dan DDoS

   Pada bab ini akan dijelaskan mengenai perhitungan dan pengukuran dari hasil topologi yang telah dibuat.

e. Bab 5: Kesimpulan

   Pada bagian ni akan dijelaskan mengenai kesimpulan yang dapat diambil dari pembahasan laporan Skripsi ini.

BAB 2
DASAR TEORI

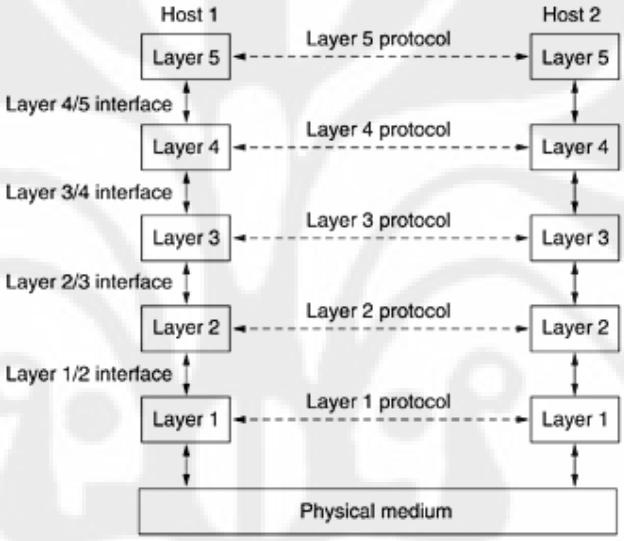
Pada bagian dasar teori ini akan dibahas mengenai jaringan komputer, *Firewall, IDS & IPS, SNORT, Virtual machine, Network Security* dan *DoS & DDoS*.

2.1. Jaringan Komputer

Jaringan komputer dapat digunakan untuk berbagai layanan, baik untuk perusahaan maupun untuk individu. Untuk perusahaan-perusahaan, jaringan komputer dipakai bersama-sama dengan menggunakan *server* yang sering menyediakan akses ke informasi perusahaan. Biasanya jaringan perusahaan mengikuti model *client-server*, dengan klien *workstation* pada *desktop* karyawan mengakses *server* yang kuat dalam ruang mesin [4]. Untuk jaringan komputer individu, jaringan menawarkan akses ke berbagai sumber daya informasi dan hiburan. Individu sering mengakses *Internet* dengan melakukan koneksi ke ISP menggunakan *modem*, meskipun saat ini semakin banyak orang yang memiliki sambungan tetap di rumah. Teknologi jaringan komputer terbaru yang akan datang adalah jaringan nirkabel dengan aplikasi baru seperti *mobile* akses *e-mail* dan *m-commerce*.

Secara kasar, jaringan dapat dibagi menjadi LAN, MAN, WAN, dan *internetwork* [4]. Jenis-jenis jaringan tersebut mempunyai karakteristik mereka sendiri-sendiri dalam teknologi seperti kecepatan, dan *Bandwith*. LAN mencakup jaringan dalam satu bangunan dan beroperasi pada kecepatan tinggi. MAN mencakup jaringan sebuah kota, misalnya, sistem televisi kabel, yang sekarang digunakan oleh banyak orang untuk mengakses *Internet*. WAN mencakup jaringan negara atau benua. LAN dan MAN adalah jaringan yang *unswitched* (yaitu jaringan yang tidak menggunakan *router*) [4]. Jaringan nirkabel menjadi sangat populer, terutama LAN yang nirkabel (*wireless*). Semua Jaringan terebut dapat saling berhubungan untuk membentuk *internetwork.*

Perangkat lunak jaringan terdiri dari beberapa *protocol* yang merupakan aturan atau proses untuk berkomunikasi antar jaringan atau *layer* [4]. *Protocol* yang dimaksud disini *connectionless* dan *connection-oriented.* Sebagian besar mendukung *protocol* jaringan hierarki, dengan menyediakan setiap lapisan layanan untuk lapisan di atasnya mereka mengisolasi rincian *protocol* yang digunakan pada lapisan bawah. Lapisan *protocol* yang baik biasanya didasarkan pada model OSI atau TCP / IP. Keduanya memiliki *network layer, transport layer,* dan *application layer*, tetapi keduanya memiliki perbedaan pada lapisan yang lain.



Gambar 2.1. Diagram alir *protocol* yang digunakan pada jaringan komputer. [4]

Jaringan komputer memberikan layanan kepada pengguna mereka. Layanan ini dapat berupa *connection-oriented* atau *connectionless*. Dalam beberapa jaringan, layanan *connectionless* disediakan dalam satu layer dan berorientasi memberikan sambungan layanan dalam lapisan di atasnya [4].

Terdapat banyak sekali jaringan komputer, beberapa yang sering dikenal adalah jaringan *Internet*, jaringan ATM, *Ethernet*, dan IEEE 802.11 LAN *nirkabel*. *Internet* berevolusi dari jaringan ARPANET, jaringan yang lain adalah jaringan yang ditambahkan untuk membentuk suatu *internetwork[4]. Internet* saat ini sebenarnya adalah sebuah kumpulan ribuan jaringan yang saling terkoneksi

menggunakan *router*. ATM banyak digunakan di dalam sistem telepon jarak jauh untuk *traffic* data. *Ethernet card* adalah jenis LAN yang paling populer yang dapat dijumpai di sebagian besar perusahaan besar dan universitas. Akhirnya, pada mengherankan LAN nirkabel kecepatan tinggi (hingga 54 Mbps) mulai digunakan secara luas.

2.1.1. Lapisan Fisik (*Physical Layer*)

Lapisan fisik adalah dasar dari semua jaringan. Alam memaksakan dua batas mendasar pada semua saluran, dan ini menentukan *bandwidth*. Batasan ini adalah batas *Nyquist*, yang berkaitan dengan saluran tak bersuara, dan batas *Shannon*, yang berkaitan dengan saluran bising [5].

Transmisi media terdiri dari 2 jenis yaitu *guided dan unguided*. Media transmisi *guided* jaringan utama adalah *twisted pair*, kabel koaksial, dan serat optik [5]. Media transmisi yang *unguided* adalah radio, gelombang mikro, inframerah, dan laser yang dimana medianya melalui udara. Sistem transmisi yang akan datang adalah komunikasi satelit, terutama sistem LEO.

Kabel UTP (Universal *Twisted pair*)

Meskipun karakteristik *bandwidth* pita magnetik sangat baik tetapi karakteristik *delay* pada pita magnetik sangat buruk. Waktu transmisi diukur dalam menit atau jam, bukan milidetik. Untuk beberapa aplikasi, koneksi online sangat diperlukan. Salah satu media yang tertua dan paling umum digunakan adalah media *transmisi twisted pair[5]*. Sebuah media transmisi *twisted pair* terdiri dari dua kabel tembaga terisolasi, biasanya tebalnya sekitar 1 mm. Kabel yang berputar bersama dalam bentuk heliks, seperti molekul DNA. Memutar dilakukan karena dua kawat sejajar merupakan sistem kabel yang buruk. Bila kawat berputar, gelombang yang mengalir melalui kabel tersebut tidak memancar keluar, sehingga mampu mendistribusikan *signal* dengan baik.

Aplikasi yang paling umum dari penggunaan kabel *twisted pair* adalah sistem telepon. Hampir semua telepon yang terhubung ke perusahaan telepon menggunakan media kabel *twisted pair*. Kabel *twisted pair* dapat menyambungkan *node* yang berjarak beberapa kilometer tanpa amplifikasi, tetapi untuk jarak yang lebih jauh, *repeater* diperlukan. Ketika banyak kabel *twisted pair paralel* yang terkoneksi dalam jarak jauh, seperti kabel yang datang dari sebuah gedung apartemen ke kantor perusahaan telepon, kabel-kabel tersebut digabung bersama-sama dan dibungkus dalam selubung pelindung.

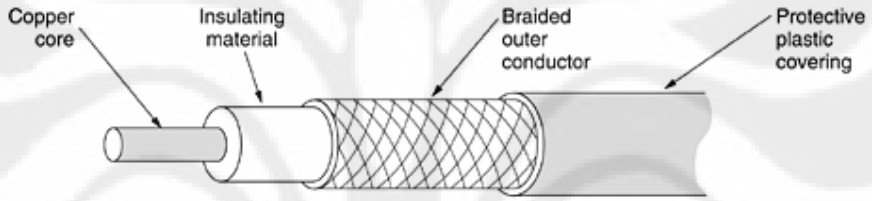
Twisted pair dapat digunakan untuk transmisi baik analog atau digital sinyal [13]. *Bandwidth* tergantung pada ketebalan kawat dan jarak yang ditempuh, tetapi beberapa megabit / detik dapat dicapai selama beberapa kilometer dalam banyak kasus. Berkaitan dengan kinerja yang memadai dan biaya rendah, memutar pasang secara luas digunakan dan kemungkinan besar akan tetap begitu selama bertahun-tahun yang akan datang.

Kabel *twisted pair* mempunyai beberapa jenis, dua di antaranya penting untuk jaringan komputer [5]. Kategori 3 *twisted pair* terdiri dari dua pasang kawat berisolasi lembut. *Four pair* biasanya dikelompokkan dalam sarung plastik untuk melindungi kabel dan menjaga mereka tetap tergabung. Sebelum tahun 1988, sebagian besar bangunan Kantor memiliki satu dari tiga kategori kabel yang terhubung dari pusat di setiap lantai ke setiap Kantor. Skema ini memungkinkan hingga empat telepon biasa atau dua *multi line* telepon di setiap kantor untuk menghubungkan ke pusat perusahaan telepon kabel.

Gambar 2.2. Penampang kabel UTP, a) kategori 3, b) kategori 5 [4]

Kabel Coaxial

Medium transmisi yang lain adalah medium transmisi kabel koaksial. Kabel koaksial mempunyai perlindungan yang lebih baik daripada kabel UTP, sehingga dapat menghubungkan *node-node* jarak jauh pada kecepatan yang lebih tinggi [5]. Dua jenis kabel koaksial yang digunakan secara luas. Jenis pertama, 50-ohm kabel, umumnya digunakan untuk untuk transmisi digital. Jenis yang lain, 75-ohm kabel, biasanya digunakan untuk transmisi analog dan kabel televisi tetapi menjadi lebih penting dengan munculnya *Internet* melalui kabel. Pembedaan ini didasarkan pada sejarah, daripada faktor-faktor teknis.



Gambar 2.3. Penampang kabel koaksial [4]

Konstruksi dari kabel koaksial memberikan kombinasi yang baik untuk *bandwidth* tinggi dan kekebalan suara yang sangat baik. Besarnya *Bandwidth* yang mungkin tergantung pada kualitas kabel, panjang, dan *signal-to-noise ratio* dari sinyal data [5]. Kabel modern memiliki *bandwidth* hampir 1 GHz. Kabel coaxial digunakan dalam sistem telepon untuk *interlokal line* tetapi sekarang sebagian besar telah digantikan oleh serat optik untuk rute jarak jauh.

Serat Optik

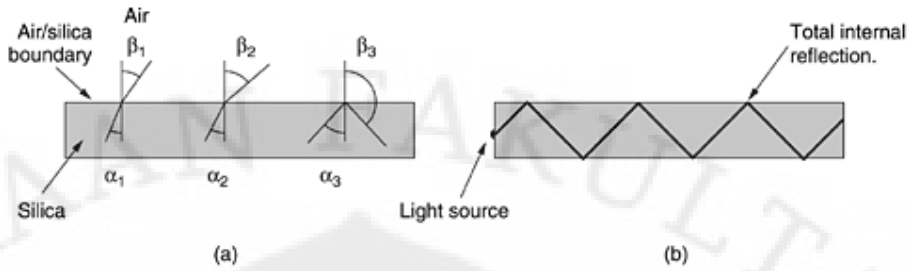
Pada periode yang sama, komunikasi data mempunyai kecepatan dari 56 kbps (ARPANET) sampai 1 Gbps (komunikasi optik modern), yang mempunyai peningkatan lebih satu faktor dari 125 per dekade, sementara pada saat yang sama tingkat kesalahan berkurang dari -10^5 per sampai angka yang hampir nol.

Selain itu, CPU mulai mendekati batas-batas fisik, seperti masalah-masalah kecepatan cahaya dan pembuangan panas. Sebaliknya, dengan teknologi serat optik saat ini, *bandwidth* yang dicapai tentu seharusnya lebih dari 50.000 Gbps (50 Tbps) dan banyak orang sangat sulit mencari teknologi dan bahan-bahan yang lebih baik. Batas yang bisa dicapai saat ini sekitar 10 Gbps, hal ini disebabkan oleh ketidakmampuan kita untuk mengkonversi antara sinyal listrik dan optik lebih cepat, meskipun di laboratorium 100 Gbps telah dapat dicapai pada satu serat [5].

Dalam kompetisi antara komputasi dan komunikasi, komunikasi mendapatkan kemenangan. Maka dari itu sebisa mungkin dihindari komputasi untuk mencapai *transfer rate* yang tinggi. Pada bagian ini akan dipelajari serat optik untuk melihat bagaimana teknologi transmisi bekerja.

Sebuah sistem transmisi optik memiliki tiga komponen: sumber cahaya, media transmisi dan detektor. Secara konvensional, sebuah pulsa cahaya menunjukkan 1 bit dan tidak adanya cahaya sedikit menunjukkan nilai 0. Media transmisi merupakan serat kaca yang ultra-tipis. Detektor menghasilkan pulsa listrik ketika cahaya jatuh di atasnya. Dengan memberikan sumber cahaya pada satu ujung serat optik kemudian detektor menerima sinyal pada ujung yang lain, kita memiliki sistem transmisi data searah yang menerima sinyal listrik, mengubah dan mentransmisikan dengan pulsa cahaya, dan kemudian mengubah kembali untuk output sinyal listrik di pihak penerima.

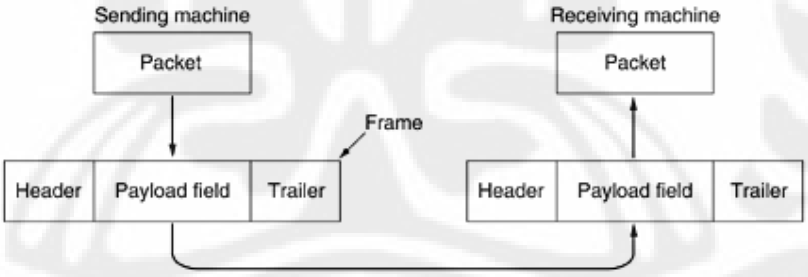
Dengan sistem tersebut serat optik merupakan salah satu media transmisi yang tercepat saat ini. Media ini sangat mempengaruhi *bandwith* dan kinerja jaringan komputer. Hal ini sangat menentukan keberhasilan serangan DoS maupun DDoS.



Gambar 2.4. Penampang kabel serat optik [4]

2.1.2. Lapisan Datalink (*Datalink Layer*)

Tugas *datalink layer* adalah untuk mengubah aliran bit mentah yang diberikan oleh *physical layer* ke dalam aliran *frame* agar dapat digunakan oleh *network layer*. Berbagai metode *framing* dapat digunakan, termasuk hitungan karakter, *byte stuffing* dan *bit stuffing*. *Protocol* Data link dapat menyediakan *control error* atau r*etransmit frame* yang rusak atau hilang. Untuk mencegah dari pengirim yang terlalu cepat pada penerima yang lambat, data link *protocol* juga dapat memberikan *flow control*. Mekanisme *sliding window* secara luas digunakan untuk mengintegrasikan *error control dan flow control* dengan cara yang nyaman [6].



Gambar 2.5 Hubungan antara paket dengan frame [4]

Dengan ini bisa diperiksa serangkaian *protocol* dalam bagian ini. *Protocol* 1 dirancang untuk *errors free environment* di mana penerima dapat menangani berbagai aliran yang dikirim ke *protocol* ini. *Protocol* 2 masih bisa dianggap *error free environment* tetapi ditambah dengan flow *control*. *Protocol* 3 menangani *error* dengan memperkenalkan nomor urut dan menggunakan algoritma *stop and wait*. *Protocol* 4 memungkinkan komunikasi dua arah dan memperkenalkan

konsep *piggybacking*. *Protocol* 5 menggunakan *protocol window sliding* dengan *go back n*. Akhirnya, *protocol* 6 menggunakan *selektif repeat* dan *negative acknowledgements*.

Banyak jaringan menggunakan salah satu *protocol* berorientasi bit-SDLC, HDLC, ADCCP, atau LAPB-di datalink layer [6]. Semua *protocol-protocol* ini menggunakan *flag byte* untuk membatasi *frame*, dan *flag stuff bit* untuk mencegah terjadi *over bytes* dari data. Semua dari mereka juga menggunakan *sliding window* untuk *flow control*. Di dalam jaringan, *Internet* menggunakan PPP sebagai *data link protocol* di atas *point to point line*.

2.1.3. Lapisan Jaringan (*Network layer*)

Network layer menyediakan layanan bagi *transport layer*. Hal ini dapat didasarkan baik pada *virtual circuit* atau *datagrams*. Dalam kedua kasus pekerjaan utama *network* layer adalah *routing* paket dari sumber ke tujuan. Dalam sirkuit *virtual subnet*, keputusan *routing* dibuat ketika rangkaian *virtual* di *set up*. Dalam *datagram subnet*, itu dibuat pada setiap paket.

Banyak *algoritma routing* yang digunakan dalam jaringan komputer. Algoritma statis termasuk algoritma *routing dan flooding* terpendek [7]. Algoritma *dynamic* meliputi vektor jarak *link state routing*. Sebagian besar jaringan aktual menggunakan salah satu dari algoritma ini.

Subnet dapat dengan mudah terjadi *Congestion*, peningkatan *delay* dan penurunan *throughput* paket. Desainer jaringan berusaha untuk menghindari *Congestion* dengan perancangan yang tepat. Teknik perancangan meliputi kebijakan *retransmission, caching*, *flow control*, dan banyak lagi. Jika *Congestion* terjadi, hal tersebut harus ditangani.

Langkah selanjutnya lebih dari sekedar berurusan dengan *Congestion*, jaringan harus mampu memberikan *QoS (Quality of Service)*. Metode yang dapat

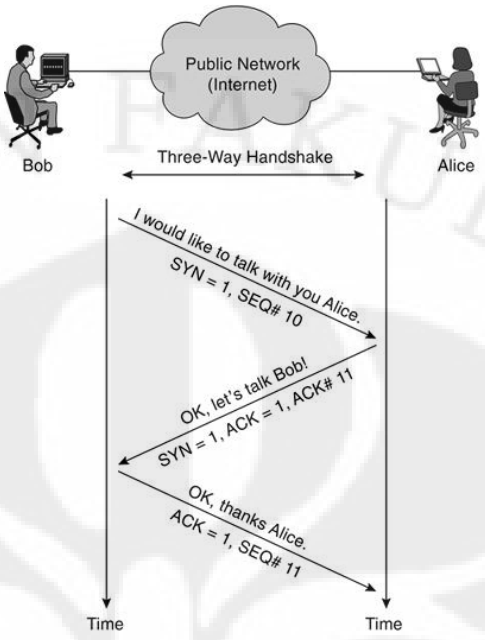
digunakan untuk hal ini mencakup *buffering* pada klien, *traffic shaping*, reservasi sumber daya, dan *admission control*.

Internet memiliki berbagai *protocol* yang berkaitan dengan *network layer*. Hal ini termasuk *data transport protocol, IP, ICMP control protocol, ARP, RARP, protocol routing* OSPF dan BGP [7]. Suatu saat jaringan *internet* akan kehabisan alamat IP, sehingga saat ini dikembangakan IP versi baru yaitu IPv6.

2.1.4. Lapisan *Transport* (*Transport layer*)

Transport layer adalah kunci untuk memahami lapisan *protocol* pada jaringan. *Transport protocol* menyediakan berbagai layanan, yang paling penting adalah *end-to-end, reliable*, *connection oriented byte stream* dari pengirim ke penerima [8]. Hal ini diakses melalui layanan primitif yang memungkinkan terjadinya pembentukan, penggunaan, dan pelepasan koneksi.

Transport protocol harus mampu melakukan pengelolaan koneksi melalui jaringan yang *unreliable*. Pembuatan koneksi menjadi rumit dengan adanya paket duplikat yang tertunda yang dapat muncul pada saat tidak tepat. Untuk menghadapi hal tersebut, *three way hand shake* diperlukan untuk membuat koneksi. Memutuskan sambungan lebih mudah daripada membangunnya tapi masih ada permasalahan yang jauh lebih sulit daripada dua masalah tersebut.



Gambar 2.6 Proses 3-way Handshake yang terjadi pada *Transport layer* [4]

Bahkan ketika lapisan jaringan benar-benar *reliable*, *transport layer* tetap memiliki banyak pekerjaan yang harus dilakukan. *Transport layer* harus menangani semua layanan primitif, mengelola koneksi dan timer, dan mengalokasikan dan memanfaatkan *memory* [4].

Internet memiliki dua *protocol transport* utama: UDP dan TCP. UDP adalah *protocol* yang *connectionless* yang berfungsi sebagai pembungkus untuk paket IP, dengan menggunakan fitur tambahan *multiplexing* dan *demultiplexing* terdapat beberapa proses pada satu alamat IP. *Protocol* UDP dapat digunakan untuk interaksi *client-server*, misalnya, menggunakan RPC. Juga dapat digunakan untuk membangun real-time *protocol* seperti RTP.

Transport protocol utama pada *Internet* adalah TCP. *Protocol* TCP Menyediakan *bidirectional byte* yang dapat diandalkan. Paket TCP Ini menggunakan *20-byte header* pada semua segmen. Segmen dapat terbagi oleh *router* di *Internet*, sehingga *host* harus siap untuk melakukan *reassembly*. Banyak penelitian yang telah dilakukan untuk mengoptimalkan kinerja TCP, dengan

menggunakan algoritma dari *Nagle, Clark, Jacobson, Karn,* dan lain-lain. *Wireless link* menambahkan berbagai komplikasi ke TCP. Transaksional TCP merupakan perpanjangan untuk TCP yang menangani interaksi *client-server* dengan penurunan jumlah paket [4].

Performa jaringan biasanya dipengaruhi oleh penggunaan *protocol* dan pengolahan *TPDU overhead*, dan situasi ini semakin buruk pada kecepatan yang lebih tinggi. *Protocol* harus dirancang untuk meminimalkan jumlah TPDUs agar kinerja jaringan dapat tercapai dengan baik.

2.1.5. Lapisan Aplikasi (*Application Layer*)

Penamaan di *Internet* menggunakan skema hirarkis yang disebut *Domain Sistem Name* (DNS) [4]. Pada tingkat atas adalah terkenal generik domain, termasuk com dan edu serta sekitar 200 negara domain. DNS diimplementasikan sebagai sistem database terdistribusi dengan *server* di seluruh dunia. DNS memegang *record* dengan alamat IP, pertukaran mail, dan informasi lainnya. Dengan *query server DNS*, suatu proses dapat memetakan nama domain *Internet* ke alamat IP yang digunakan untuk berkomunikasi dengan domain tersebut.

E-mail adalah salah satu dari dua aplikasi yang terkenal di *Internet*. Semua orang dari anak-anak kecil ke kakek-nenek sekarang menggunakannya. Sebagian besar sistem *e-mail* di dunia menggunakan sistem *e-mail* sekarang didefinisikan dalam RFC 2821 dan 2822 [4]. Pesan yang dikirim dalam sistem ini menggunakan ASCII sistem pesan *header* untuk mendefinisikan properti. Berbagai macam konten dapat dikirim menggunakan MIME. Pesan yang dikirim menggunakan *protocol* SMTP, yang bekerja dengan membuat sebuah koneksi TCP dari *host* sumber ke *host* tujuan dan langsung mengantarkan e-mail melalui koneksi TCP.

Aplikasi terkenal yang lain untuk *internet* adalah *World Wide Web*. Web adalah sistem *hypertext* untuk menghubungkan dokumen. Awalnya, setiap

dokumen adalah halaman yang ditulis dalam HTML dengan *hyperlink* ke dokumen lain. Kini, XML secara bertahap mulai mengambil alih dari HTML. Juga, sejumlah besar konten yang dihasilkan secara dinamis, dengan menggunakan *script* dari sisi *server* (PHP, JSP, dan ASP), serta *client side script* (terutama *JavaScript*). Browser dapat menampilkan dokumen dengan membentuk sebuah koneksi TCP ke *server*, meminta dokumen, dan kemudian menutup sambungan. Pesan permintaan ini mengandung berbagai *header* untuk memberikan informasi tambahan. Caching, replikasi, dan pengiriman konten jaringan banyak digunakan untuk meningkatkan performa Web.

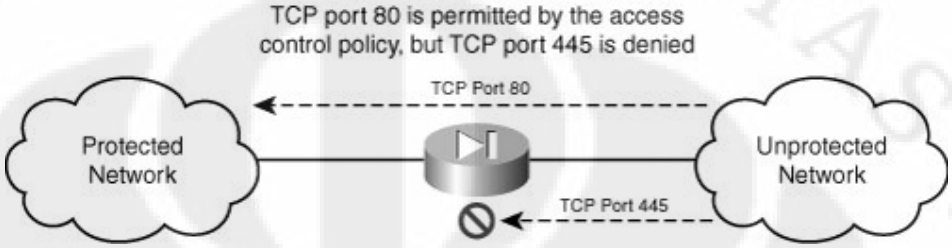
Web nirkabel baru saja dapat di aplikasikan akhir-akhir ini. Sistem pertama WAP dan i-mode, masing-masing dengan layar kecil dan *bandwidth* terbatas, tetapi generasi berikutnya akan lebih kuat.

Multimedia juga merupakan bidang yang sangat potensial berkembang pesat dalam jaringan. Hal ini memungkinkan *audio* dan *video* yang akan didigitasi dan diangkut secara elektronik untuk ditampilkan. *Audio* memerlukan lebih sedikit *bandwidth*, sehingga lebih mudah untuk diimplementasikan. *Streaming audio, Internet radio, dan voice over IP* adalah sebuah teknologi yang mampu di realisasikan sekarang. *Video on demand* adalah area potensial yang mana sangat menarik untuk dikembangkan . Akhirnya, MBone adalah sebuah eksperimental, *world wide digital live television* yang dikirim melalui *Internet [4]*.

2.2. *Firewall*

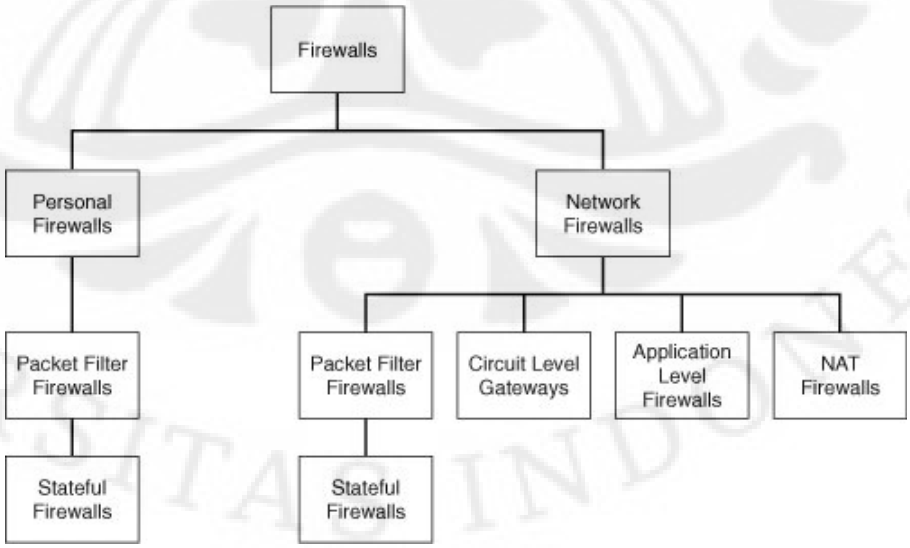
Firewall, terlepas dari bagaimana kompleks dalam desain dan implementasi, memiliki tanggung jawab sederhana untuk bertindak sebagai poin penegakan kebijakan pada keamanan. *Firewall* dapat melakukan ini dengan memeriksa data yang diterima dan melacak koneksi yang dibuat untuk menentukan data apakah koneksi tersebut diijinkan ata ditolak [12]. Selain itu, *firewall* dapat bertindak sebagai perantara dan permintaan *proxy host* yang dilindungi, sementara pada saat yang sama menyediakan sarana otentikasi akses

untuk lebih memastikan bahwa hanya perangkat akses diberikan. Akhirnya, *firewall* dapat melakukan pelaporan sehingga kita bisa waspada terhadap kejadian-kejadian yang berkaitan dengan semua proses ini agar *administrator* mengetahui apa yang terjadi dengan *firewall*.



Gambar 2.7 *Firewall* melindungi jaringan dari koneksi yang tidak mempunyai izin. [9]

Ada beberapa motif yang mendorong orang untuk melakukan ancaman terhadap sistem jaringan. Dengan memeriksa ancaman dan tanggapan yang sesuai, dapat dikembangkan kebijakan keamanan yang meminimalkan risiko yang dapat timbul oleh ancaman tersebut melalui pelaksanaan dan konfigurasi *firewall* yang tepat. Meskipun *firewall* tidak dapat mencegah semua serangan tetapi *firewall* adalah salah satu metode terbaik untuk melindungi sumber daya. *Firewall* setidaknya lebih dapat membantu membuat data aman daripada tanpa *firewall* sama sekali.



Gambar 2.8 Diagram jenis-jenis *firewall* [12]

2.2.1. Personal *Firewall*

Personal *firewall* dirancang untuk melindungi sebuah *host* dari akses yang tidak legal. Selama bertahun-tahun, hal ini telah berkembang sehingga saat ini personal *firewall* modern mengintegrasikan kemampuan tambahan seperti pemantauan perangkat lunak antivirus dan dalam beberapa kasus mampu meng-analisa perilaku serta *intrusion detection* untuk melindungi jaringan. Microsoft's *Internet Connection Firewall* merupakan salah satu personal *firewall* yang berjalan Windows XP dengan Service Pack 2. [12]

Personal *firewall* membuat arti besar di jaringan *internet* dan pengguna rumah karena mereka memberikan perlindungan *end-user* serta mampu mngendalikan kebijakan dalam melindungi sistem komputer [12]. Mungkin kekhawatiran terbesar bagi perusahaan pengguna yang berkaitan dengan personal *firewall* adalah kemampuan untuk menyediakan mekanisme kontrol yang terpusat pada *firewall* itu sendiri. Kebutuhan untuk mensentralisasi kontrol sangat penting untuk menggunakan personal *firewall* di lingkungan perusahaan untuk meminimalkan beban administrasi. Oleh karena itu, sangatlah penting bahwa ketika jumlah *firewall* meningkat, kemampuan untuk mengelola *firewall* tersebut harus tidak menjadi terlalu membebani jaringan. Dengan sentralisasi kontrol dan pemantauan banyak *vendor* berharap mampu mengurangi upaya konfigurasi *firewall* pada end-user.

2.2.2. *Network Firewall*

Network firewall dirancang untuk melindungi seluruh jaringan dari serangan. *Network firewall* terdiri dalam dua bentuk utama: *special tools* atau perangkat lunak *firewall* suite yang diinstal di atas sistem operasi *host*. Contoh *tools* jaringan yang berbasis *firewall* adalah *Cisco PIX, Cisco ASA*, *NetScreen Juniper firewall, Nokia firewall, dan Symantec Enterprise Firewall [8]*. *Network firewall* yang lebih populer merupakan *firewall* berbasis *software* termasuk *Check Point Firewall*-1 NG atau NGX *Firewall*, Microsoft ISA *Server*, IPTables

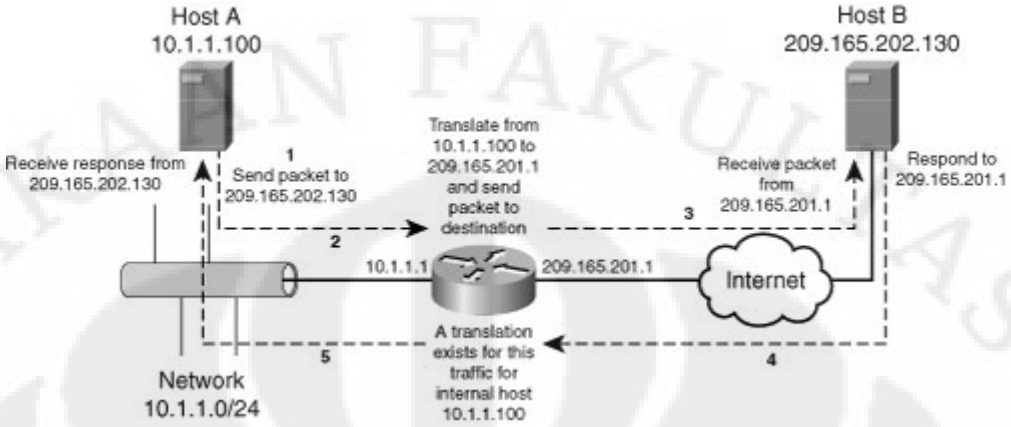
berbasis Linux, dan BSD pf *filter* paket. Sistem operasi Sun Solaris, di masa lalu, telah dibundel dengan *firewall* dari perusahaan Sun itu sendiri, *SunScreen*. Dengan rilis Solaris 10, Sun telah mulai *bundling open source IP Filter* (IPF) *Firewall* sebagai alternatif SunScreen.

Network Firewall pada banyak pengguna memberikan fleksibilitas maksimum dan perlindungan dalam sistem *firewall*. *Firewall* ini memiliki lebih banyak fitur baru seperti *in-line* intrusi deteksi dan *virtual private network* (VPN), *firewall* ini juga mempunyai kontrol yang baik untuk *LAN-to-LAN* VPN serta *akses-remote-user* VPN. *Firewall* dapat mengidentifikasi *traffic protocol* tidak hanya dengan melihat informasi pada Layer 3 dan Layer 4 tetapi dengan mempelajari semua jalan ke dalam data aplikasi *firewall* sehingga dapat membuat keputusan mengenai cara terbaik untuk menangani arus *traffic* jaringan.

2.2.3. Proses Kerja *Firewall* pada *Router*

Banyak *Broadband router* dan fungsi *firewall* dipakai menggunaakan *Network Address Translation (NAT)* untuk menyembunyikan sistem internal di belakang sebuah alamat IP *eksternal*. Sistem ini disebut "NAT *router*" atau "NAT *firewall*" melakukan pekerjaan yang memadai dengan menyembunyikan *resource* untuk melindungi dari metode serangan biasa, tetapi system tersebut tidak melakukan fungsi *firewall* sehingga sedikit keliru menyebut mereka *firewall.* Setidaknya *firewall* seperti Cisco Secure PIX *Firewall*, Microsoft ISA *Server*, dan Check Point *Firewall*-1 dianggap produk *firewall*. Sebaliknya, banyak *Broadband router* dan *firewall* NAT hanya berbasis packet-*filter*ing *router* yang memberikan tingkat privasi, tetapi mereka biasanya tidak memiliki fitur-fitur *firewall* canggih seperti *stateful packet inspection* (SPI), proxy data, atau *deep packet inspection[9]*.

Berikut ini akan dijelaskan mengenai bagaimana proses *firewall* dan NAT bekerja. Disini *Broadband router* sebagai *host* yang mampu dikonfigurasi dan di-*install* aplikasi yang mendukung NAT.



Gambar 2.9 Diagram yang menunjukan bagaimana NAT bekerja [12]

Langkah-langkah dalam Gambar dapat lebih dijelaskan sebagai berikut:

1. Klien memulai sambungan ke *host eksternal* (*Host*B).
2. *Broadband router* / *firewall* menerima permintaan dan menerjemahkan permintaan dari alamat IP internal ke alamat dari *router* / *firewall* 's *interface eksternal*. *Router* / *firewall* melacak terjemahan ini dalam sebuah tabel terjemahan.
3. Paket dikirim ke tujuan *eksternal* (*Host*B), yang percaya bahwa paket berasal dari alamat IP *eksternal* dari *router* / *firewall*. *Host eksternal* (*Host*B) menanggapi sesuai dengan alamat IP *eksternal* dari *router* / *firewall*.
4. Ketika *router* / *firewall* menerima jawaban dari *host eksternal*, mengecek tabel terjemahannya untuk permintaan *outbound* yang cocok.
5. Jika menemukan satu, *router* / *firewall repackages* dan menyampaikan paket ke *host* internal (*Host*A), yang berpikir bahwa respon dari *host eksternal* (*Host*B).

Selain itu, sebagian besar *Broadband router* / *firewall* yang dirancang untuk tidak mengizinkan segala bentuk paket-paket dari *host eksternal* untuk disampaikan kepada *host* internal [12].

Walaupun hal ini pada umumnya merupakan tingkat perlindungan yang memadai pada kebanyakan lingkungan *personal*, tetapi penting untuk dimengerti bahwa ketergantungan pada NAT untuk melindungi *host* adalah kesalahan yang fatal karena NAT tidak menjamin keamanan dalam dirinya sendiri, seperti tercantum dalam RFC 2663 Bagian 9.0. Sebagai contoh, perangkat NAT rentan terhadap serangan tertarget, seperti serangan *Denial of Service* (DoS), sebagai perangkat non-NAT [34]. NAT juga tidak menyediakan penyaringan paket yang sebenarnya meninggalkan jaringan internal, sebaliknya, hal itu memungkinkan semua *traffic* keluar asalkan dapat diterjemahkan sesuai. Meskipun perbedaannya tipis, NAT lebih menyediakan privasi ketimbang keamanan.

Oleh karena itu, bila digunakan bersama dengan teknologi lain NAT dapat berfungsi sebagai mekanisme keamanan yang efektif. *Broadband router* terbaik / *firewall* (misalnya, banyak dari *Linksys Broadband firewall)* termasuk aplikasi yang mampu melakukan penyaringan paket, *deep packet inspection* (DPI), SPI, *firewall*, dan NAT [9].

2.2.4. Manajemen akses pada *Firewall*

Pengendalian akses ke perangkat *interface* manajemen infrastruktur jaringan sangat penting. Perangkat jaringan seperti *router, switch, IDS sensor*, dan *firewall* harus dapat diakses hanya oleh para pengguna yang membutuhkan dan mempunyai kewenangan untuk melakukan konfigurasi. Persyaratan ini terjadi karena jika ada pengguna yang *illegal* memungkinkan seseorang dengan niat jahat dapat mengubah konfigurasi atau menonaktifkan perangkat, dan dengan demikian hal tersebut mampu menurunkan keamanan jaringan sekitarnya. Manajemen akses terdiri dari dua bentuk: *in-band* dan *out-of-band [12]*.

In-band Manajemen

In-band manajemen mengacu pada akses administratif pada sistem dan perangkat jaringan melalui jaringan yang sama yang digunakan oleh *traffic* yang di*filter* [9]. *In-band* manajemen dapat menggambarkan risiko signifikan yang mungkin terjadi kepada *Administrator*, jika tindakan pencegahan tertentu tidak diputuskan. Pusat risiko ini terutama pada seputar penggunaan saluran komunikasi yang terenkripsi. Perhatian khusus harus dilakukan untuk penggunaan komunikasi terenkripsi seperti SSH dan HTTPS. Penggunaan *Telnet* atau HTTP sederhana dapat mengakibatkan *password administratif* dapat diambil oleh seorang penyerang yang mampu melakukan *sniffing traffic* antara *interface* administratif dari *firewall* ke seluruh jaringan. *In-band* manajemen juga memungkinkan risiko rentan terhadap serangan *denial-of-service* (DoS) [9]. Hal ini membuat lebih sulit untuk mengkonfigurasi ulang *firewall*. Ketika terjadi peristiwa semacam itu *traffic* perlu diblokir atau bahkan mematikan *server* untuk mengalahkan serangan jika perlu.

Out-of-band Manajemen

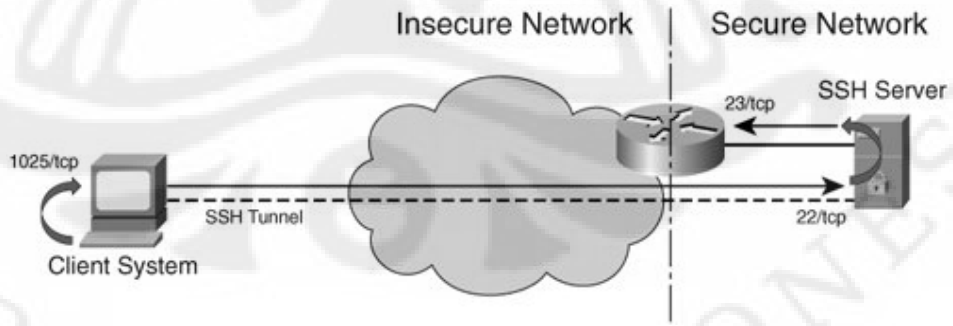
Out-of-band merupakan hasil manajemen dalam akses *firewall* melalui saluran sekunder yang tidak membawa *traffic production[12]*. Hal ini memungkinkan terjadinya pengaturan VLAN akses administratif untuk perangkat jaringan *host* atau sebuah jaringan yang sama sekali terpisah secara fisik. Selain itu, *out-of-band* manajemen dapat digunakan untuk menyediakan akses ke serial port dari perangkat jaringan. *Out-of-band* manajemen dapat lebih memakan waktu untuk menimplementasikannya dan membutuhkan biaya besar sehingga tidak efektif untuk jaringan kecil, tetapi *Out-of-band* merupakan manajemen yang paling aman dan dapat diandalkan metode administrasi *firewall* dan perangkat jaringan lainnya [12].

Telnet vs SSH

Telnet adalah sebuah *protocol* komunikasi jaringan tidak terenkripsi yang biasanya digunakan untuk menyediakan akses remote pada sistem dan perangkat

lain [10]. *Telnet* awalnya didefinisikan pada RFC 854 dan dikembangkan jauh sebelum *Internet* ada di jaringan. *Telnet* tidak mempunyai banyak pertimbangan yang diberikan dalam desain *confidential protocol Telnet* dalam data yang sedang dikirim menggunakan *protocol*. Oleh karena itu, semua data yang dikirim menggunakan *protocol Telnet* dapat di lihat dan rentan untuk di *capture*.

Protocol SSH menyediakan perlindungan kriptografi data, otentikasi dan memastikan bahwa integritas serta kerahasiaan komunikasi dapat diamankan. Jika perangkat jaringan dapat mendukung SSH sebagai metode akses ke baris perintah, sebaiknya menggunakan metode SSH daripada *Telnet*. Atau, jika perangkat GUI dapat diakses dalam jaringan yang aman dan perlu diatur perangkat remote melintasi jaringan yang tidak aman sehingga koneksi SSH dapat dibuat, maka memungkinkan untuk melakukan koneksi melalui *channel* SSH. Untuk membentuk sebuah *tunneling* pada SSH antara dua *host* harus menggunakan *port forwarding [10]*. Dalam contoh yang ditunjukkan pada Gambar 2.10 klien menetapkan sebuah koneksi SSH melalui SSH *server* pada TCP port 22 (port standar SSH). Namun, klien menggunakan *port-forwarding* kemampuan untuk melanjutkan dengan local*host* port TCP 1025 dan pengalihan ke *Telnet* port pada *router*. Untuk mengakses *Telnet* port dari *router* melalui *tunnel*, klien hanya perlu *telnet localhost* ke port TCP 1025 dan dia akan otomatis dibawa melalui *tunnel* SSH ke *router*'s *Telnet* port.



Gambar 2.10 Gambaran *protocol Telnet* dengan SSH [12]

Dengan cara ini *traffic* yang berjalan melalui SSH dienkripsi antara klien dan *server* SSH kemudian *traffic* dapat diteruskan dengan menggunakan *protocol* yang tidak aman seperti *Telnet*.

HTTP vs HTTPS

Sebuah diskusi tentang penggunaan HTTP vs HTTPS mempunyai metode yang sama seperti diskusi sebelumnya tentang *Telnet* versus SSH. HTTP adalah *protocol* tidak terenkripsi yang memungkinkan para penyadap untuk melihat komunikasi antara klien dan *server* [9]. Meskipun penyerang mungkin tidak selalu dapat menangkap sandi untuk *server* web, mereka mungkin dapat menangkap informasi lain seperti informasi konfigurasi tertentu atau mungkin mencuri *cookie* yang memungkinkan penyerang untuk menyamar sebagai pengguna yang sah dan mendapatkan akses ke *interface administrative firewall*.

HTTPS menggunakan *Secure Sockets Layer* (SSL) teknologi enkripsi untuk mengenkripsi komunikasi antara klien dan *firewall* web *server* [9]. Hal ini tidak mungkin bagi seorang penyerang untuk menguping manajemen mencegat sesi atau informasi apapun yang dapat digunakan untuk memperoleh akses ke *firewall* atau mendapatkan informasi tentang konfigurasi *firewall*

2.3. *Intrusion Detection Sistem* (IDS) dan *Intrusion prevention Sistem* (IPS)

Teknologi IDS dan IPS merupakan teknologi baru yang masih berkembang. Tetapi teknologi IDS dan IPS saat ini menjadi perhatian beberapa pihak khususnya perusahan yang berkaitan erat dengan jaringan Komputer, karena semakin tingginya tingkat ancaman dalam dunia *internet*. [11]

2.3.1. IPS (*Intrusion prevention Sistem*)

Intrusion prevention Sistems (IPS) adalah perlindungan keamanan perangkat atau aplikasi yang dapat mencegah serangan terhadap perangkat jaringan [11]. Sistem ini mulai ada sebagai fitur tambahan dari produk-produk lama, seperti *firewall* dan produk-produk antivirus, dan kemudian berkembang menjadi mandiri dengan fitur lengkap serta produk dalam hak mereka sendiri. Ada dua jenis IPSs: *Network* dan *Host*. Bagian ini membahas faktor-faktor yang menyebabkan keberadaan IPSs. Ini menggambarkan evolusi *threat* dari keamanan komputer, evolusi serangan mitigasi, dan beberapa kemampuan IPSs.

Kemampuan *Host Intrusion prevention System (HIPS)*

Sebuah pencarian terbaru dari *Internet* terdaftar lebih dari 700.000 halaman yang mengandung kata-kata *Host Intrusion prevention*. Hal ini menyulitkan untuk mengetahui produk mana yang benar-benar HIPS dan yang tidak menghadirkan *threat* pada jaringan, terutama mengingat volume informasi yang tersedia. Tantangan ini dipersulit dengan tidak adanya definisi resmi untuk kategori yang diterima oleh semua *vendor* HIPS.

Untuk membantu membedakan produk HIPS yang baik diperlukan pertimbangan yang matang, bagian ini berisi *ability* yang dapat dibandingkan untuk menentukan apakah produk HIPS tersebut baik. Bagian ini akan menjelaskan standar menentukan kualitas untuk HIPS. Untuk memenuhi syarat sebagai HIPS, sebuah produk harus memiliki *ability* berikut:

- ✓ Mampu memblokir tindakan yang mengandung kode berbahaya
- ✓ Tidak mengganggu operasi normal
- ✓ Membedakan antara peristiwa serangan dan normal
- ✓ Menghentikan serangan baru dan tidak dikenal
- ✓ Melindungi terhadap kekurangan dalam aplikasi diperbolehkan

Memblokir *Malicious Code*

Sebuah HIPS harus mampu melakukan lebih dari sekedar menghasilkan peringatan atau log ketika kode berbahaya menyerang sebuah *host* [11]. Hal ini harus dapat secara aktif menghalangi tindakan yang dilakukan oleh kode berbahaya. Jika tindakan mampu diblokir maka serangan tidak akan berhasil. produk HIPS harus juga menyimpan *log* dan dapat menghasilkan peringatan, sehingga pengguna akan mengetahui apa yang HIPS lakukan, tetapi yang harus menjadi catatan adalah bahwa HIPS harus mampu mengambil tindakan.

Sebagai contoh, salah satu cara untuk kode berbahaya menyebar dari suatu sistem telah berkompromi *host* lain adalah dengan menyalin dirinya sendiri untuk membuka jaringan berbagi. Beberapa alat-alat keamanan mungkin bisa mendeteksi kode berbahaya salinan usaha tetapi tidak mengambil tindakan terhadap itu. A HIPS harus mampu mendeteksi dan secara aktif usaha blok itu.

Tidak Mengacaukan Operasi Normal

Salah satu cara untuk mengamankan sebuah *host* adalah dengan mencabut *host* tersebut dari jaringan. Memutuskan hubungan dari jaringan memang akan membuatnya lebih aman, tetapi juga akan menghilangkan layanan jaringan yang bergantung pada pengguna bisnis. Pemutusan bukanlah penanggulangan keamanan yang berguna karena itu sama saja mengganggu operasi normal.

HIPS harus mampu beroperasi tanpa mengganggu operasi normal [11]. Sebagai contoh, lampiran e-mail mungkin menimbulkan risiko keamanan karena lampiran dapat berisi kode berbahaya. Salah satu cara untuk mengurangi risiko ancaman adalah dengan melepaskan semua pesan dari lampiran. Namun, lampiran e-mail sering kali merupakan bagian penting dari operasi normal. Sebuah produk yang akan menghapus semua lampiran e-mail tidak memenuhi syarat sebagai suatu produk HIPS karena mengganggu operasi normal.

Dapat Membedakan Antara *event* Serangan dan Normal

Sebuah produk keamanan yang memperlakukan peristiwa serangan seperti biasa dan normal seperti serangan peristiwa akan hampir sia-sia. HIPS produk harus cukup akurat untuk menentukan benar peristiwa serangan dan mana yang normal. Kita harus menguji beberapa *false positive* ketika pertama kali mengimplementasikan HIPS, tapi selanjutnya harus mencari mekanisme yang ada dalam produk untuk menghapus *false postive* tanpa mengganti produk untuk mendeteksi serangan [11].

Menghentikan Serangan Baru dan Tidak Diketahui

Kita dapat menggunakan berbagai metode untuk menghentikan terjadinya serangan. Sebagai contoh, kita dapat menerapkan *patch* untuk *software host* untuk menghilangkan *vulnerability*. Kita juga dapat mengupdate *signature* antivirus atau mengkonfigurasi ulang peralatan jaringan untuk mencegah serangan memasuki jaringan. Untuk setiap kelemahan baru yang mungkin terjadi, Kita mungkin perlu mengulang proses update dan konfigurasi ulang.

Sebuah teknologi yang memerlukan pembaruan atau konfigurasi ulang untuk menghentikan serangan baru yang tidak dikenal bukan merupakan jenis HIPS. HIPS harus mampu menghentikan serangan baru dan tidak dikenal tanpa konfigurasi ulang atau memperbarui. Cara HIPS menghentikan serangan dengan memiliki kemampuan untuk beradaptasi dengan serangan yang mungkin terjadi.

Melindungi Terhadap Kerusakan pada Aplikasi

Untuk memperoleh manfaat dari *host* dan jaringan, organisasi harus memungkinkan aplikasi untuk berjalan di *host* dan mengakses jaringan. Sebuah produk yang mencegah aplikasi diizinkan untuk menggunakan sumber daya yang dibutuhkan tidak memenuhi "tidak mengganggu operasi normal" kriteria. Dengan cara yang sama, seorang HIPS tidak boleh membiarkan aplikasi yang diizinkan untuk dikompromikan oleh sebuah serangan. Dengan demikian, produk HIPS

harus memiliki kemampuan untuk melindungi terhadap kekurangan dalam aplikasi diperbolehkan.

Internet menghadap web *server*, misalnya, diizinkan untuk menerima koneksi dari *host* yang tidak dikenal di *Internet*. Yang membuatnya lebih mudah bagi penyerang untuk mengambil keuntungan dari setiap kelemahan dalam aplikasi *server web*. HIPS harus memungkinkan *web server* untuk menerima koneksi dari *Internet*, tetapi juga mencegah web *server* dari gangguan [11].

2.4. Virtual machine (VM)

Virtual machine merupakan *software implementation* dari sebuah mesin yang mana mampu menjalankan program layaknya mesin fisik [15]. *Virtual machine* terbagi menjadi 2 kategori utama yaitu: *system virtual machine* yang mana terdiri dari mesin yang lengkap sehingga mampu mejalankan system operasi, dan *process virtual machine* yang hanya mampu menjalankan beberapa program dalam satu mesin. Karateristik yang penting pada *virtual machine* adalah keterbatasan *resource* yang mampu dijalankan pada program yang ada didalamnya. Program tersebut tidak akan mampu keluar dari *resource* yang telah dibatasi oleh *virtual machine* itu sendiri.

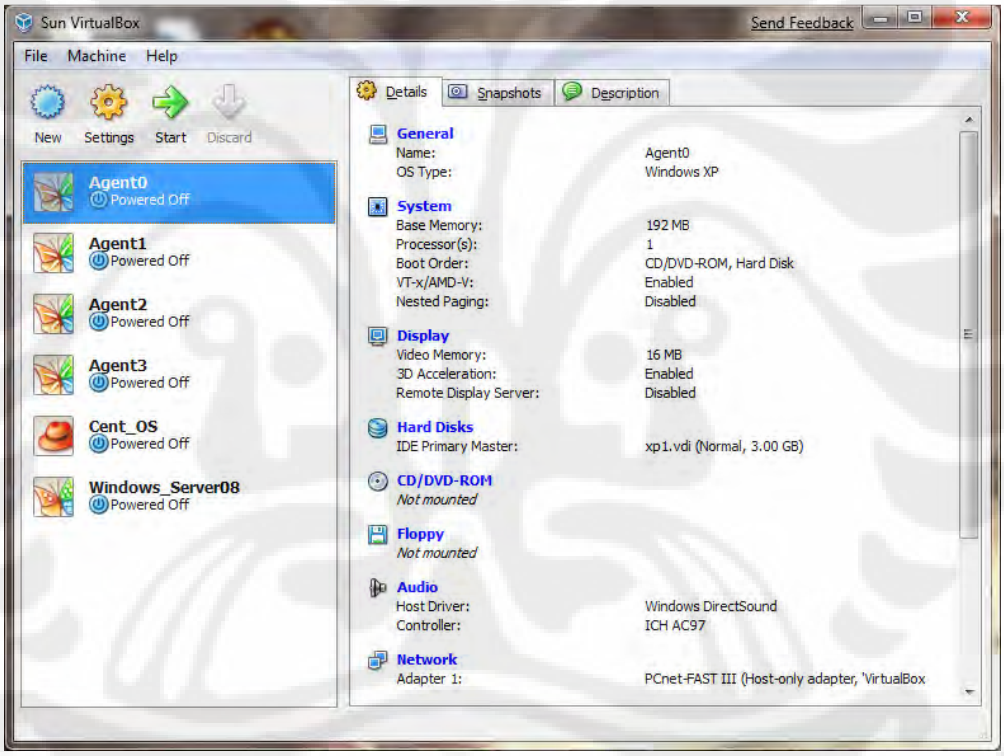
System virtual machine sering disebut dengan *hardware virtual machine* karena VM tersebut mampu menjalankan program besar bahkan operating system didalamnya. VM ini mampu menjalankan beberapa program atau OS dalam satu komputer. Program tersebut memiliki lingkungan sendiri yang terpisah *host*-nya ataupun VM lainnya. VM ini sering dimanfaatkan untuk melakukan *testing* terhadap *software* yang baru dibuat.

Process virtual machine sering disebut dengan *application virtual machine*. VM ini berjalan pada OS dan hanya mampu menjalankan satu proses saja.VM ini ada ketika proses tersebut dibuat dan kemudian hilang ketika proses tersebut tiada. Tujuan dari proses *virtual machine* ini adalah untuk memberikan

platform lingkungan yang berbeda untuk menjalankan program yang hanya mampu berjalan pada *platform* lingkungan tersebut.

Berikut ini contoh aplikasi-aplikasi *virtual machine* yang sering digunakan pada jaringan public.

- VMware
- *VirtualBox*
- *Virtual*PC
- Xen



Gambar 2.11 Contoh gambar salah satu *virtual machine* yang sering digunakan, Sun xVM-*VirtualBox*

VirtualBox merupakan salah satu *tools* yang nantinya akan digunakan pada skripsi ini. *VirtualBox* mampu menjalankan beberapa *virtual machine* dengan beberapa *Operating System* didalamnya.

2.5. *Network* Security

Kriptografi adalah suatu metode yang dapat digunakan untuk menyimpan informasi rahasia dan untuk memastikan integritas dan keaslian [8]. Semua sistem kriptografi modern didasarkan pada prinsip Kerckhoff dikenal memiliki publik algoritma dan sebuah kunci rahasia. Banyak algoritma kriptografi menggunakan transformasi kompleks yang melibatkan substitusi dan permutasi untuk mengubah *plaintext* ke dalam *ciphertext*. Namun, jika kriptografi kuantum dapat dibuat secara nyata, penggunaan pads satu kali dapat memberikan *cryptosystems* benar-benar tidak bisa dipecahkan [9].

Algoritma kriptografi dapat dibagi menjadi *Symetric key algorithm* dan *Asymetric key algorithm* yang memanfaatkan public key untuk memecahkan kodenya. *Symetric key algorithm* merubah bit dalam serangkaian putaran yang berparameter oleh kunci tersebut untuk mengubah *plaintext* ke dalam *ciphertext*. Triple DES dan Rijndael (AES) adalah *Symetric key algorithm* yang paling populer saat ini. Algoritma ini dapat digunakan dalam modus kode buku elektronik, *cipher block chaining mode, stream cipher mode, counter mode*, dan lain-lain.

Asymetric key algorithm atau yang sering disebut *public key algorithm* memiliki properti dengan menggunakan kunci yang berbeda untuk enkripsi dan dekripsi dan kunci dekripsi tidak dapat diturunkan dari kunci enkripsi [9]. Properti ini memungkinkan untuk men-*generate* kunci publik. Algoritma kunci publik yang utama adalah RSA, yang berdasarkan dari fakta mempunyai kekuatan yang sangat sulit dipecahkan untuk faktor dalam jumlah besar.

Manajemen kunci publik dapat dilakukan dengan menggunakan sertifikat, yang merupakan dokumen yang mengikat seorang pelaku pada sebuah kunci publik. Sertifikat ini ditandatangani oleh orang yang mempunyai otoritas yang terpercaya atau oleh seseorang (secara rekursif) disetujui oleh otoritas yang terpercaya.

Metode kriptografi ini dapat digunakan untuk mengamankan *traffic* jaringan. IP Security beroperasi di lapisan jaringan, enkripsi paket mengalir dari *host* ke *host*. *Firewall* bisa menyaring *traffic* akan masuk atau keluar dari suatu jaringan, sering kali berdasarkan *protocol* dan port yang digunakan. Jaringan VPN (*Virtual Private Network*) dapat mensimulasikan jaringan *leased-line* untuk menyediakan keamanan yang diinginkan oleh sistem tertentu [8]. Akhirnya, jaringan nirkabel memerlukan keamanan yang baik dan 802,11 's WEP belum menyediakan hal itu, walapun begitu 802.11i tetap harus jauh lebih ditingkatkan.

Ketika dua pihak mendirikan sebuah sesi, mereka harus mengotentikasi satu sama lain dan jika perlu, membuat *session key* untuk melakukan *sharing* koneksi. Berbagai *protocol* otentikasi yang ada termasuk beberapa yang menggunakan pihak ketiga yang terpercaya adalah *Diffie-Hellman, Kerberos,* dan *public-key cryptography* [9].

Keamanan web juga merupakan topik yang penting, dimulai dengan *secure address*. DNSsec menyediakan metode untuk mencegah DNS *spoofing,* seperti nama-nama sertifikasi diri. Sebagian besar e-commerce situs Web menggunakan SSL untuk mengamankan jaringan mereka, authentikasi sesi antara klien dan *server*. Berbagai teknik digunakan untuk menangani kode mobile, terutama *sandboxing dan Signing Code [13]*.

BAB 3

PERANCANGAN TOPOLOGI JARINGAN

Pada bab ini menjelaskan mengenai perancangan jaringan dan instalasi *software* yang diperlukan untuk melakukan simulasi DoS dan DDoS. Peralatan yang dibutuhkan meliputi *software VirtualBox, Windows XP, Flooding Tools* dan *PC* dengan spesifikasi cukup. Pada bab ini nantinya akan dijelaskan diagram alir serangan DDoS yang dilakukan.

3.1. DOS (*Denial of service*) & DDOS (*Distributed denial of service*)

Tujuan dari serangan DoS adalah untuk mengganggu beberapa kegiatan yang *legal*, seperti *browsing* halaman Web, mendengarkan *radio online*, *transfer* uang dari bank, atau bahkan *system* komunikasi kapal yang akan berlabuh ke pelabuhan. Efek *Denial of service* ini terjadi dengan cara mengirimkan pesan ke target yang mengganggu operasi, dan membuatnya *hang, crash, reboot*, atau melakukan pekerjaan yang sia-sia.

Salah satu cara untuk mengganggu operasi yang *legal* yaitu mengeksploitasi kelemahan yang ada pada mesin target atau sasaran di dalam aplikasi. Penyerang mengirim beberapa pesan yang dibuat dalam cara tertentu yang mengambil keuntungan dari kelemahan *system* yang diberikan. Cara lainnya adalah dengan mengirim sejumlah besar pesan yang mengkonsumsi sumber daya beberapa utama pada target seperti *bandwith, CPU time, memory*, dll. Aplikasi target, mesin, atau jaringan menghabiskan seluruh sumber daya kritis dalam menangani serangan tersebut dan tidak dapat menangani klien yang *legal*.

Tentu saja, untuk menghasilkan seperti sejumlah besar pesan, penyerang harus mengendalikan mesin yang sangat kuat dengan prosesor yang cukup cepat serta *bandwidth* besar pada jaringan yang tersedia. Agar serangan berhasil, penyerang harus membebani sumber daya target. Ini berarti bahwa mesin penyerang harus mampu menghasilkan lebih banyak *traffic* dari target, atau infrastruktur jaringan yang lebih besar yang mampu ditangani target.

Sekarang mari kita asumsikan bahwa penyerang ingin meluncurkan serangan DoS *www.example.com* dengan membombardir dengan berbagai pesan. Asumsikan juga *Example.com* mempunyai sumber daya yang melimpah, maka sulit bagi penyerang untuk menghasilkan sejumlah pesan yang cukup dari satu mesin untuk membanjiri sumber daya tersebut. Namun, misalkan penyerang mempunyai kendali lebih dari 100.000 mesin dan melibatkan mereka dalam menghasilkan pesan untuk *example.com* secara bersamaan. Masing-masing mesin penyerang sekarang mungkin hanya mempunyai sumber daya yang sedikit (misalnya, memiliki prosesor lambat dan berada pada *link modem*) tetapi bersama-sama mereka membentuk jaringan serangan yang hebat, dengan penggunaan yang tepat maka serangan tersebut mampu melakukan *overload* pada korban. Ini yang disebut dengan *Distributed denial of service - DDoS.*

Agar lebih jelas akan diterangkan proses serangan *Denial of service* dan *Distributed denial of service* pada Sub Bab berikut ini.

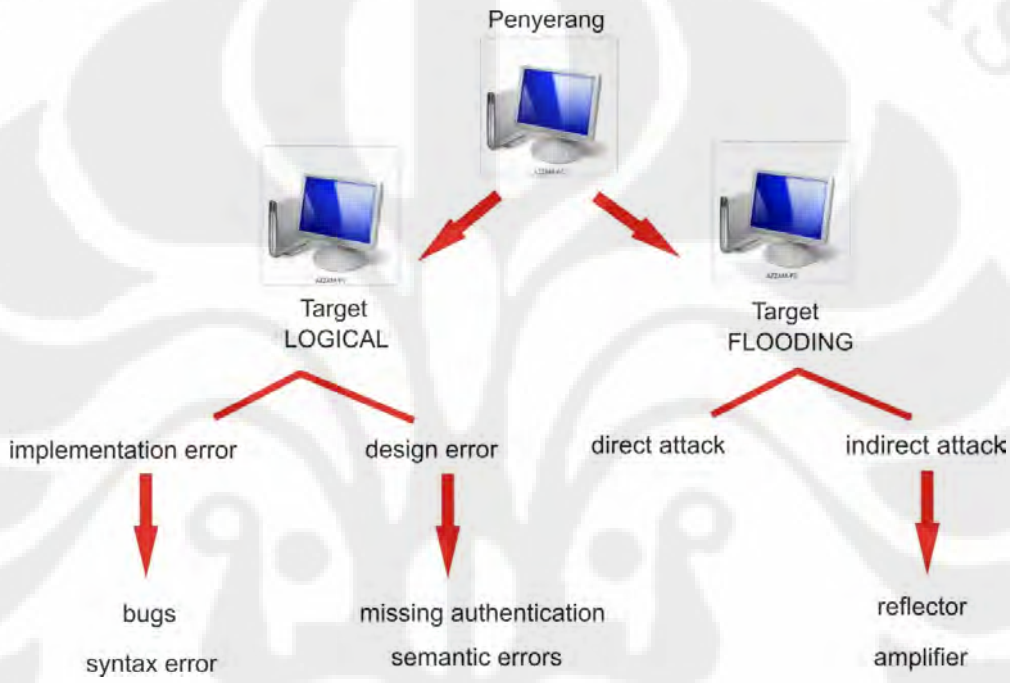
3.1.1. DoS (*Denial of service*)

Denial of service merupakan serangan yang dilakukan secara *individua*l menggunakan satu mesin komputer. Biasanya serangan ini dapat dijalankan ketika mesin penyerang lebih kuat dibandingkan dengan targetnya, sehingga penyerang mampu membanjiri targetnya dengan paket-paket yang dia kirimkan. Kuat yang dimaksud pada definisi tersebut adalah dalam hal besarnya *Bandwith*, Kecepatan *Processor*, dan kapasitas *memory*. Jika mesin komputer penyerang lebih lemah dibandingkan dengan targetnya maka akan terjadi sebaliknya. Penyerang tidak mampu melakukan koneksi karena jaringannya penuh dengan paket yang dia kirimkan kepada penyerang. [2]

Denial of service mempunyai beberapa tipe serangan. Berikut ini akan dijelaskan mengenai tipe-tipe serangan *Denial of service*.

Tipe-tipe serangan DoS ada 2 yaitu:

1. *Logical*, merupakan tipe serangan yang memanfaatkan kelemahan aplikasi, Operating System atau kesalahan *Syntax* pada mesin target. Contohnya: SMBNUKE
2. *Flooding*, merupakan tipe serangan yang menggunakan *protocol* TCP, UDP, atau ICMP untuk membanjiri targetnya dengan paket-paket *request* yang dikirimkan oleh penyerang. Contohnya: *Tcp-Flood*

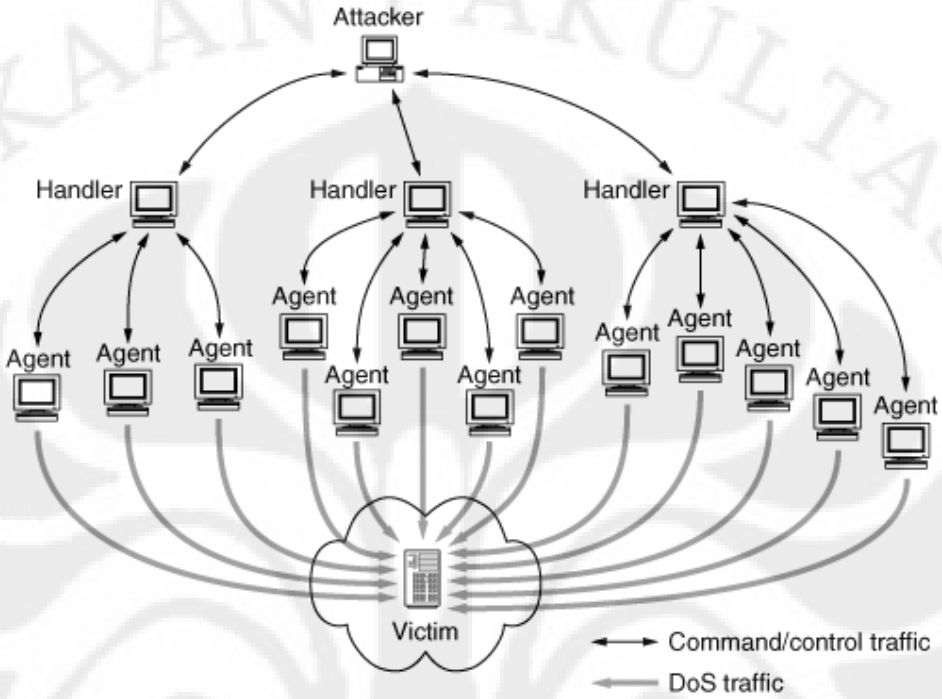


Gambar 3.1. Tipe serangan DoS dan DDoS yaitu: *Logical dan Flooding*

3.1.2. DDoS (*Distributed denial of service*)

Pada dasarnya serangan ini tidak jauh berbeda dengan DoS hanya saja serangan ini dilakukan secara simultan dengan banyak mesin. Pada serangan ini logikanya penyerang melakukan kompromisasi dengan mesin-mesin yang telah dia tanam dengan *malicious application*, sehingga penyerang mampu melakukan serangan secara bersama-sama dari beberapa titik kepada satu target yang telah ditentukan. Dalam hal ini mesin penyerang tidak harus mempunyai *bandwith* yang tinggi untuk melakukan serangan DDoS karena jumlah *resource* dari beberapa komputer penyerang mampu melebihi batas *resource* yang dimiliki oleh Target atau korban.[2]

Berikut ini gambaran serangan DDoS yang dikendalikan oleh satu mesin untuk melakukan *flooding*.



Gambar 3.2 Skema serangan *Distributed denial of service*

Pada survey yang dilakukan oleh Cisco serangan DoS dan DDoS sering dilakukan menggunakan *protocol* TCP. Penyerang memanfaatkan *3 way handshake* pada *protocol* ini untuk meminta *request* paket yang banyak sehingga menghasilkan *traffic* yang padat. Ketika jaringan target terjadi *flooding* dan *down* maka serangan tersebut dianggap berhasil. Maka dari itu diperlukan mekanisme pertahanan yang mampu menghentikan serangan berbasis ini.

Pada tugas akhir ini saya bermaksud membuat sebuah *topology* jaringan yang mampu merepresentasikan serangan DoS dan DDoS. Tujuan dari *topology* ini adalah untuk mencari mekanisme pertahanan dari serangan DoS dan DDoS yang baik. Untuk mensimulasikan serangan DOS dan DDOS diperlukan *topology* yang cukup agar mampu mewakili jaringan yang sebenarnya. Pada subbab berikut akan dijelaskan rancangan yang dilakukan untuk melakukan serangan DoS dan DDoS.

3.2.Peralatan yang dibutuhkan

Pada bagian ini akan dijelaskan mengenai alat-alat dan *software* yang dibutuhkan untuk membuat simulasi ini. Peralatan utama yang dibutuhkan adalah sebagai berikut ini:

3.2.1. Personal Komputer

PC yang digunakan pada percobaan ini mempunyai spesifikasi sebagai berikut: AMD X2 4400 2.4 Ghz, DDR2 2Gb, HDD 320Gb, Ati 4670 512Mb

3.2.2. Pantech WWAN *Control*ler Modem *Broadband*

Modem ini merupakan modem *internet* dengan koneksi *Broadband*. Modem ini menggunakan jaringan *wireless* sehingga *response time* pada paket ICMP cukup besar yaitu nilainya berkisar 100 ms (kondisi normal). *Internet* ini disediakan oleh Mobi dari mobile 8 dengan *bandwith* maksimum 1 Mbps.

3.2.3. *VirtualBox*

VirtualBox merupakan aplikasi yang mampu membuat *virtual machine* dimana *virtual machine* tersebut dapat diinstall dengan *Operating System*. *VirtualBox* merupakan *software* yang di produksi dari Sun Microsystems dan lisensi yang digunakan *freeware* atau gratis.

3.2.4. IDS Center dan SNORT

SNORT merupakan aplikasi yang berfungsi sebagai pertahanan jaringan dalam suatu system ataupun dan suatu pc. *Software* ini juga mempunyai lisensi yang *freeware* atau gratis. *Software* ini berhubungan dengan *IDS center* dan *WinpCap* agar dapat beroperasi dengan baik.

3.2.5. WinArp*Attack*er

WinArpAttacker merupakan *tools* yang mampu men-*generate* paket tcp, arp, dan udp. *Tools* ini mampu membanjiri targetnya dengan paket-paket tersebut hingga targetnya *down*.

3.3.Proses Instalasi Jaringan Menggunakan *VirtualBox*

Uji coba dapat dilakukan menggunakan jaringan yang nyata ataupun dengan simulasi. Uji coba pada jaringan nyata biasanya menghasilkan data yang lebih mendekati pada kebenaran. Uji coba simulasi biasanya terkendala akan *traffic* jaringan yang tidak nyata. Saat ini dicoba menggunakan simulasi yang mendekati kondisi nyata yaitu dengan menggunakan *Virtual*isasi OS yang dijalankan pada waktu yang sama dan dikoneksikan ke jaringan *internet*. Dengan kondisi tersebut dapat dihasilkan data-data yang mendekati kenyataan. Untuk membuat jaringan dalam *virtual*isasi OS diperlukan peralatan dan cara yang tidak sederhana. Berikut ini merupakan langkah-langkah membuat jaringan dengan menggunakan *Virtual OS*.

1. Pertama install *VirtualBox* sebagai media untuk menjalankan *virtual OS*.
   Berikut ini gambaran singkat instalasi *VirtualBox*

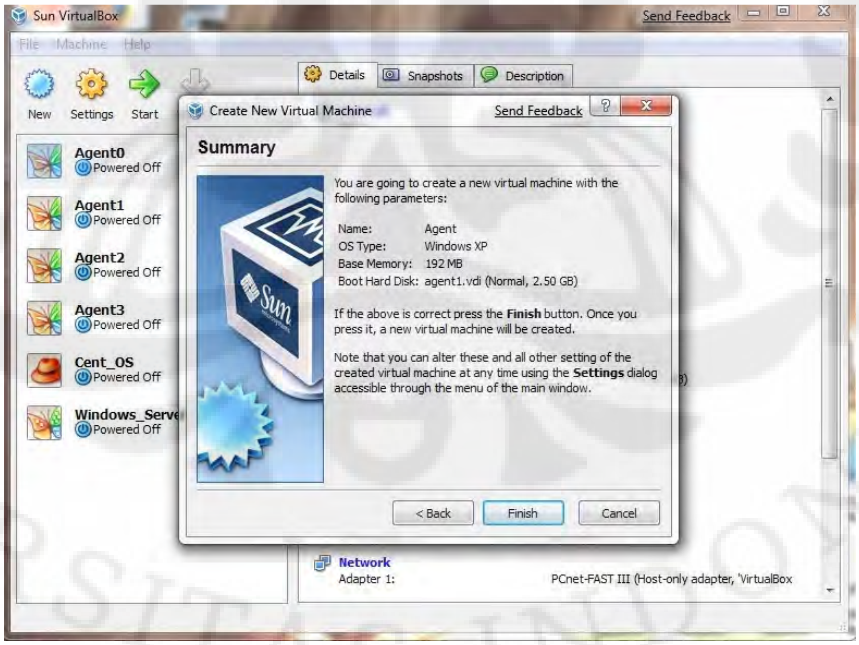
Gambar 3.3 Instalasi *VirtualBox* sebagai media *Virtual machine*

Proses instalasi dilakukan dengan mengikuti langkah-langkah yang diberikan pada *software installer.*

2. Setelah proses instalasi *VirtualBox* berakhir dilanjutkan dengan proses pembuatan *VirtualOS* dengan setting *network*nya.

Berikut ini langkah utama untuk men-*install Virtual machine:*

Gambar 3.4. Proses pembuatan *Virtual machine* pada VB

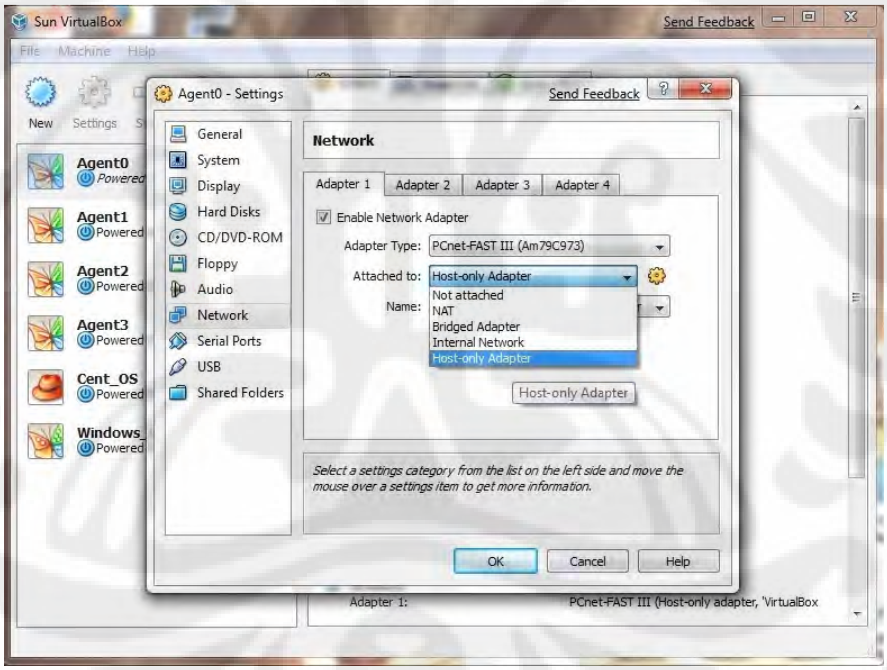


Gambar 3.5. Proses pembuatan *Hardisk Virtual*

Langkah –langkah utama membuat *Virtual Machine* terbagi menjadi 2 yaitu:

- Membuat mesin komputer dengan mengalokasikan sumber daya fisik (*hardware) seperti Ram, Video, dan Processor*.
- Membuat *virtual* hardisk yang berfungsi sebagai media penyimpanan *VirtualOS* tersebut.

3. Setelah keempat mesin yang dibutuhkan telah terinstall, masukan CD *installer OS* yang akan di install kemudian lakukan penginstalan OS pada keempat mesin tersebut.
4. Pada setiap mesin disetting jaringan menggunakan *network-card* yang sama agar mudah untuk saling mengenali. Berikut ini setting untuk membuat semua mesin tersebut tergabung menjadi satu jaringan nyata.



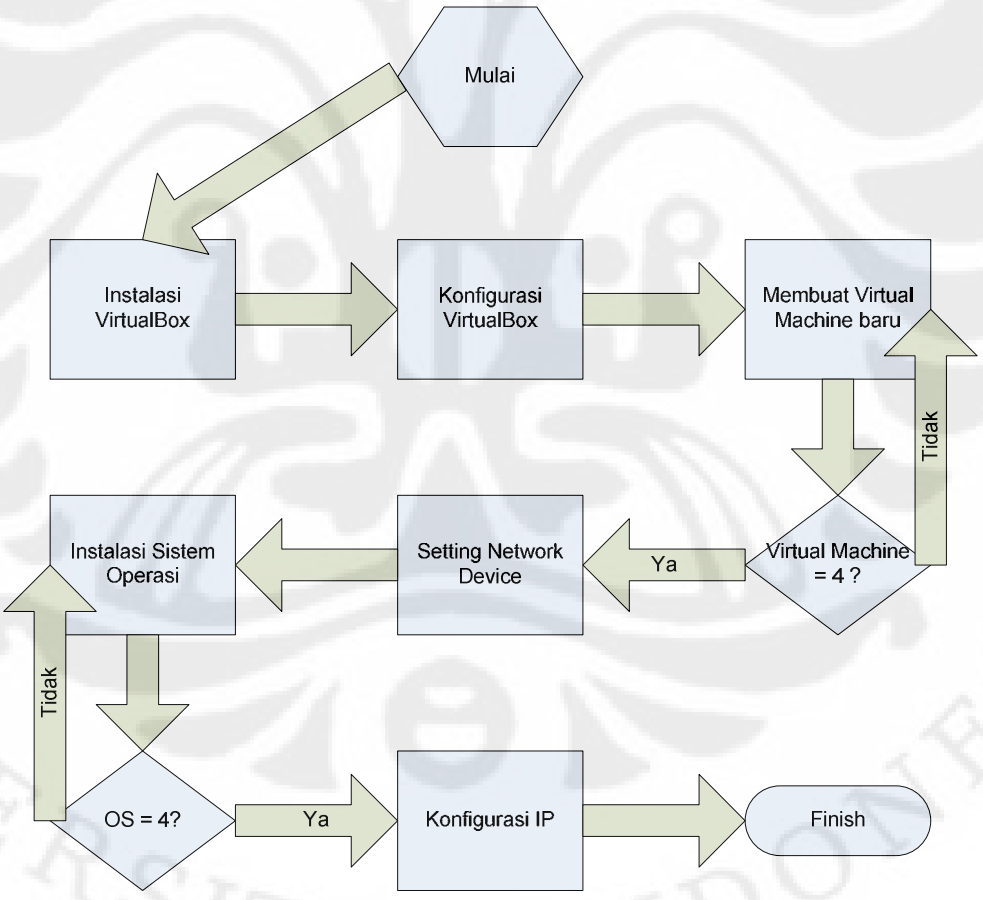
Gambar 3.6. Konfigurasi *Host*-Only *Adapter* sebagai *Ethernet* yang menghubungkan VM ke *Host*

- *Host-only adapter* berfungsi sebagai *Ethernet card* yang menghubungkan *Host* dengan *Virtual OS*. Setting ini dilakukan untuk semua mesin agar semua mesin terkoneksi dengan *Host*-nya.

- Pilih *VirtualBox Host-Only Ethernet adapter* sehingga PC *host* mampu mengenali semua *Virtual OS* yang telah dibuat sebelumnya.

5. Langkah kelima yaitu memberikan konfigurasi *IP address, Subnet Mask* dan *DNS server* agar setiap *Virtual OS* tersebut mampu melakukan koneksi *internet* melalui *PC-host.*
6. Setelah semua *Virtual OS* terkoneksi dengan baik langkah terakhir adalah *testing*. *Testing* dilakukan agar menjamin semua jaringan mendapatkan *bandwith* yang sama.

Berikut ini adalah blok diagram untuk keseluruhan proses instalasi topologi jaringan pada *VirtualBox.*



Gambar 3.7. Diagram alir pembuatan topologi jaringan menggunakan *Virtual machine*

Bagan diatas diawali dengan melakukan instalasi *VirtualBox* yang dapat dilihat. Pada pembuatan *virtual machine* dilakukan 4 kali sesuai dengan jumlah *virtual OS* yang akan dibuat. Setelah *virtual machine* dibuat langkah selanjutnya *men-install OS* yang akan digunakan dalam hal ini Windows XP. Setelah semua OS terinstall, langkah terakhir adalah memberikan konfigurasi *IP Address* pada setiap *virtual OS*.

3.4.Topologi Jaringan Utama

Seperti yang telah dijelaskan sebelumnya dalam tugas akhir ini digunakan *VirtualBox* sebagai alat untuk melakukan simulasi serangan DoS dan DDoS ini. Berikut ini *tools* yang digunakan untuk membuat simulsi jaringan DoS dan DDoS:

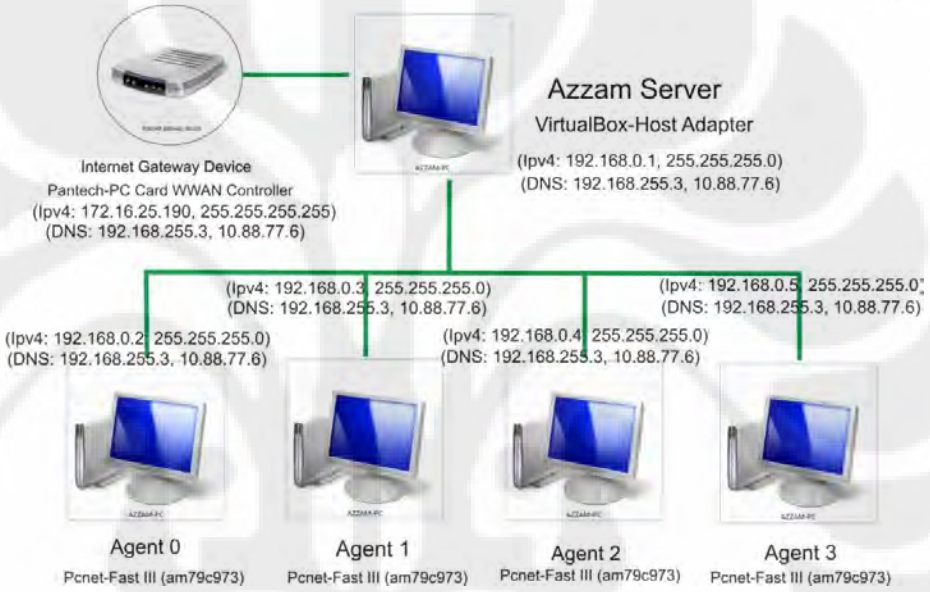
- *VirtualBox*, sebagai *virtual*isasi OS yang mampu menjalankan beberapa OS dalam 1 PC dan membuat jaringan didalammnya.
- WindowsXP, sebagai OS yang menjadi media *agent* untuk melakukan serangan.
- Windows7, sebagai media *Host* yang berperan sebagai *server* dan menghubungkan jaringan *agent* ke *internet*.
- PC dengan Spesifikasi, AMD X2 2.4Ghz, 2GB Ram, 320GB HDD, 512MB HD 4670, 8 slot USB, Pantech-PC-*card* Modem, Mobi *Broadband*.
- *Vulnerability tools* seperti; Ip Scanner, Port Scanner, TCP *Flooding*, DDOS-Ping, Snifer-PRO, WinArp*Attack*er.
- *Network Defense Tools* seperti ; IDS center, SNORT, *Firewall*, Avira antivirus.

Simulasi dilakukan menggunakan alat-alat tersebut berdasarkan skenario yang telah dibuat. Skenario dibuat berdasarkan parameter yang akan dihitung dalam percobaan simulasi tersebut. Berikut ini merupakan parameter-parameter yang ditentukan untuk melakukan perhitungan pertahanan dalam serangan DoS dan DDoS.

- *Reliability*, seberapa besar peluang mekanisme pertahanan berhasil dilakukan

- *Response time*, seberapa cepat mekanisme pertahanan tersebut mampu mengenali serangan DoS dan DDoS.

Dari parameter-parameter tersebut dihitung *performance tools* dan mekanisme pertahanan sesuai dengan topologi jaringan.. Setelah semua *tools* dan *software* terinstall dengan baik, topologi jaringan baru bisa dibuat. Berikut ini merupakan *topology* yang dibuat untuk men-simulasikan serangan DoS dan DDoS.

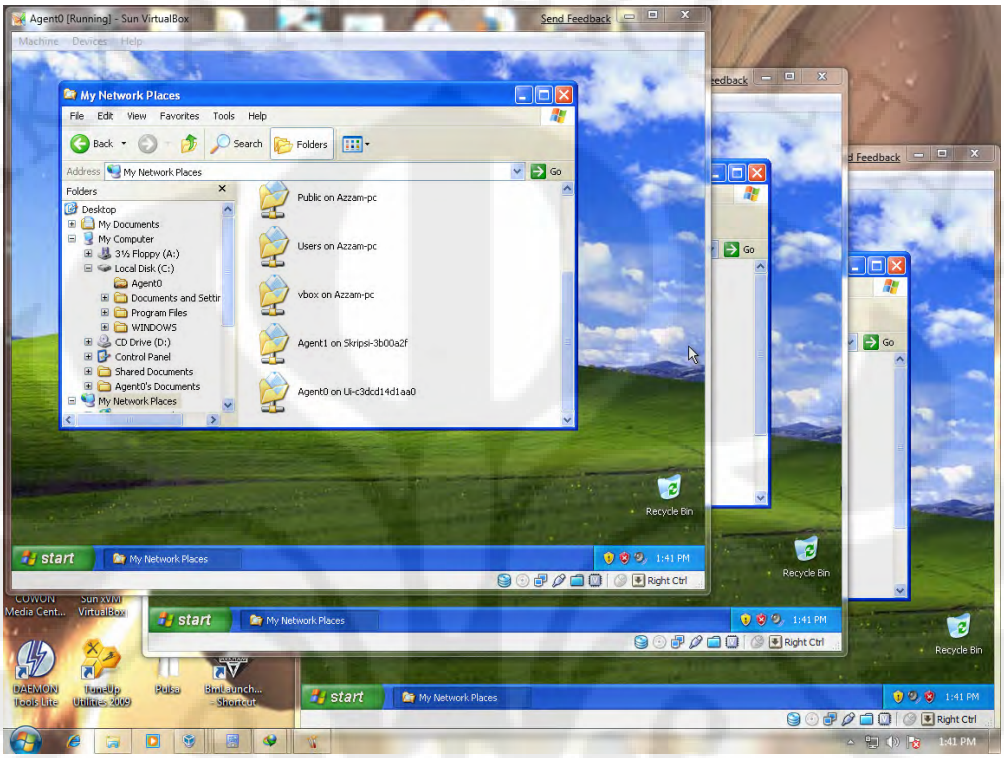


Gambar 3.8. Topologi jaringan yang telah dibuat untuk melakukan simulasi

Gambar tersebut merupakan topologi jaringan utama. Nantinya ada beberapa *flow* diagram yang menggambarkan skenario yang dilakukan agar mendapatkan data-data yang diperlukan. Data-data tersebut dihitung berdasarkan parameter yang dilakukan pada BAB 4.

Dengan topologi jaringan tersebut PC *server* berfungsi sebagai *host* yang menjembatani *agent-agent* yang ada dibawahnya ke jaringan *internet*. Uji coba nantinya akan dilakukan dengan memilih satu komputer sebagai target atau korban.

Berikut ini merupakan gambaran *Virtual machine* bersama *Host* yang telah ter-install dan menjadi jaringan yang satu.



Gambar 3.9. Tampilan *host* dan *Virtual OS* yang berjalan secara simultan dalam satu PC

Gambar diatas merupakan visualisasi *Host* dengan Windows Xp sebagai *Virtual OS* didalamnya. Empat *Virtual OS* tersebut nantinya akan dijalankan secara bersama-sama untuk membentuk satu jaringan yang nyata.

3.5. Skenario Diagram

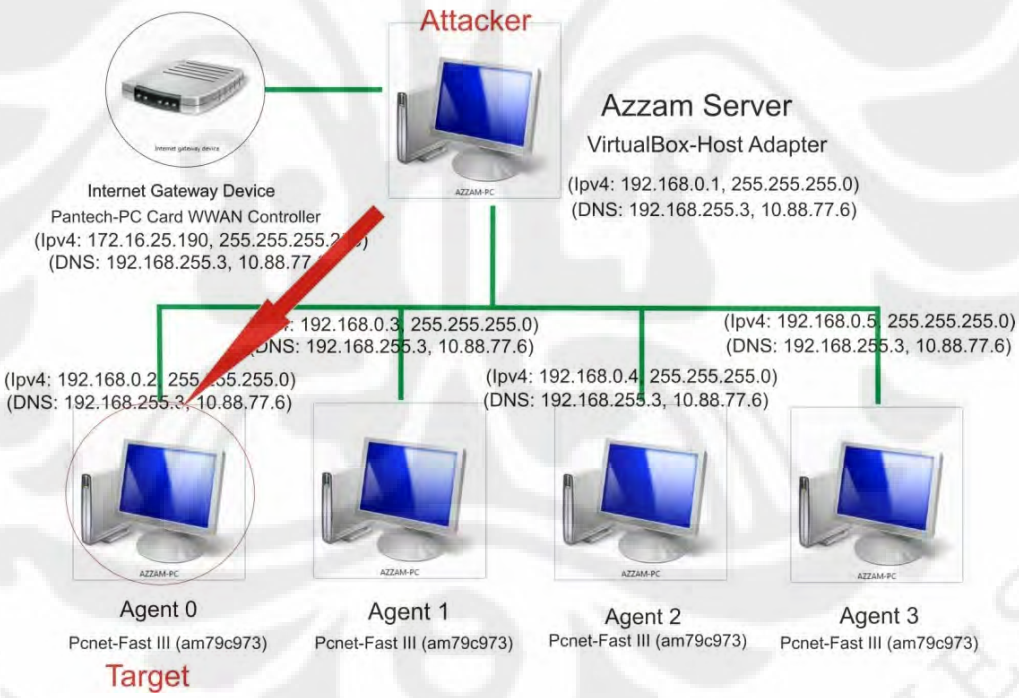
Pada bagian ini akan dijelaskan mengenai rancangan skenario yang dibuat untuk melakukan serangan DoS dan DDoS pada jaringan *virtual* yang telah dibuat. Seperti yang telah dijelaskan sebelumnya tipe dari serangan DoS dan DDoS terbagi menjadi dua yaitu *Logical* dan *Flooding*. Berdasarkan hasil Survey dari Cisco sebagian besar serangan DDoS merupakan tipe *Flooding* dengan memanfaatkan TCP *protocol* SYN *request*. Maka dari itu pada Bab ini akan dirancang jaringan yang memanfaatkan TCP sebagai metode serangan.

Rancangan jaringan ini dibagi menjadi 2 skenario:

1. Skenario pertama adalah serangan DoS menggunakan TCP *Flooding* sebagai *attack tools* dan menggunakan IDS Center sebagai *network defense tools*.
2. Skenario kedua adalah serangan DDoS menggunakan TCP *Flooding* sebagai *attack tools* dan menggunakan IDS *Center* + SNORT sebagai *defense tools*.

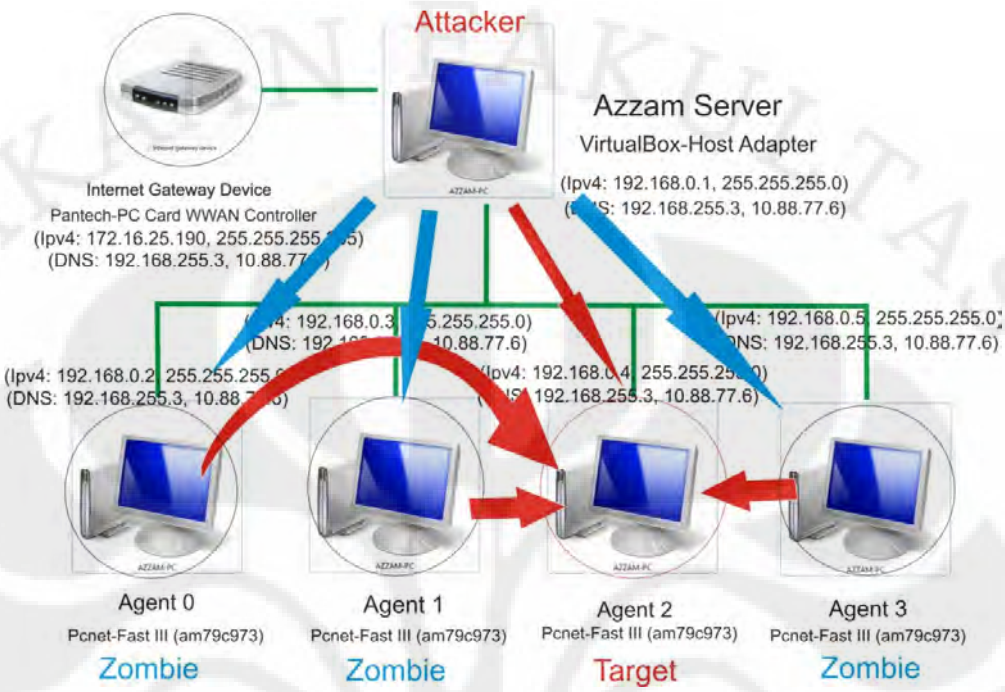
3.5.1. Skenario DoS

Dari dua skenario tersebut akan dihitung parameter-parameter yang telah ditentukan sebelumnya pada Sub Bab 3.3 untuk memperjelas gambaran dari skenario tersebut berikut ini gambar skenario yang telah dibuat.



Gambar 3.10. Skenario simulasi *Denial of service*

3.5.2. Skenario DDoS



Gambar 3.11. Skenario simulasi *Distributed denial of service*

Berikut ini teori perhitungan parameter-parameter yang digunakan untuk menentukan keberhasilan mekanisme pertahanan pada serangan DoS dan DDoS.

Reliability

Pada setiap skenario dilakukan 10 kali percobaan serangan DoS dan DDoS menggunakan TCP *Flooding tools*. IDS dan SNORT akan di konfigurasi berdasarkan fase *prevention* dan fase *termination*. Hasilnya akan diukur dengan prosentase yang dihitung dengan ***%Reliability* = Jumlah Serangan yang mampu di Blok / 10**.

Response time

Untuk mendapatkan perhitungan *response time* dilakukan dengan cara melakukan koneksi ke *internet*. Sebagai acuan koneksi *internet* dilakukan dengan

cara melakukan ping ke *www.google.com*. *Response time* didapat dari persamaan **RT=∑ (0-100) RTT /100.**

3.6. Hipotesa mekanisme pertahanan DoS dan DDoS

Pertahanan dalam serangan DoS dan DDoS tidaklah mudah. Ada beberapa langkah yang harus dilakukan untuk mengatasi serangan ini. Hal ini telah dibahas pada dasar teori di bagian jaringan *komputer* dan *network security*. DoS dan DDoS memanfaatkan kelemahan yang ada pada jaringan target. Dengan mengetahui kelemahan tersebut mereka mampu melakukan serangan yang mematikan kepada jaringan kita. Maka langkah pertama untuk melakukan pertahan terhadap serangan DoS dan DDoS adalah dengan mengetahui kelemahan dari jaringan kita sendiri.

Kelemahan pada jaringan kita dapat dideteksi menggunakan berbagai *tools* yang telah tersedia di *Internet*. Ketika kelemahan jaringan yang ada mampu diketahui maka kita bisa melakukan antisipasi sebelum serangan itu terjadi. Mekanisme pertahanan yang seperti ini bisa disebut sebagai *prevention phase. Prevention phase* adalah *fase* mekanisme pertahanan yang dilakukan sebelum terjadinya serangan.

Ketika suatu saat jaringan kita mengalami masalah yang diakibatkan oleh serangan maka kita harus mampu melakukan deteksi. Deteksi dilakukan agar kita dapat melakukan langkah selanjutnya untuk mempertahankan jaringan kita dari penyerang tersebut. Setelah kita mampu mendeteksi serangan kita harus menentukan kebijakan yang harus dilakukan untuk mengkhiri masalah. Metode pertahanan ini juga sering disebut dengan *Termination Phase Mekanism. Termination Phase Mekanism* yaitu mekanisme pertahanan yang dilakukan untuk menghentikan serangan seperti deteksi dan terminasi.

Gambar 3.12. Skema mekanisme Pertahanan DoS dan DDoS

Kedua mekanisme tersebut sangat penting untuk mempertahankan jaringan atau komputer kita dari serangan DoS dan DDoS. Berikut ini akan dijelaskan tentang *Prevention phase dan Termination phase.*

3.6.1. *Prevention Phase*

Seperti yang telah disebutkan *fase* pertahanan ini dilakukan sebelum terjadinya serangan. Hal ini telah dijelaskan pada dasar teori bagian IPS (*intrusion prevention* sistem). Pertahanan ini diakukan dengan tujuan untuk melindungi jaringan dari ancaman-ancaman yang mungkin terjadi sehingga prosentase serangan yang mungkin terjadi dapat dikurangi. Ada poin-poin yang bisa dilakukan pada fase *prevention* ini yaitu;

- Mengamankan *end-host*

Fase ini dapat dilakukan dengan memberikan tingkat security yang cukup pada end-*host* sehingga mampu meminimalisir terkontaminasinya end-*host* oleh *malicious application*.

- Melindungi *protocol asymmetric*

Protocol asymmetric sering digunakan pada mekanisme perthanan untuk melakukan *decryption* dan *encryption* pada paket yang dikirim atau diterima. *Protocol* ini harus diamankan agar paket yang dikirim melaui *internet* dapat terjaga kerahasiaanya.

- Mengalokasikan sumber daya

Hal ini dapat dilakukan untuk memberikan performa terbaik pada jaringan. Jika sumber daya yang dibutuhkan tidak terpenuhi maka mekanisme pertahanan tidak mampu berjalan dengan baik.

- Menyembunyikan *host*

Cara mengamankan *host* yang paling baik adalah menyembunyikan *host* itu sendiri dari jaringan *internet*. Dengan cara ini hanya orang-orang yang mempunyai kewenangan saja yang mampu mengakses *host* ini. Dengan begitu system ini akan terlindungi dari ancaman serangan di dalam *internet* itu sendiri.

3.6.2. *Termination Phase*

Metode mekanisme pertahanan kedua adalah *fase termination*. *Fase* ini dilakukan ketika jaringan mengalami serangan. Serangan yang terjadi harus segera dihentikan agar menjaga stabilitas jaringan. *Fase* ini terdiri dari 2 mekanisme. Mekanisme yang pertama adalah mekanisme deteksi. Untuk menghentikan ancaman serangan kita harus mengetahui serangan itu sendiri. Dengan mekanisme ini kita dapat mengetahui serangan yang sedang terjadi. Mekanisme ini diakukan untuk menentukan tindakan selanjutnya. Mekanismme yang kedua adalah mekanisme reaksi. Mekanisme ini berfungsi untuk melakukan tindakan yang berkaitan dengan pemberhentian serangan pada jaringan. Ada beberapa jenis dari setiap mekanisme ini.

- Deteksi *signature*

Deteksi *signature* yaitu metode deteksi yang mampu mengenali sumber yang terpercaya dan sumber yang tidak terpercaya.

- Deteksi *anomaly*

Deteksi *anomaly* berjalan ketika terjadi koneksi yang tidak umum yang terjadi pada jaringan tersebut.

- Deteksi perilaku aneh (*missbehaviour*)

Ketika terjadi kondisi yang mengarah pada kondisi aneh maka deteksi ini akan berjalan untuk menghentikan serangan tersebut.

- Mekanisme berbasis filtrasi

Mekanisme ini dilakukan dengan cara men-*filter* semua paket yang masuk pada jaringan. Mekanisme ini mampu menghentikan paket-paket yang mencurigakan. Ini sangat berguna untuk menghentikan serangan DoS maupun DDoS.

- Mekanisme berbasis *congestion control*

Congestion control adalah mekanisme yang mampu menghentikan terjadinya kemacetan pada jaringan. Mekanisme ini mampu untuk menghentikan serangan DoS dan DDoS yang berbasis *flooding.*

- Mekanisme berbasis TCP

Seperti yang telah dijelaskan pada awal Bab 3 ini bahwa TCP merupakan *protocol* yang sering dimanfaatkan *attack*er untuk melakukan serangan terhadap jaringan. Maka dari itu perlu adanya mekanisme sendiri yang mampu menghentikan serangan yang berbasis TCP ini. Mekanisme ini memberikan solusi untuk mempertahankan jaringan dari serangan DoS dan DDoS.

Dengan mengunakan mekanisme-mekanisme tersebut akan dihitung tingkat efesiensi dan performa dari mekanisme tersebut. Tingkat efesiensi dan performa mekanisme tersebut dilihat menggunakan 3 parameter yang telah dijelaskan sebelumnya yaitu *reliability* dan *response time*. Dengan menggunakan *tools* IDS Center dapat dilakukan pendekatan konfigurasi terhadap mekanisme-mekanisme yang telah dibuat tadi. Kemudian disimulasikan pada *topology* jaringan yang telah dibuat. Analisa Data-data dan pembahasan hasilnya akan dibahas pada Bab 4.

BAB 4

ANALISA DAN PEMBAHASAN

Pada bab ini akan diterangkan mengenai proses analisa dan pembahasan sistem pertahanan DoS dan DDoS dengan menggunakan *VirtualBox*. Dari pendekatan mekanisme pertahanan yang dijelaskan pada Bab 3 nantinya akan di implementasikan pada aplikasi IDS, SNORT maupun aplikasi standar seperti *firewall*. Sebelum melakukan analisa tentunya dibutuhkan parameter dan *variable* yang berfungsi sebagai standar maupun pembanding. Pada bab ini juga akan dijelaskan proses perhitungan parameter yang sudah ditentukan pada bab sebelumnya.

Analisa dilakukan pada topologi jaringan di *VirtualBox* berdasarkan perhitungan parameter. Pengukuran parameter pertama menentukan tingkat *reliability* yang dapat tercapai pada mekanisme pertahanan yang telah diterapkan menggunakan IDS dan SNORT. Pengukuran parameter kedua mencari hasil *response time* yang dapat dicapai pada kedua metode *defense* tersebut yaitu *prevention phase* dan *termination phase.* Parameter dihitung menggunakan formula yang sudah ditentukan sebelumnya. Dari hasil parameter tersebut akan dilakukan pembahasan mengenai optimalisasi konfigurasi dan kesimpulan dari hasil penelitian. Berikut ini merupakan penjelasan penggunaan parameter yang telah dibahas pada Bab 3.

4.1. Penentuan Parameter Pengukuran

Parameter pengukuran ditentukan berdasarkan kebutuhan akan penelitian ini. Untuk dapat mengetahui tingkat keberhasilan mekanisme pertahanan DoS dan DDoS pada suatu *system* harus diketahui tingkat *reliability* metode tersebut. Selain itu perlu diketahui seberapa cepat *response time host* yang terkonfigurasi dengan IDS 1.1.rc. Kemudian perlu diketahui juga tingkat kesalahan yang mungkin terjadi pada mekanisme pertahanan tersebut.

Hal-hal tersebut menjadi standar ukuran untuk menentukan parameter pada penelitian kali ini. Sehingga ada 2 parameter utama yang akan digunakan sebagai alat ukur yaitu *reliability*, dan *response time*.

4.2. Penentuan Skenario Serangan

Skenario serangan ditentukan untuk mencari konfigurasi mekanisme pertahanan yang paling optimal pada kondisi jaringan tertentu. Pada penelitian kali ini digunakan 2 skenario serangan. Skenario serangan pertama adalah serangan DoS pada satu target yaitu *Agent*0 dengan penyerangan *Host* dari *VirtualBox* itu sendiri. Skenario kedua adalah serangan DDoS pada jaringan tersebut dengan target *Agent2* dan pnyerangnya semua *Agent* dan *Host VirtualBox*.

Setiap skenario serangan dilakukan dua kali analisa dengan mekanisme pertahanan *prevention phase* dan *termination phase*. Sehingga nanti ada 4 hasil perhitungan dan analisa. Tipe serangan merupakan serangan DoS atau DDoS menggunakan *Direct method*.

Pada penelitian ini diasumsikan topologi jaringan merupakan jaringan LAN menggunakan *virtual machine* yang dijalankan oleh *VirtualBox*. Tetapi pengukuran parameter tetap menggunakan jaringan *internet*. Seperti *response time* akan diukur dengan perhitungan RTT (*round trip time*) ke jaringan *internet* dalam hal ini *www.google.com*. Skenario dibatasi oleh 2 serangan tersebut agar lebih mudah menganalisa serangan dan mekanisme pertahanannya.

Seperti yang telah dijelaskan pada dasar teori bahwasanya *system* pertahanan jaringan dapat menggunakan aplikasi *firewall*. Pada skenario ini *firewall* akan dimanfaatkan sebagai mekanisme untuk pertahanan juga yang masuk pada kategori *prevention phase*.

Mekanisme deteksi dapat menggunakan *port listener* atau aplikasi lainnya yang mampu mengawasi *traffic* yang melalui jaringan kita. Output dari mekanisme deteksi adalah alert. Setelah alert diberikan *system* akan memberikan response sesuai yang telah di konfigurasi pada *termination phase*.

Untuik memperjelas jenis serangan DoS dan DDoS yang dipakai berikut ini tabel tipe-tipe serangan DoS dan DDoS:

Category	Sub-category	Attack name	DDoS condition	Efficient placement of defense mechanisms	Status
Flooding	Direct	TCP SYN Floods	End-point resource Exhaustion Link congestion	End-host and Intermediate network routers	Most popular DoS attack (27%)
Flooding	Direct	UDP Floods	Link congestion	Intermediate network (core or border) routers	2nd popular DoS attack (26%)
Flooding	Direct	ICMP Floods	Link congestion	Intermediate network (core or border) routers	
Flooding	Indirect	TCP Floods	Link congestion	Intermediate network (core or border) routers	
Flooding	Indirect (Amplified)	Smurf attack Fraggle attack	Link congestion End-point resource Exhaustion	Subnet routers	Few networks remain smurfable today
Flooding	Indirect (Amplified)	DNS Amplification Attacks	Link congestion	DNS server	75% of DNS servers still allowed recursion
Flooding	Direct	Low-rate TCP Based Attacks	Reduction of end-to-end flow throughput	Intermediate network (core or border) routers	
Logic attacks		Land attack, Teardrop , etc…	End-point crash	End-host	Invalid when vulnerability is fixed

Gambar 4.1. tabel serangan DoS dan DDoS [9]

Pada tabel tersebut diambil salah satu jenis metode serangan yang sering digunakan yaitu *TCP/SYN Floods*. Dengan menggunakan serangan ini simulasi dijalankan sesuai skenario dan dihitung sesuai dengan parameter-parameter yang telah ditentukan.

4.3. Perhitungan dan Analisa

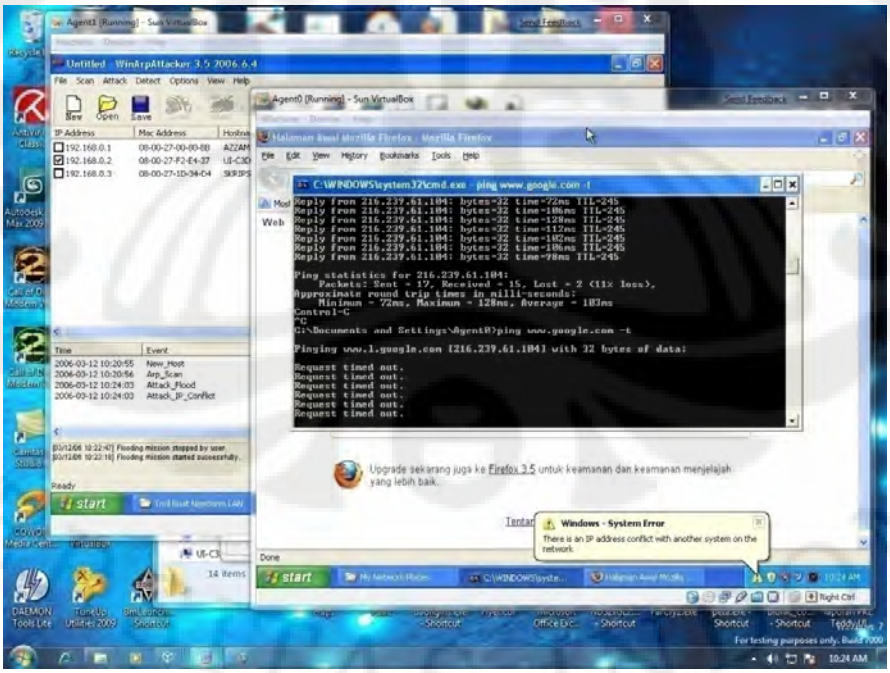
Perhitungan dan analisa dilakukan pada setiap skenario. Parameter yang dihitung adalah *reliability* dan *response time*. Perhitungan pertama akan dilakukan pada skenario 1 yaitu serangan DoS.

4.3.1. Perhitungan Skenario 1

Pada skenario ini akan dilakukan serangan DoS dari *Agent*1 ke *Agent*0 menggunakan *Tools* WinArp*Attack*er 3.50. Dengan menggunakan *tools* ini akan dicoba serangan yang mampu membanjiri target dengan paket TCP dan ARP. *Tools* ini juga mampu men-*generate* paket yang menimbulkan IP *Conflict* pada

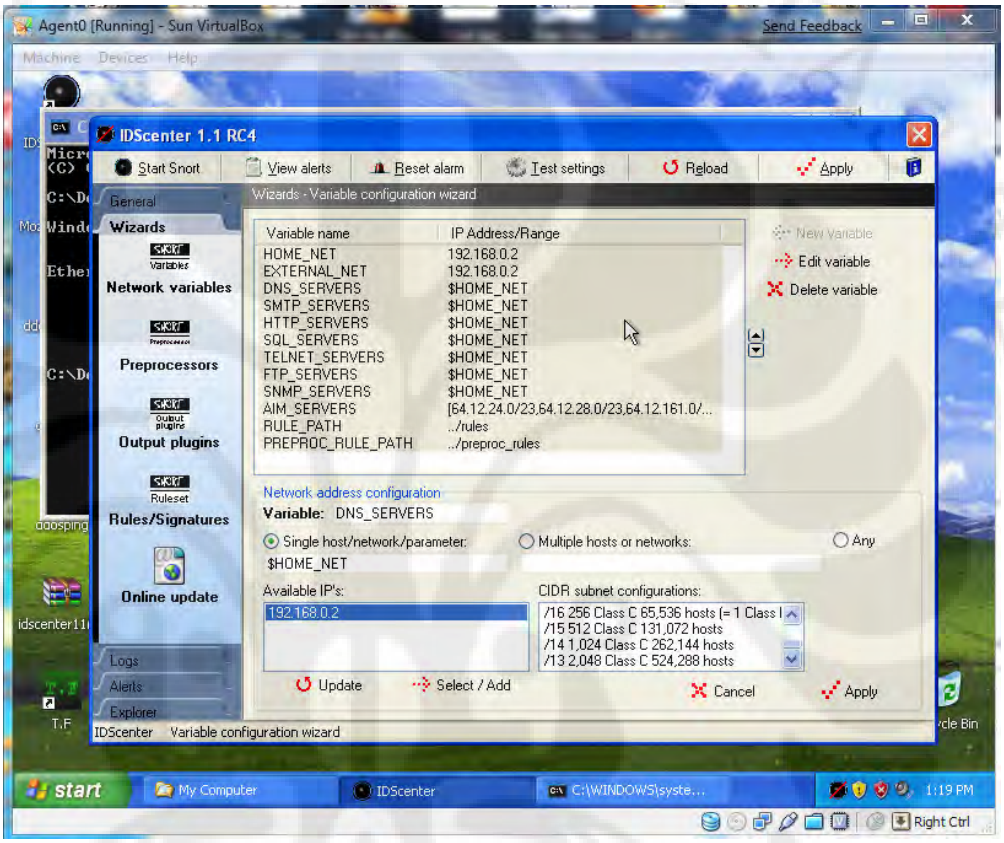
target. Parameter yang dihitung pertama adalah *reliability*. Berikut ini hasil percobaan yang telah dilakukan.

Gambar 4.2. Kondisi sebelum dilakukan serangan DoS pada *Agent*0



Gambar 4.3 Kondisi saat terjadi serangan sebelum dinyalakan IDS

Pada gambar tersebut diterangkan tentang kondisi serangan sebelum dilakukan mekanisme pertahanan. *Agent*0 mengalami *flooding* sehingga tidak mampu melakukan koneksi ke *internet* yaitu *www.google.com*. Dari 100 paket yang dikirim hanya 2 paket yang kembali, artinya terjadi packet loss 98%.

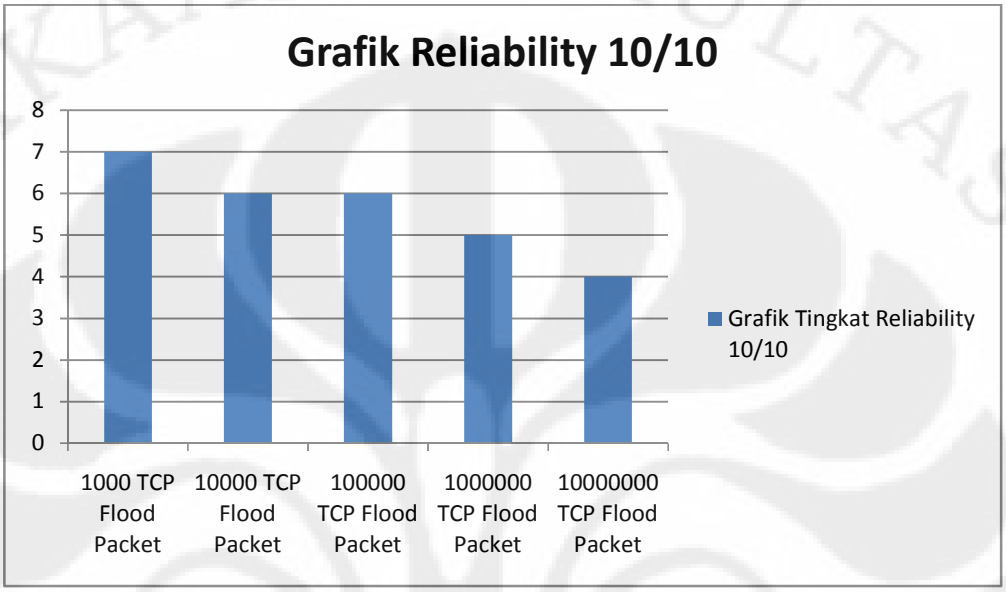


Gambar 4.4 Konfigurasi pertahanan menggunakan IDS Center pada *Agent*0 untuk mencegah serangan.

Perhitungan *Prevention phase* pada Skenario 1

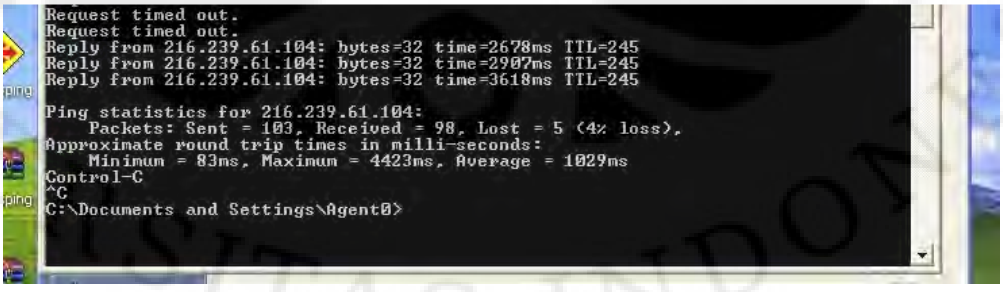
Pada skenario 1 ini *Agent*0 di konfigurasi menggunakan IDS center untuk melakukan pertahanan. Seperti yang sudah dijelaskan pada Bab 3 bagian hipotesa, pada mekanisme pertahanan fase *prevention*dilakukan pendekatan mekanisme yaitu: mengamankan end-*host*, melindungi *protocol*, alokasi *resource* dan menghilangkan *host*. Pada *prevention phase* ini yang akan diimplementasikan adalah mengamankan end-*host* dan melindungi *protocol* yaitu dengan memasang *firewall*, dan konfigurasi IDS agar mampu mencegah serangan. Konfigurasi dapat

dilihat pada gambar 4.4. Setelah dipasang semua perlengkapan yang dibutuhkan diakukan uji coba kembali menggunakan WinArp*Attack*er 3.50. kemudian dihitung parameter *reliability* dan *response time*.



Tabel 4.1. Grafik yang menunjukan hasil percobaan tingkat *reliability* skenario 1 dengan mekanisme *prevention phase*.

Pada percobaan yang telah dilakukan pada skenario 1 menghasilkan tingkat *reliability* yang cukup rendah yaitu (28/5)X10 = 56%. Ini disebabkan karena tidak adanya mekanisme deteksi dan bloking. Ketika DoS *tools* mampu mengirimkan paket pada target mekanisme ini tidak mampu menahan serangan. Mekanisme bloking hanya dijalankan pada fase *termination* dan belum dikonfigurasi pada skenario ini.

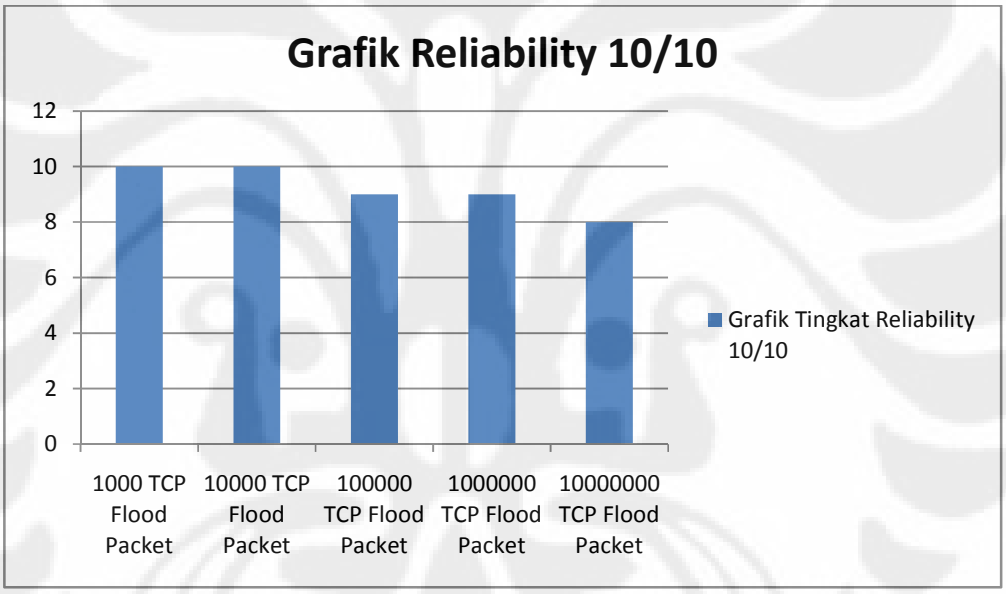


Gambar 4.5. Hasil perhitungan *response time* berdasarkan RTT (*round trip time*) pada paket ICMP.

Pada gambar 4.5. merupakan hasil dari perhitungan *response time* pada satu percobaan. Dari 103 paket yang dikrim terdapat 5 paket yang tidak kembali sehingga packet loss yang terjadi 4%. *Response time* didapat dari rata-rata packet yang diterima. Hasil *response time* setelah dikonfigurasi adalah 1029ms.

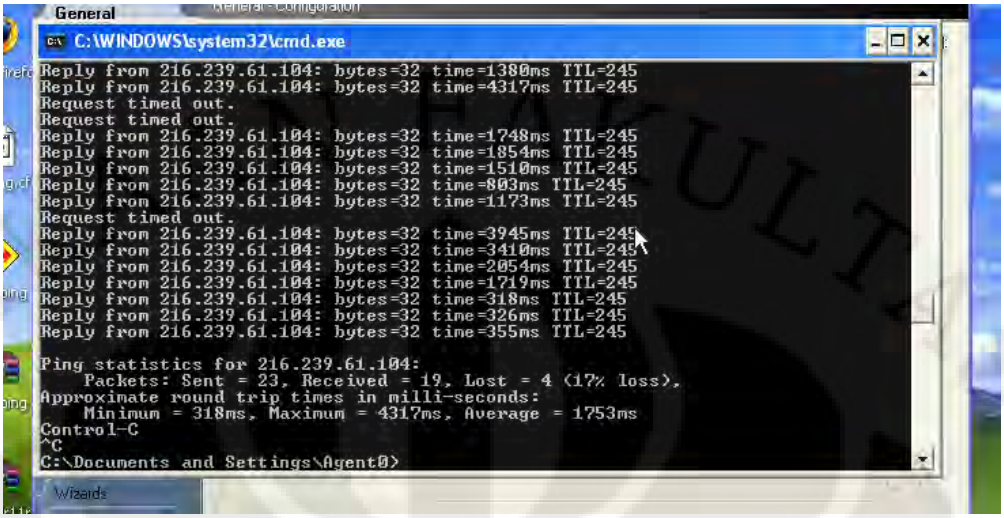
Perhitungan *Termination phase* pada Skenario 1

Pada skenario ini dilakukan konfigurasi mekanisme pertahanan menggunakan mekasnisme *prevention phase* dan *termination phase*. Parameter yang dihitung sama dengan percobaan sebelumnya. Topologi dan *tools* yang digunakan juga masih sama yaitu WinArp*Attack*er 3.50.



Tabel 4.2. Grafik yang menunjukan tingkat *reliability* pada skenario 1 dengan mekanisme pertahanan *termination phase* dan *prevention phase*.

Pada percobaan ini terjadi peningkatan *reliability* yang cukup signifikan. Dari hasil perhitungan pada percobaan ini menghasilkan (46/5) X 10 = 92% yang sebelumnya cuman menghasilkan tingkat *reliability* 56%. Hal ini membuktikan bahwasanya implementasi mekanisme pertahanan DoS harus menggunakan kedua mekanisme tersebut. Tetapi mekanisme ini tentunya mempengaruhi *response time* yang terjadi.



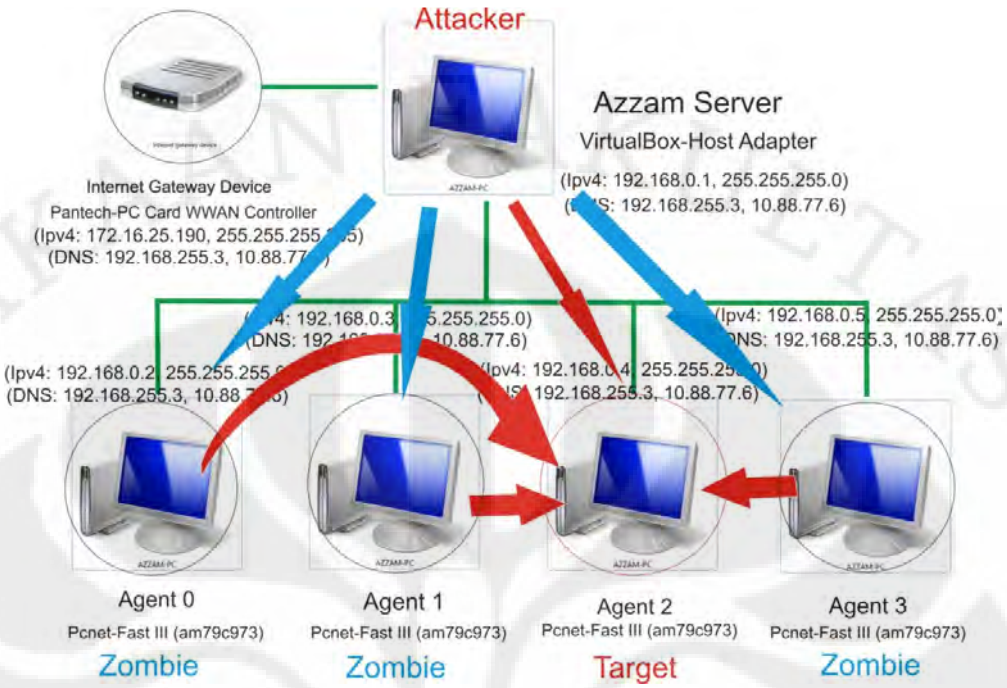
Gambar 4.6. Data *response time* pada skenario 1 menggunakan mekanisme pertahanan *prevention phase* dan *termination phase*.

Dari gambar tersebut tercatat rata-rata *response time* yang diambil dari 23 paket yang dikirim ke *www.google.com* adalah 1753 ms. Data ini naik dari 1029 ms ke 1753 ms atau mengalami peningkatan waktu sebesar 724 ms. Hal ini membuktikan bahwasanya konfigurasi mekanisme *defense* mempengaruhi respon time dari jaringan *host* ke *internet*.

Dengan hasil perhitungan dan analisa pada skenario satu ini dapat disimpulkan sementara bahwa mekanisme pertahanan menggunakan IDS center dan Snort memberikan perlindungan yang cukup signifikan yaitu 92% *reliability*. Tetapi mekanisme ini juga memberikan efek negative yaitu *response time* yang lambat dilihat dari data RTT (*round trip time*) yaitu 1753 ms dibandingkan dengan kondisi normal yang berkisar 100 ms.

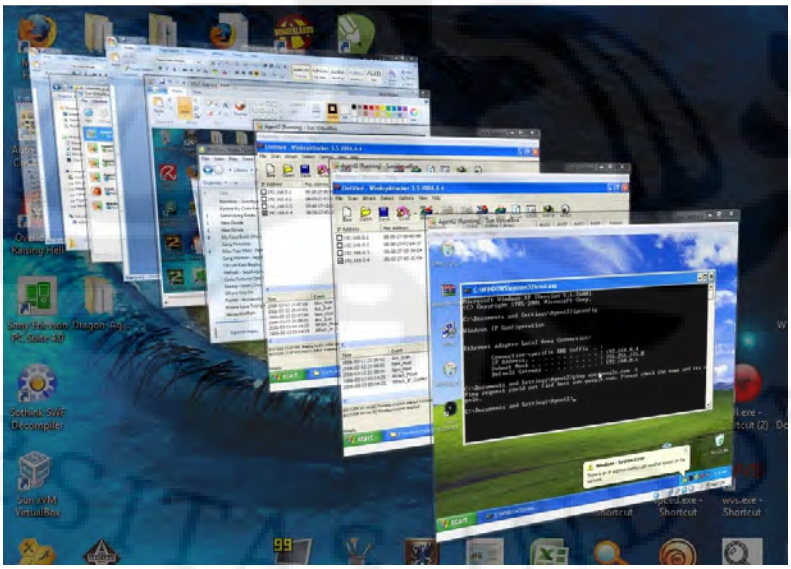
4.3.2. Perhitungan Skenario 2

Skenario ini merupakan simulasi pertahanan pada serangan DDoS (Distributed Denial of Service) yang memanfaatkan beberapa mesin untuk melakukan serangan. Pada topologi jaringan yang telah dibuat pada Bab 3 Target diserang menggunakan beberapa *Virtual machine* dan *Host*. Target disini adalah *Agent*2 kemudian diserang menggunakan 3 *virtual machine* dan 1 *host*.



Gambar 4.7. Topologi jaringan serangan DDoS pada *VirtualBox*

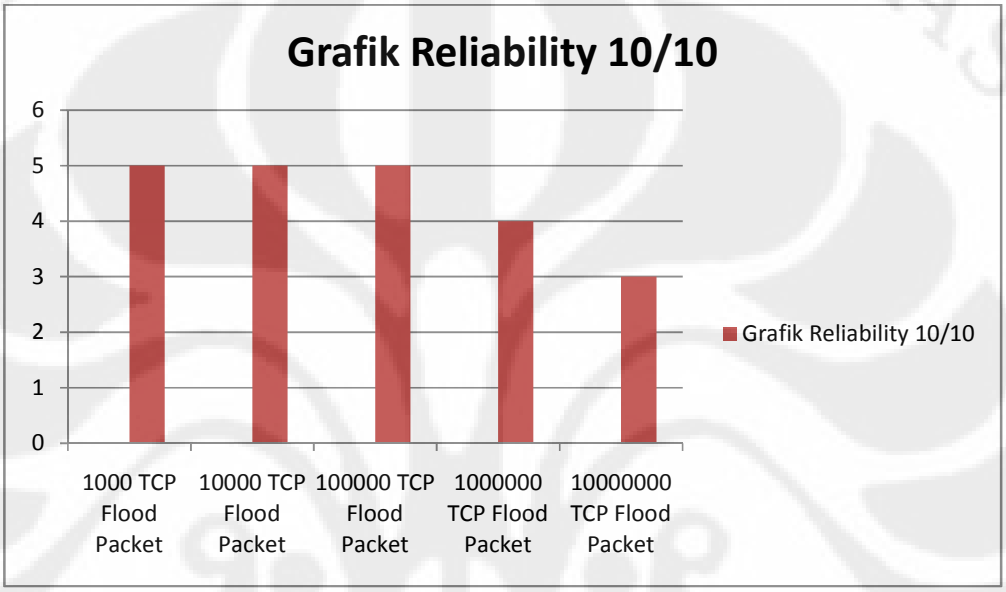
Seperti yang dilakukan pada skenario pertama pada skenario ini pun akan dihitung *response time* dan *reliability* di mesin target. Tentunya serangan ini lebih memakan *resource* karena target di banjiri paket dari beberapa mesin sekaligus. Berikut ini kondisi saat terjadi serangan sebelum mekanisme *defense* diterapkan.



Gambar 4.8. Kondisi *Agent*2 tidak dapat melakukan koneksi ke jaringan

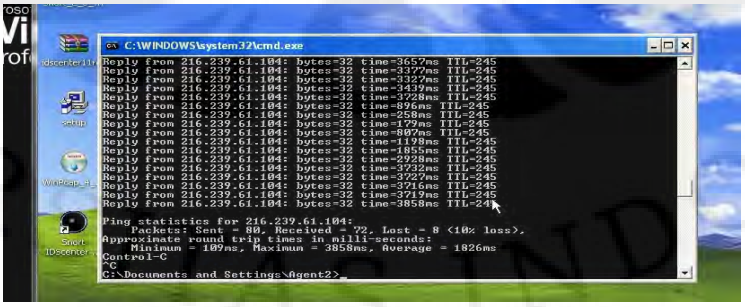
Perhitungan *Prevention phase* pada Skenario 2

Percobaan pertama kali dilakukan pada mekanisme pertahanan *prevention phase*. Seperti percobaan sebelumnya target akan diserang sebanyak 10 kali dengan 5 jenis beban yaitu: 1.000 paket, 10.000 paket, 100.000 paket, 1.000.000 paket dan 10.000.000 paket secara simultan.



Tabel 4.3. Grafik tingkat *reliability* pada percobaan skenario 2

Pada percobaan kali ini terlihat *reliability* pada 3 beban pertama tidak jauh berbeda. Dari sepuluh kali serangan 5 serangan mampu dihindari. Pada serangan terakhir yang memakan banyak *resource* target tidak mampu menahan kecuali dengan memutuskan koneksi. Analisa berikutnya adalah nilai *response time* pada saat serangan mampu dihindari target.

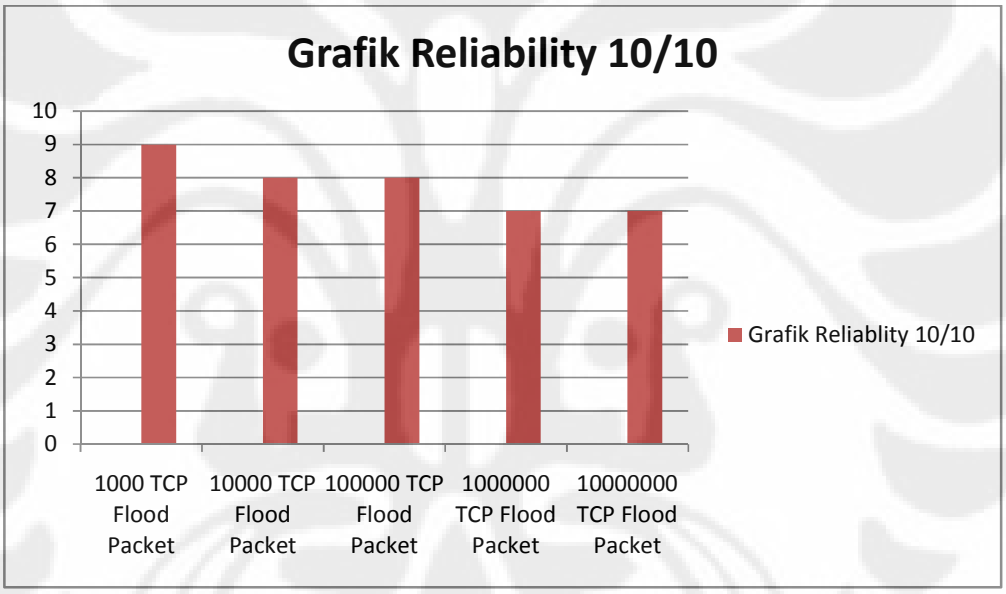


Gambar 4.9. Nilai *response time* berdasarkan RTT (*round trip time*)

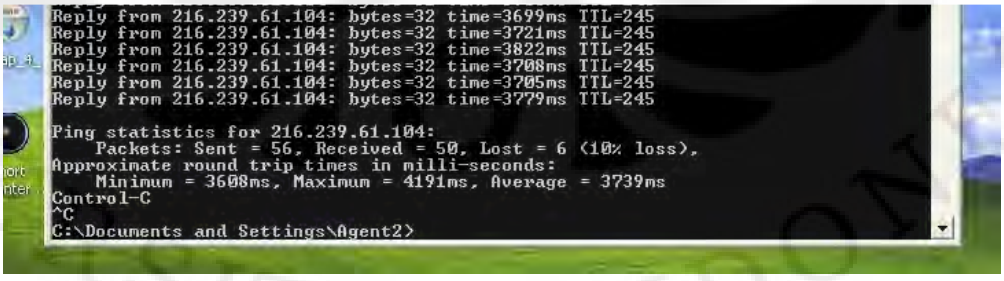
Pada percobaan di skenario 2 ini nilai *response time* yang dicapai adalah 1826 ms. Nilai ini meningkat dibandingkan dengan skenario 1 yang hanya berkisar 1000 ms. Hal ini terjadi karena target mengalami kegagalan dalam menangani *resource* yang sangat terbatas.

Perhitungan *Termination phase* pada Skenario 2

Mekanisme pertahanan *termination phase* dan *prevention phase* di gabung pada skenario ini untuk melihat nilai *reliability* dan *response time* yang tercapai. Seperti yang telah dilakukan pada percobaan sebelumnya *reliability* dinilai berdasarkan 10 kali percobaan serangan dengan berbagai jenis beban.



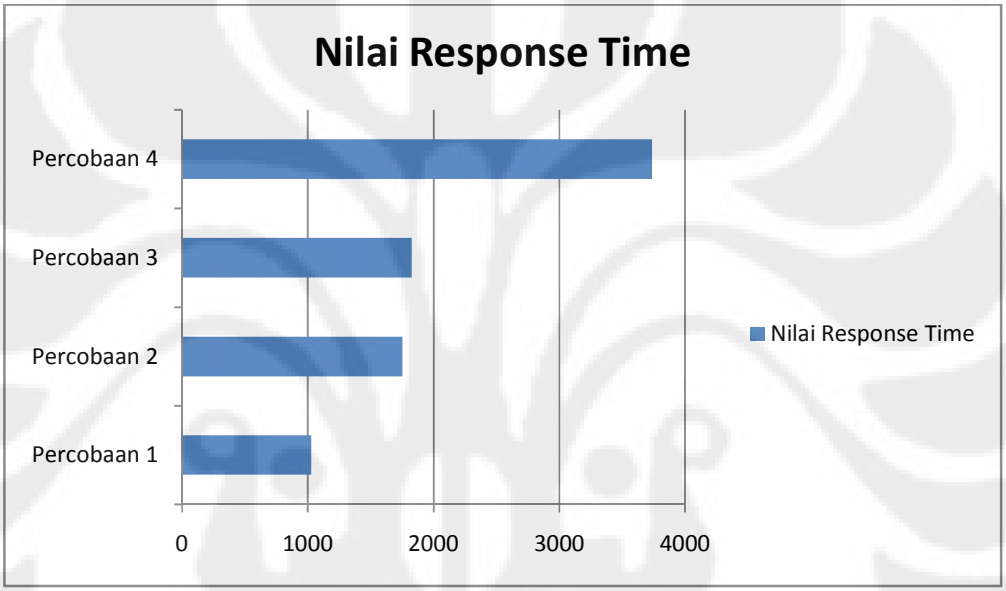
Tabel 4.4. Nilai *reliability* yang dicapai pada skenario 2



Gambar 4.10 Nilai *response time* pada skenario 2 (*termination phase*)

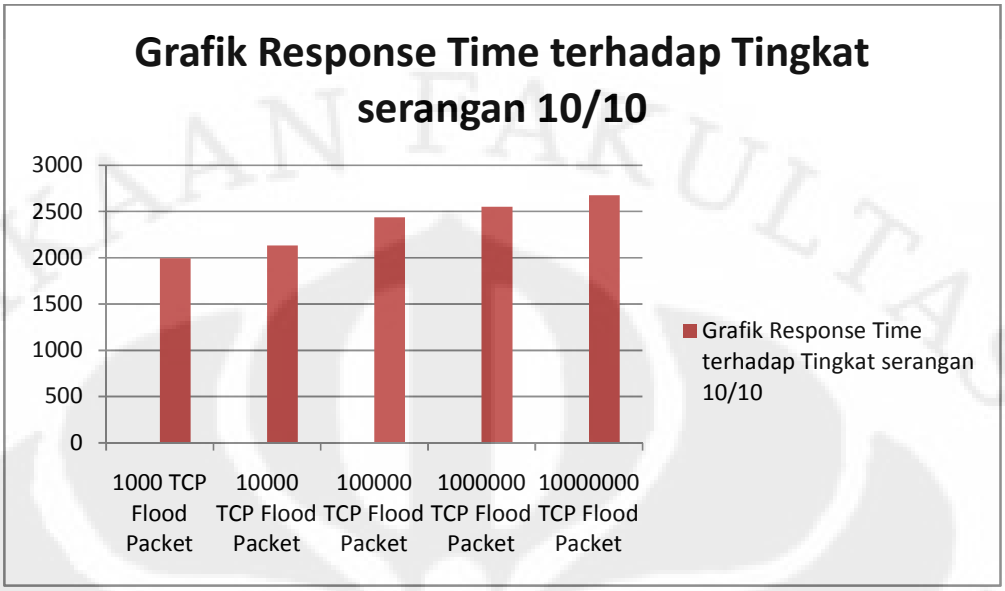
4.4. Analisa dan Pembahasan

Bagian ini akan menjelaskan tentang pembahasan pada hasil percobaan yang telah dilakukan. Dari hasil percobaan diatas menunjukan bahwa terjadi nilai *reliability* yang cukup konstan. Artinya mekanisme pertahanan yang diterapkan pada IDS center dan Snort membawa hasil yang baik. Akan tetapi tidak sejalan dengan *response time*. Nilai ini terus bertambah ketika konfigurasi yang dilakukan semakin *secure* dan *complex*.



Tabel 4.5. Nilai *response time* pada beberapa percobaan (dalam ms)

Seperti yang dapat dilihat pada grafik 4.5. terjadinya peningkatan *response time* berkaitan dengan beban serangan dan beban konfigurasi. Hal ini membuktikan bahwasanya tingkat keamanan berbanding terbalik dengan kenyamanan dan fungsionalitas. Semakin aman suatu *system* maka semakin sulit digunakan dan berkurang fungsi dari *system* tersebut.



Tabel 4.6 Tabel hubungna antara nilai *response time* terhadap tingkat serangan DoS dan DDoS

Pada tabel diatas menggambarkan nilai *response time* terhadap tingkat serangan DoS dan DDoS. Dalam hal ini nilai *response time* mengalami kenaikan yang tidak signifikan. Tandanya sistem pertahanan yang dilakukan oleh mekanisme pertahanan DoS dan DDoS yang diimplementasikan pada IDS Center berhasil.

Pembahasan pada percobaan ini memberikan kesimpulan bahwa penanganan serangan DoS dan DDoS cukup memakan banyak *resource*. Banyak hal yang harus dipertimbangkan juga dalam menerapkan mekanisme pertahanan pada suatu jaringan atau *system*. Karena pada hal ini tingkat keamanan sangat berkaitan dengan kenyamanan dan fungsi dari *system* tersebut.

BAB 5

KESIMPULAN

Pada bab ini akan diterangkan tentang kesimpulan yang dapat diambil dari percobaan dan penelitian ini. Dari hasil percobaan dapat ditarik kesimpulan sebagai berikut.

1. Dari simulasi percobaan yang telah dilakukan menggunakan VirtualBox memberikan hasil data yang cukup baik terhadap pertahanan DoS dan DDoS.
2. Mekanisme pertahanan dalam simulasi percobaan dapat dilakukan dengan men-implementasikan *variable-variable* yang berpengaruh pada serangan seperti *tcp connection.*
3. Dari hasil perhitungan pada scenario 1 (Simulasi Pertahanan DoS) *reliability* yang didapatkan adalah 92% dari 5 jenis serangan yang berbeda. Nilai rata-rata *response time* pada sistem pertahanan ini adalah 1753 ms.
4. Dari hasil perhitungan pada scenario 2 (Simulasi Pertahanan DDoS) *reliability* yang didapatkan adalah 82% dan nilai *response time* rata-rata adalah 2086.75 ms.
5. Tabel 4.6 yang menggambarkan hubungan *response time* dengan tingkat serangan DoS dan DDoS menunjukan peningkatan *delay* yang tidak signifikan.
6. Tingkat keamanan pada suatu sistem tidak berbanding lurus dengan kemudahan penggunaan dan fungsionalitas dari sistem tersebut. Semakin tinggi tingkat kemanan suatu sistem maka akan semakin sulit digunakan dan semakin terbatas fungsinya.

DAFTAR REFERENSI

[1] "____", "*Internet* World Stats" , 2009, diakses dari :

http://www.*internet*worldstats.com/stats.htm

[2] Mirkovic, Jelena, dkk, "*Internet* Denial of Service". Prentice Hall, 2004.

[3] "____", "CSI/FBI Komputer Crime and Security Survey" 2005, didapat dari :

Distributed Denial of Service.ppt, anegroni@cisco.com

[4] Tannenbaum, Andre S, "Komputer *Network*", 4th Edition, Prentice Hall, 2003.

[5] "____". "The Physical Layer", 2009, diakses dari :

http://www.mindspring.com/~cari/*network*s/physlayer.html

[6] "____", " Datalink Layer" 2009, diakses dari:

www.ee.ui.ac.id/endangs/jarkom/Jarkom3b.ppt

[7] "____", "Basic *Routing*", 2009, diakses dari:

http://www.cisco.com/en/US/docs/*internetwork*ing/technology/handbook/*Routing*-Basics.html

[8] Cole, Eric, dkk, "*Network* Security Bible", Wiley Publishing, 2005.

[9] Manzuik, Steve, dkk, "*Network* Security Assessment", Synggres, 2007

[10] "____", "Telnet Vs SSH", 2009, diakses dari :

http://ccnablog.globalknowledge.com/2009/06/10/telnet-vs-ssh/

[11] Endorf, Carl, dkk, "Intrusion Detection & Prevention", Osborne, 2004.

[12] Noonan, Wes, "*Firewall* Fundamentals", Cisco Press, 2006.

[13] "____", "DNS Sec" 2009, diakses pada :

http://www.dnssec.net/

[14] Lukatsky, Alex, "Protect Your Information with Intrusion Detection", A-List Publishing, 2003

[15] "____"," *Virtual machine*", 2009, diakses pada :

www.*VirtualBox*.org

[16] "____", " IDS Center", 2009, diakses pada :

www.engagesecurity.com/products/idscenter