UNIVERSITAS INDONESIA

IMPLEMENTASI IDS (INTRUSION DETECTION SYSTEM) SERTA MONITORING JARINGAN DENGAN INTERFACE WEB BERBASIS BASE PADA KEAMANAN JARINGAN

SKRIPSI

MONIKA KUSUMAWATI

0606078430

DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
PROGRAM TEKNIK KOMPUTER
DEPOK
Juni 2010

UNIVERSITAS INDONESIA

IMPLEMENTASI IDS (INTRUSION DETECTION SYSTEM) SERTA MONITORING JARINGAN DENGAN INTERFACE WEB BERBASIS BASE PADA KEAMANAN JARINGAN

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

MONIKA KUSUMAWATI

0606078430

DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
PROGRAM TEKNIK KOMPUTER
DEPOK

Juni 2010

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,

dan semua sumber yang dikutip maupun dirujuk

telah saya nyatakan dengan benar.

Nama	: Monika Kusumawati
NPM	: 0606078430
Tanda tangan	:
Tanggal	: 1 Julni 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Monika Kusumawati

NPM : 0606078430

Program Studi : Teknik Komputer

Judul Skripsi : Implementasi IDS (Intrusion Detection System) serta monitoring jaringan dengan interface web berbasis BASE pada keamanan jaringan.

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

Pembimbing **: Muhammad Salman, ST, MIT** (_____)

Penguji **: Ir. Endang Sriningsih MT,Si** (_____)

Penguji **: Prima Dewi Purnamasari ST, M.T, M.Sc** (_____)

Ditetapkan di : Depok

Tanggal : 1 Juli 2010

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelap Sarjana Teknik Jurusan Teknik Komputer pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi 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 skripsi ini;
(2) Mercator office beserta Sheli dan Dita yang telah mengizinkan saya untuk dapat meminjam ruangan beserta fasilitas yang ada pada ruangan Mercator Office UI, Enginering Center;
(3) Pak Cherry yang sudah banyak meluangkan waktunya untuk membantu dan mendengarkan keluhan saya;
(4) Ibu saya yang telah mendukung saya selama ini dan selalu mendoakan saya serta almarhum ayah saya yang selalu mengingatkan saya untuk selalu percaya pada Tuhan;
(5) Kakak – kakak saya yang telah memberikan bantuan dukungan material maupun moral;
(6) Kak Cholid yang sudah memberikan saya semangat dan harapan;
(7) Adit yang telah memberikan masukan-masukannya;
(8) Yudha sebagai rekan kerja dan rekan seperjuangan;
(9) Winda, Yomma, Barnas, dan Cesilia yang telah memberikan semangat dan juga dorongan dalam dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini nantinya akan membawa manfaat bagi pengembang ilmu.

Depok, Juni 2010

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

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

Nama	: Monika Kusumawati
NPM	: 0606078430
Program Studi	: Teknik Komputer
Departemen	: Teknik Elektro
Fakultas	: Teknik
Jenis Karya	: Skripsi

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 :

IMPLEMENTASI IDS (INTRUSION DETECTION SYSTEM) SERTA MONITORING JARINGAN DENGAN INTERFACE WEB BERBASIS BASE PADA KEAMANAN JARINGAN

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Juli 2010

Yang menyatakan

(……)

ABSTRAK

Nama : Monika Kusumawati

Program Studi : Teknik Kompuer

Judul : Implementasi IDS (Intrusion Detection System) serta monitoring jaringan dengan interface web berbasis BASE pada keamanan jaringan.

Teknologi informasi (TI) telah berkembang dengan pesat, terutama dengan adanya jaringan internet yang dapat memudahkan untuk melakukan komunikasi dengan pihak yang lain. Namun dengan mudahnya pengaksesan terhadap informasi tersebut menyebabkan timbulnya masalah baru yaitu informasi atau data-data penting dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab untuk mendapatkan keuntungan sendiri. Sehingga suatu sistem keamanan pada jaringan menjadi salah satu aspek yang penting untuk diperhatikan dari sebuah sistem informasi. Oleh karena itu untuk mendapatkan sebuah keamanan jaringan maka diperlukan suatu *tools* yang dapat mendeteksi adanya serangan di dalam jaringan. Banyaknya *tools* ini, maka dapat dibandingkan antara sistem yang hanya dapat mendeteksi dengan sistem yang dapat melakukan tindakan juga. Sistem yang hanya mendeteksi ini akan diimplementasikan dengan menggunakan aplikasi IDS yaitu ***Snort***. Sistem IDS ini yaitu sistem yang mampu memberikan *alerting* maupun log apabila terjadi serangan di dalam jaringan, selain itu IDS ini juga mampu memonitoring serangan melalui interface web.

Sistem IDS ini menggunakan *Operating System* Windows 7. Sistem ini dibagi menjadi beberapa modul yaitu IDS *software* yaitu snort, report modul yaitu BASE, dan juga kiwi syslog yang mampu mengirimkan *alerting,* untuk *network device* yang digunakan adalah sebuah hub.

Pengujian sistem dilakukan dengan menggunakan beberapa Janis serangan yaitu *IP Scan, Port Scan,* dan *Flooding*. Skenario dalam pengujian ini berdasarkan *functionality test* dan *response time*. Pengujian *Functionality test* ini akan membandingkan nilai dari serangan terhadap 1 *client,* 2 *client,* dan 3 *client* begitu juga dengan *response time*. Berdasarkan percobaan yang telah dilakukan terjadi kenaikan alert sebesar 23,12 % dari 1 client ke 2 client, 13,54 % dari 2 client ke 3 client, serta 39,79 % dari 1 client ke 3 client selain itu terjadi kenaikan response time sebesar 20.31 % dari 1 client ke 2 client, 12,29 % dari 2 client ke 3 client, serta 35,10 % dari 1 client ke 3 client.

Kata Kunci : IDS, Snort, BASE, IP Scan, Port Scan, Flooding

ABSTRACT

Name : Monika Kusumawati

Major : Computer Engineering

Title : Implementation of IDS (Intrusion Detection System) and Network Monitoring with BASE-based web interface on network security.

Information technology has been growing rapidly, especially with the existence of internetwork which make it easier to communicate with others. However the advantages of internetwork should be paid with the cyber crime, those who unauthentically access and take data or information for certain purposes. This fact shows us that nework security is a critical matter that needs special attention when we build an information system. Nowadays, there are several tools that can be used to detect an attempt of network intrusion, some of the tools are only able to detect the intrusion usualy called as Intrusion Detection System (IDS), others are capable of both detection and prevention usualy called as Intrusion Prevention System (IPS). Snort is one of IDS tools that commonly implemented in network security system, this IDS generate an alert and log if there is an attempt of network intrusion. Snort allows the network administrator to monitor the network through web interface

IDS System that implemented in this project runs on operating system Windows 7. The system is devided into several modules those are IDS software (Snort), module report (BASE), and alert generator (Kiwi Syslog). This network uses hub as network device.

The network system then tested by using several types of attack such as IP Scan, Port Scan, and Flooding. Testing scenario is based on functionality test and response time. Functionality test and response time assessments are to compare the value of attack on 1 client, 2 client, and 3 client. Based on experiment that have been done there is an increase of 23.12% alert from 1 client to two client, 13.54% from 2 client to 3 client, and 39,79% from 1 client to 3 client other than that there is an increase of response time at 20.31% of 1 client to 2 client, 12.29% from 2 client to 3 client, and 35.10% from 1 client to 3 client.

Keywords : IDS, Snort, BASE, IP Scan, Port Scan, Flooding

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN

ARP : Address Resolution Protocol

BASE : Basic Analysis Security Engine

DOS : Denial of Service

DDOS : Distribute Denial of Service

GUI : Graphical User Interface

HIDS : Host-based Intrusion Detection System

ICMP : Internet Control Message Protocol

IDS : Intrusion Detection System

IPS : Intrusion Prevention System

LAN : Local Area network

MAN: Metropolitan Area Network

Mysql : My Structured Query Language

NIDS : Network-based Intrusion Detection System

PHP : Personal Home Page

SNMP : Simple Network Management Protocol

TCP/IP : Transmission Control Protocol / Internet Protocol

TI : Teknologi Informasi

UDP : User Datagram Protocol

WAN : Wide Area Network

BAB I
PENDAHULUAN

1.1 LATAR BELAKANG

Pada era global saat ini, teknologi informasi (TI) telah berkembang dengan pesat, terutama dengan adanya jaringan internet yang dapat memudahkan dalam melakukan komunikasi dengan pihak yang lain. Selain itu, para pengguna atau *user* dapat mengakses hampir seluruh informasi yang dibutuhkan baik itu informasi yang bersifat publik maupun bersifat pribadi. Namun dengan mudahnya pengaksesan terhadap informasi tersebut menyebabkan timbulnya masalah baru yaitu informasi atau data-data penting dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab untuk mendapatkan keuntungan sendiri. Sehingga suatu sistem keamanan pada jaringan menjadi salah satu aspek yang penting untuk diperhatikan dari sebuah sistem informasi. Biasanya sistem keamanan tergantung dari ketersediaan dan kecepatan administrator dalam menangani ganguan yang akan terjadi pada jaringan tersebut. Apabila jaringan mengalami gangguan yang menyebabkan jaringan tidak berfungsi maka administrator juga tidak dapat lagi mengakses sistem bahkan administrator tidak dapat memperbaiki atau memulihkan sistem dengan cepat. Oleh karena itu, dibutuhkan suatu sistem dalam menangani gangguan atau ancaman yang akan terjadi secara optimal dalam waktu yang cepat dan otomatis yang hasil keluaran dari serangan tersebut dapat pula ditampilkan dalam bentuk grafik. Bahkan keamanana jaringan yang terus mendapatkan perhatian dari para pemakai jaringan yang membuat semakin banyaknya *tools* yang digunakan untuk mendeteksi bahkan dapat mengambil tindakan apabila terjadi serangan yang masuk ke dalam jaringan. Banyaknya *tools* yang tersedia saat ini dapat dimanfaatkan untuk mendapatkan *tools* yang lebih dipercaya untuk melindungi jaringan. Banyaknya *tools* ini dapat membandingkan antara sistem yang hanya dapat mendeteksi dengan sistem yang dapat melakukan tindakan. Sistem yang hanya mendeteksi ini akan diimplementasikan dengan menggunakan aplikasi *Intrusion Detection System* (IDS) yaitu ***Snort***.

Implementasi ini dilakukan dengan menggunakan beberapa *tools* seperti *IP Scan, Port Scan* dan *Flooding* sebagai penyerang, sedangkan untuk mendeteksi apabila terjadi serangan, maka dilakukan implementasi dengan menggunakan *Kiwi Syslog*. Kiwi Syslog Server merupakan freeware Syslog Server untuk Windows. Kiwi Syslog Server ini berguna sebagai *logs, displays, alert,* dan melakukan tindakan lainnya di pesan syslog dan perangkap SNMP yang dapat diterima dari host seperti firewall, router, switch, Unix host dan syslog lainnya yang diaktifkan.

Hasil keluaran serangan yang dilakukan dapat ditampilkan oleh Kiwi Syslog yang berupa alert. Hasil keluaran atau *log file* dari sistem yang diserang ini nantinya akan ditampilkan dalam bentuk interface yang berbasiskan pada web yaitu dengan menggunakan ***BASE*** (*Basic Analysis and Security Engine*).

1.2 TUJUAN PENELITIAN

Tujuan dari penulisan skripsi ini adalah untuk merancang bangun sebuah aplikasi *Intrusion Detection System (IDS)* yang dapat mendeteksi adanya serangan di dalam jaringan dengan cepat serta menganalisa alert dan log yang dihasilkan melalui monitor dengan menggunakan *kiwi syslog* dan BASE (*Basic analysis and security engine*) sebagai peringatan kepada Administrator serta menghitung nilai terhadap jumlah alert dan waktu yang dibutuhkan oleh sistem untuk mendeteksi adanya serangan di dalam jaringan.

1.3 BATASAN MASALAH

Untuk menghindari meluasnya materi pembahasan skripsi ini, maka penulis membuat batasan masalah sebagai berikut :

1. Merancang sebuah aplikasi IDS dengan menggunakan aplikasi *snort* yang memiliki fungsi untuk melakukan deteksi terhadap serangan dan memberikan *alerting*. *Report module* yang dipakai untuk mengelola data-

data *alerting* yang dihasilkan oleh snort dan menampilkannya dalam bentuk tampilan web yaitu dengan menggunakan BASE (*Basic Analysis Security Engine*).

2. Apabila terjadi serangan maka snort akan mendeteksi serangan tersebut dan memberikan *alerting* yang kemudian akan dikirim ke Kiwi Syslog untuk diklasifikasikan berdasarkan prioritas serangan. Setiap serangan juga disimpan yang kemudian akan ditampilkan dalam interface web berbasis BASE.
3. Sistem operasi yang digunakan oleh sistem IDS yaitu Window 7 sedangakan untuk BASE membutuhkan database MySQL.
4. Sistem Operasi pada *Client* menggunakan Window Vista dan 7.

1.4 METODE PENELITIAN

Metode penelitian yang digunakan pada Skripsi ini adalah :

1. **Studi literatur dan pustaka,** yaitu melakukan berbagai diskusi pembahasan baik dengan dosen pembimbing maupun dengan orang yang berkompeten pada kasus ini serta dari pustaka yang mendukung.
2. **Pendefinisian masalah dan kebutuhan sistem.**
3. **Analisa dan perancangan sistem,** yang meliputi tahapan terstruktur sebagai berikut :
   a. Perancangan sistem *Intrusion Detection System* (IDS).
   b. Perancangan interface web berbasis BASE untuk menampilkan hasil dari setiap serangan.
   c. Implementasi dan Uji Coba
4. **Implementasi perancangan perangkat lunak,** sistem yang akan diimplementasikan adalah sistem *Intrusion Detection System* (IDS), yaitu sistem yang dapat mendeteksi adanya serangan yang masuk ke dalam jaringan.
5. **Uji Coba dan Evaluasi Sistem,** melakukan ujicoba dan mengevaluasikan sistem yang telah diimplementasikan.

6. **Mengambil kesimpulan,** pengujian *Intrusion Detection System* (IDS) yang dapat disimpulkan dari hasil log yang ada.

1.5 SISTEMATIKA PENULISAN

Sistematika penulisan Skripsi ini dibagi menjadi beberapa bab yang meliputi :

BAB I Pendahuluan

Bab ini berisi tentang latar belakang, tujuan penulisan, batasan masalah, metologi penelitian yang dipakai dalam penelitian, dan sistematika penulisan yang memuat susunan penulisan Skripsi ini.

BAB II Landasan Teori

Bab ini membahas definisi-definisi dan konsep-konsep dasar yang digunakan dalam penelitian ini, meliputi teori *IDS, Snort, BASE,* dan hal lain yang dianggap perlu sebagai rujukan masalah.

BAB III Perancangan Sistem

Bab ini berisi pembahasan tentang desain dari sistem IDS dan juga perancangan dari sistem interface web berbasis BASE.

BAB IV Pengujian dan Analisa

Bab ini berisi tentang analisa dan pengujian dari sistem IDS tersebut yang hasil dari alerting dan log-nya dapat ditampilkan di dalam Kiwi Syslog dan BASE.

BAB V Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan yang di dapat dari hasil penelitian yang dilakukan serta saran untuk pengembangan lebih lanjut.

BAB II

PENGENALAN SISTEM IDS DAN WEB BASE

Pada bab ini akan membahas tentang teori dasar yang melandasi permasalahan dan penyelesaian yang diangkat dalam skripsi ini. Dasar teori yang diberikan meliputi keamanan jaringan, konsep dari *IDS, SNORT,* dan *BASE.*

2.1 KEAMANAN JARINGAN

Keamanan jaringan dapat digambarkan secara umum yaitu apabila komputer yang terhubung dengan jaringan lebih banyak mempunyai ancaman keamanan dari pada komputer yang tidak terhubung ke mana-mana. Namun dengan adanya pengendalian maka resiko yang tidak diinginkan dapat dikurangi. Adanya keamanan jaringan maka para pemakai berharap bahwa pesan yang dikirim dapat sampai dengan baik ke tempat yang dituju tanpa mengalami adanya kecacatan yang diterima oleh si penerima, misalnya saja adanya pemodifikasian pesan. Biasanya jaringan yang aksesnya semakin mudah, maka keamanan jaringannya semakin rawan, namun apabila keamanan jaringan semakin baik maka pengaksesan jaringan juga semakin tidak nyaman.

Di dalam keamanan jaringan terdapat pula resiko jaringan komputer yang merupakan segala bentuk ancaman baik fisik maupun logic yang langsung atau tidak langsung mengganggu kegiatan yang sedang berlangsung dalam jaringan. Resiko dalam jaringan komputer disebabkan oleh beberapa faktor yaitu : [6]

- Kelemahan manusia
- Kelemahan perangkat keras komputer
- Kelemahan sistem operasi jaringan
- Kelemahan sistem jaringan komunikasi

Selain itu, kemanan jaringan juga mempunyai tujuan yang dapat membuat keamanan jaringan lebih ditingkatkan lagi, yaitu :

- **Confidentiality** : Adanya data-data yang penting yang biasanya tidak boleh di akses oleh seseorang, maka dilakukan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. Biasanya *confidentiality* ini berhubungan dengan informasi yang diberikan ke pihak lain.
- **Integrity** : Bahwa pesan yang disampaikan tetap orisinil yang tidak diragukan keasliannya, tidak dimodifikasi selama dalam perjalanan dari sumber ke penerimannya.
- **Availability** : Dimana *user* yang mempunyai hak akses diberi akses tepat pada waktunya, biasanya ini berhubungan dengan ketersediaan informasi atau data ketika dibutuhkan. Apabila sistem informasi ini diserang maka dapat menghambat bahkan menyebabkan tidak dapat mengakses informasi tersebut.

Tujuan keamanan jaringan dapat dicapai dengan suatu metode keamanan jaringan yang dapat melindungi sistem baik dari dalam maupun dari luar jaringan, namun bukan hanya melindungi tetapi harus dapat bertindak apabila terjadi serangan yang ada di dalam jaringan. Salah satu metode tersebut yaitu *Intrusion Detection System (IDS)* dan *Intrusion Prevention System (IPS)*. Namun, selain metode tersebut dibutuhkan juga suatu pemahaman tentang menentukan kebijakan keamanan (*security policy*) dalam keamanan jaringan. Jika ingin menentukan apa saja yang harus dilindungi maka harus mempunyai perencanaan keamanan yang matang dan baik berdasarkan pada prosedur dan kebijakan keamanan jaringan, karena apabila tidak direncanakan maka tidak akan sesuai dengan yang diharapkan dalam perlindungan jaringan.

2.2 KEBIJAKAN KEAMANAN

Salah satu problem *network security* yang paling penting adalah menentukan kebijakan dalam *network security*. Kebanyakan orang menginginkan solusi teknis untuk setiap masalah yaitu dapat berupa program yang dapat memperbaiki masalah-masalah *network security*. Padahal, perencanaan keamanan yang matang berdasarkan prosedur dan kebijakan dalam network security akan membantu menentukan apa-apa yang harus dilindungi, berapa besar biaya yang

harus ditanamkan dalam melindunginya, dan siapa yang bertanggung jawab untuk menjalankan langkah-langkah yang diperlukan untuk melindungi bagian tersebut. Di dalam keamanan jaringan, peran manusia memegang tanggung jawab keamanan yang cukup berperan. Keamanan jaringan tidak akan efektif kecuali orang-orangnya mengetahui tanggung jawabnya masing-masing. Dalam menentukan *network security policy*, diperlukan adanya ketegasan apa yang diharapkan, serta dari siapa hal tersebut diharapkan. Selain itu, kebijakan ini harus mencakup : [4]

1) Tanggung jawab keamanan *network user*, meliputi antara lain keharusan *user* untuk mengganti passwordnya dalam periode tertentu, dengan aturan tertentu, atau memeriksa kemungkinan terjadinya pengaksesan oleh orang lain.
2) Penggunaan yang benar sumber-sumber network, dengan menentukan siapa yang dapat menggunakan sumber-sumber tersebut, apa yang dapat dan tidak boleh dilakukan.
3) Langkah-langkah yang harus diperbuat bila terdeteksi masalah keamanan, siapa yang harus diberitahu. Hal ini harus dijelaskan dengan lengkap, bahkan hal-hal yang sederhana seperti menyuruh *user* untuk tidak mencoba melakukan apa-apa atau mengatasi sendiri bila masalah terjadi, dan segera memberitahu sistem administrator.

Adanya kebijakan tersebut maka manusia merupakan salah satu faktor yang sangat penting, namun sering dilupakan dalam pengembangan teknologi informasi, begitu juga dengan pengembangan di bidang keamanan jaringan. Salah satu contohnya adalah dalam penggunaan *password* yang sulit justru menyebabkan pengguna menuliskannya pada kertas yang ditempelkan pada komputer atau meja. Kebijakan keamanan dapat dijaga atau disusun karena faktor manusia dan budaya setempat juga harus diperhitungkan dan dipertimbangkan. Selain faktor dari manusia dan budaya itu sendiri, faktor yang dibutuhkan juga tergantung dari organisasi, keputusan yang di ambil merupakan keputusan tentang keamanan komputer, dan juga masalah biaya dari suatu sistem keamanan.

2.2.1 Mengenali ancaman terhadap network security

Langkah awal dalam mengembangkan rencana *network security* yang efektif adalah dengan mengenali ancaman yang mungkin datang yaitu : [6]

- Akses tidak sah, oleh orang yang tidak mempunyai wewenang.
- Kesalahan informasi, segala masalah yang dapat menyebabkan diberikannya informasi yang penting atau sensitif kepada orang yang salah, yang seharusnya tidak boleh mendapatkan informasi tersebut.
- Penolakan terhadap *service*, segala masalah mengenai *security* yang menyebabkan sistem mengganggu pekerjaan-pekerjaan yang produktif.

2.3 INTRUSION DETECTION SYSTEM

Intrusion Detection System (IDS) adalah sebuah aplikasi perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan, jadi IDS atau Sistem Deteksi Penyusupan merupakan sebuah sistem komputer yang dapat dikombinasikan antara *hardware* dan *software* yang dapat melakukan deteksi penyusupan pada sebuah jaringan. IDS pada dasarnya adalah suatu sistem yang memiliki kemampuan untuk menganalisa data secara *realtime* dalam mendeteksi, mencatat (log) dan menghentikan penyalahgunaan dan penyerangan. IDS merupakan *security tools* yang dapat digunakan untuk menghadapi aktivitas *hackers.* IDS mempunyai beberapa komponen yaitu :

1. **Sensor** yang dapat mengenali adanya *security events.*
2. **Console** yang dapat memonitor event dan alerts dan mengontrol sensor.
3. **Central Engine** yang berguna untuk menyimpan *events logged* yang dilakukan oleh sensor kedalam database dan menggunakan aturan-aturan keamanan yang berguna untuk menangani *event* yang terjadi.

Apabila ada aktivitas yang dianggap mencurigakan di dalam jaringan maka IDS ini akan memberitahukan atau mendeteksi terhadap serangan tersebut. Namun IDS ini tidak dapat melakukan tindakan atau pencegahan jika terjadi serangan atau penyusupan di dalam jaringan tersebut. IDS mempunyai peran yang

cukup membantu dalam hal-hal yang berkaitan dengan keamanan jaringan, diantaranya : [6]

- Secara aktif mengamati segala macam kegiatan yang mencurigakan.
- Memeriksa *audit logs* dengan sangat cermat dan seksama.
- Mengirimkan alert kepada administrator saat adanya serangan-serangan khusus di deteksi
- Memberi tanda segala macam kerentanan yang ditemukan.

Namun kembali lagi pada kemampuan dari IDS yang hanya mampu untuk menghentikan serangan yang sedang berlangsung dan memang mempunyai kemampuan yang terbatas yang tergantung pada bagaimana melakukan konfigurasi IDS yang baik. IDS yang bermanfaat untuk mengatasi pencegahan terhadap suatu serangan atau penyusupan memang diperlukan suatu pemeliharaan yang mencukupi suatu sistem keamanan secara keseluruhan.

Dilihat dari kemampuan mendeteksi serangan atau penyusupan di dalam jaringan, maka IDS dibagi menjadi 2 yaitu :

1. *Network-based Intrusion Detection System* (NIDS) yaitu NIDS ini akan menganalisa semua lalu lintas yang melewati ke sebuah jaringan yang akan mencari apakah ada percobaan serangan atau penyusupan ke dalam sistem jaringan. Biasanya NIDS berada di dalam segmen jaringan penting di di mana server berada atau berada di pintu masuk jaringan. Walaupun demikian NIDS mempunyai kelemahan yaitu bahwa NIDS agak rumit untuk diimplementasikan dalam sebuah jaringan yang menggunakan *switch Ethernet*, meskipun beberapa *vendor switch Ethernet* sekarang telah menerapkan fungsi IDS di dalam *switch* buatannya untuk memonitor *port* atau koneksi.

2. *Host-based Intrusion Detection System* (HIDS) yaitu sistem yang mampu mendeteksi hanya pada *host* tempat implementasi IDS. Aktivitas sebuah *host* jaringan individual akan dipantau apakah terjadi sebuah percobaan serangan atau penyusupan ke dalamnya atau tidak. HIDS seringnya diletakkan pada

server-server kritis di jaringan, seperti halnya *firewall, web server*, atau *server* yang terkoneksi ke Internet.

Mengingat bahwa IDS ini hanya mampu mendeteksi adanya serangan yang masuk pada jaringan maka memang kebanyakan produk IDS merupakan sistem yang bersifat pasif. Peringatan yang terjadi akibat adanya serangan di dalam jaringan akan memberitahukan admin bahwa ada seranngan ataupun gangguan terhadap jaringan. Berkembangnya dunia IT ini juga ikut memacu berkembangnya IDS yaitu IDS yang bersifat aktif yang memang merupakan dari hasil pengembangan IDS, IDS yang aktif ini dapat melakukan beberapa tugas yang mampu melindungi *host* atau jaringan dari serangan ketika terdeteksi, seperti halnya menutup beberapa *port* atau memblokir beberapa alamat IP. Produk seperti ini umumnya disebut sebagai *Intrusion Prevention System* (IPS). Beberapa produk IDS juga menggabungkan kemampuan yang dimiliki oleh HIDS dan NIDS, yang kemudian disebut sebagai sistem hibrid (*Hybrid Intrusion Detection System*).

2.4 NTRUSION DETECTION SYSTEM (IDS) MENGENALI ADANYA INTRUDER

Intrusion atau penyusupan dapat didefinisikan sebagai sebuah kegiatan yang bersifat *anomaly, incorrect* atau *inappropriate* yang terjadi di jaringan atau di host. Pada IDS, pengenalan terhadap *intruder* dibagi menjadi dua bagian : [1]

1. *Knowledgebased* atau *misuse detection* yaitu mengenali adanya penyusupan atau serangan dengan cara menyadap paket data kemudian mambandingkannya dengan database rule yang berisi *signature-signature* serangan, apabila paket data mempunyai pola yang sama atau setidaknya salah satu pola terdapat di database rule, maka di anggap adanya serangan.

2. *Behavior based* atau *anomaly based* yaitu mengenali adanya penyusup dengan mengamati adanya kejanggalan-kejanggalan pada sistem, atau adanya penyimpangan–penyimpangan dari kondisi normal, sebagai contoh ada penggunaan memori yang melonjak secara terus menerus atau koneksi

parallel dari 1 (satu) port IP dalam jumlah yang banyak dan dalam waktu yang bersamaan.

Biasanya rule dan signature hanya berisi pola serangan yang selalu di *update* secara rutin, hal ini dikarenakan adanya serangan baru setiap hari. Namun untuk proses deteksi *anomaly* tidak menggunakan rule dan signature, hanya mengamati kondisi normal dari sistem jaringan, jika suatu waktu kondisi dari jaringan tidak normal, hal seperti ini dianggap sebagai suatu serangan. Keunggulan dari sistem deteksi ini dapat mengenali serangan baru yang polanya tidak ada pada rule dan signature hasil dari pembelajaran sistem deteksi itu sendiri.

Kekurangan dari sistem deteksi *anomaly* ini adalah banyaknya *alert false positive* yang yang dikirim ke *user*. Contoh jika suatu waktu server menerima banyak *request* dari *true client internal* dan kinerja sistem meningkat dengan cepat (memory, prosesor), maka sistem deteksi akan melaporkan sebagai serangan. IDS yang dibangun menggunakan dua sistem deteksi ini untuk mengatasi masalah serangan yang terjadi, jika suatu pola serangan tidak ada pada rule dan signature, maka sistem deteksi *anomaly* berfungsi untuk mencari pola serangan baru.

2.5 MELINDUNGI IDS

Salah satu hal utama adalah bagaimana melindungi sistem, di mana perangkat lunak deteksi intrusi sedang berjalan. Jika keamanan IDS terganggu, maka akan mulai mendapatkan alarm palsu atau tidak ada alarm sama sekali. Penyusup dapat menonaktifkan IDS sebelum benar-benar melakukan serangan apapun. Ada berbagai cara untuk melindungi sistem, mulai dari rekomendasi yang sangat umum dan juga canggih. Beberapa di antaranya disebutkan di bawah ini.

- Hal pertama yang dapat dilakukan adalah untuk tidak menjalankan layanan pada sensor IDS sendiri.
- Platform yang menjalankan IDS harus ditambah dengan rilis terbaru dari vendor. Misalnya, jika Snort adalah Microsoft yang berjalan pada mesin Windows, maka harus memiliki semua *patch* keamanan terbaru dari Microsoft yang diinstal.

- Konfigurasi mesin IDS sehingga tidak merespon untuk melakukan ping paket.
- Jika menjalankan Snort pada mesin Linux, gunakan Netfilter / iptable untuk memblokir setiap data yang tidak diinginkan. Snort akan tetap dapat melihat semua data.

Penggunaan IDS hanya untuk tujuan deteksi intrusi. Tidak boleh digunakan untuk kegiatan lain dan *account user* tidak boleh dibuat kecuali yang mutlak diperlukan.

2.6 SNORT

Snort merupakan bagian dari IDS dan merupakan sebuah perangkat lunak *open source*. Snort mampu melakukan analisa *realtime traffic* dan *packet logger* pada jaringan IP dan dapat menganalisa protocol dan melakukan pendeteksian variasi penyerangan. Snort juga memiliki kemampuan *realtime alert*, dimana mekanisme pemasukan alert dapat berupa *user syslog, file, unix socket* ataupun melalui database.

Dalam mengoperasikan snort mempunyai tiga buah mode, yaitu :

1. *Sniffer Mode*

Sniffer Mode ini berfungsi untuk melihat paket yang lewat di jaringan., maka untuk menjalankan snort pada *sniffer mode* tidak terlalu susah. Berikut ini adalah beberapa contoh perintahnya :

```
#snort –v
#snort –vd
#snort –vde
#snort –v –d – e
```

Dengan menambahkan beberapa switch –v, -d, -e akan menghasilkan beberapa keluaran yang berbeda, yaitu :

-v untuk melihat TCP/IP header paket yang lewat

-d untuk melihat isi paket

-e untuk melihat header link layer paket seperti Ethernet header

2. Packet logger mode

Packet logger mode berfungsi untuk mencatat semua paket yang lewat dijaringan yang kemudian akan dianalisa. Bahkan dapat menyimpan paket ke dalam disk. Sehingga perlu diinisialisasikan terlebih dahulu logging direktorinya pada file configurasi snort. Contoh dari perintahnya yaitu :

```
./snort –dev –l ./log
```

3. Network Intrusion Detection System (NIDS)

Dengan menggunakan *network Intrusion Detection System (NIDS)* tidak diperlukan lagi untuk menyimpan seluruh paket yang datang pada sebuah jaringan. Karena pada mode ini data yang disimpan atau ditampilkan adalah paket-paket yang berbahaya dengan cara mengkonfigurasi file snort.conf terlebih dahulu. Berikut ini adalah perintahnya dalam mengkonfigurasi snort.conf :

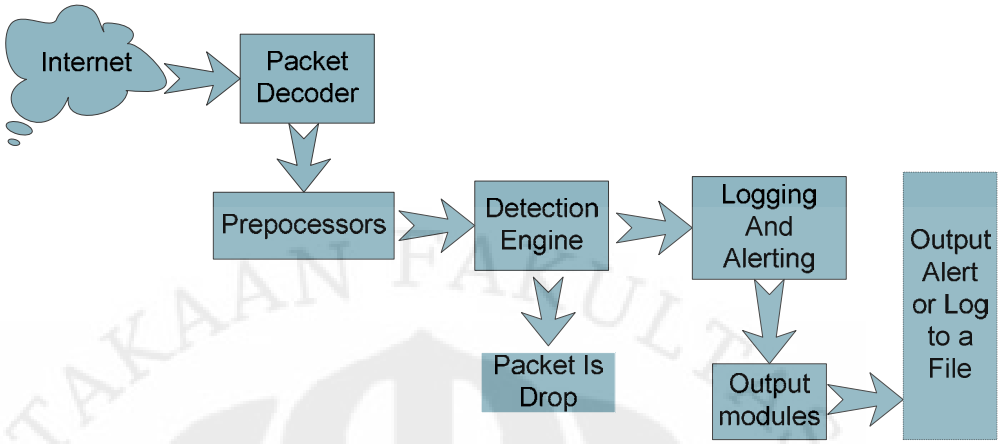
```
./snort –c snort.conf
```

2.6.1 Komponen-komponen Snort

Snort adalah logical yang dapat dibagi bagi menjadi beberapa komponen. Komponen ini yang nantinya akan bekerja bersama-sama untuk mendeteksi serangan khusus dan menampilkan keluaran yang diinginkan dari format *detection system*.

Snort merupakan bagian dari IDS yang terdiri dari beberapa komponen :

1. Packet Decoder
2. Preprocessors
3. Detection Engine
4. Logging and Alerting System
5. Output Modules



Gambar 2.1 Komponen-komponen Snort [5]

2.6.1.1 Packet Decoder

Packet Decoder mengambil paket dari berbagai jenis perangkat jaringan dan mempersiapkan paket data untuk dapat masuk ke preprocessed atau untuk dikirim ke mesin deteksi (*Detection Engine*).

2.6.1.2 Preprocessors

Preprocessors adalah komponen atau *plug-ins* yang dapat digunakan dengan Snort untuk mengatur atau memodifikasi paket data sebelum *Detection Engine* melakukan beberapa operasi untuk mengetahui apakah paket sedang digunakan oleh penyusup. Beberapa preprocessors juga melakukan deteksi yang ditemukan oleh anomali dalam paket header dan menghasilkan alert. Preprocessors sangat penting bagi setiap IDS untuk mempersiapkan paket data yang harus dianalisis terhadap *Rule* dalam *Detection Engine.*

2.6.1.3 The Detection Engine

Detection Engine bagian terpenting dari snort. Tanggung jawabnya adalah untuk mendeteksi jika ada aktivitas intrusi dalam sebuah paket. *Detection Engine* menggunakan *rule* snort untuk tujuan ini. *Rule* dibaca dalam struktur data internal

atau *rule* dapat dirangkaikan di mana *rule-rule* tersebut dicocokkan atau dibandingkan dengan semua paket. Jika sebuah paket cocok dengan *rule* apa pun, maka tindakan yang tepat diambil tetapi jika tidak paket dibuang. Tindakan yang tepat mungkin akan mendata paket (*Logging packet*) atau menghasilkan alert. *Detection Engine* adalah *time-critical* penting dari Snort. Tergantung pada seberapa kuat mesin dan berapa banyak *rule* yang telah ditetapkan, mungkin diperlukan jumlah waktu yang berbeda untuk merespon paket yang berbeda. Jika lalu lintas (*traffic*) di jaringan terlalu tinggi, maka ketika Snort NIDS bekerja dalam mode ini, mungkin ada beberapa paket yang didrop dan mungkin tidak mendapatkan *real-time* respon yang benar. Beban pada *Detection Engine* tergantung pada faktor berikut :

• Jumlah *rule*
• Kekuatan mesin yang menjalankan Snort
• Kecepatan bus internal yang digunakan dalam mesin Snort
• Beban pada jaringan

2.6.1.4 Logging and Alerting System

Tergantung pada apa yang *Detection Engine* temukan dalam sebuah paket, paket digunakan untuk mencatat aktivitas atau menghasilkan peringatan (*alert*).

2.6.1.5 Output modules

Output modul atau *plug-in* dapat melakukan operasi yang berbeda-beda tergantung pada bagaimana ingin menyimpan output yang dihasilkan oleh *logging* dan sistem alert dari snort.

2.7 SUPPORTED PLATFORMS

Snort didukung pada sejumlah *platform* perangkat keras dan sistem operasi. Saat ini Snort tersedia untuk sistem operasi berikut:

• Linux
• OpenBSD
• FreeBSD

• NetBSD
• Solaris (both Sparc and i386)
• HP-UX
• AIX
• IRIX
• MacOS
• Windows

2.8 JENIS SERANGAN

Ada beberapa jenis dan teknik serangan yang dapat mengganggu keamanan jaringan komputer, diantaranya : [3]

a) Denial of Service (DOS)

Merupakan sebuah jenis serangan terhadap sebuah komputer atau *server* di dalam jaringan internet. DOS ini bekerja dengan cara menghabiskan *resource* yang dimiliki oleh komputer tersebut sampai akhirnya komputer tersebut tidak dapat menjalankan fungsinya dengan benar yang secara tidak langsung mencegah pengguna lain untuk memperoleh akses layanan dari komputer yang diserang tersebut. DOS ini akan menyerang dengan cara mencegah seorang pengguna untuk melakukan akses terhadap sistem atau jaringan yang dituju. Ada beberapa cara yang dilakukan oleh DOS untuk melakukan serangan tersebut, yaitu:

- Membajiri *trafic* atau lalu lintas jaringan dengan banyaknya data-data sehingga lalu lintas jaringan yang datang dari pengguna yang terdaftar menjadi tidak dapat masuk ke dalam sistem jaringan. Biasanya teknik ini disebut sebagai *traffic flooding*.
- Membanjiri jaringan dengan cara *merequest* sebanyak-banyaknya terhadap sebuah layanan jaringan yang disediakan oleh sebuah *client* sehingga request yang datang dari para pengguna terdaftar tidak dapat dilayani aleh layanan tersebut. Biasanya teknik ini disebut sebagai *request flooding*.

- Menggangu komunikasi antara sebuah *client* dan kliennya yang terdaftar dengan menggunakan banyak cara, termasuk dengan cara mengubah informasi konfigurasi sistem bahakan adanya perusakan fisik terhadap komponen dan server.

b) Telnet

Telnet tergolong unik yang dirancang dengan mengecualikan pencatatan *rlogin*. Telnet dirancang untuk memungkinkan seorang *user log in* ke mesin lain dan mengeksekusi perintah disana. Telnet seperti halnya *rlogin* bekerja seperti halnya pada konsol mesin remote tersebut, seolah-olah secara fisik berada di depan mesin remote tersebut, menyalakan, dan mulai bekerja.

c) Port Scanning

Merupakan sustu proses untuk mencari dan membuka port pada suatu jaringan komputer. Hasil *scanning* tersebut akan didapatkan letak kelemahan sistem tersebut. Pada dasarnya sistem *port scanning* mudah untuk dideteksi, namun penyerang akan menggunakan berbagai metode untuk menyembunyikan serangan.

d) IP-Spoofing

IP Spoofing juga dikenal sebagai *Source Address Spoofing*, yaitu pemalsuan alamat *IP attacker* sehingga sasaran menganggap alamat *IP attacker* adalah alamat IP dari host di dalam network bukan dari luar network. Misalkan *attacker* mempunyai IP address type A 66.25.xx.xx ketika *attacker* melakukan serangan jenis ini maka *Network* yang diserang akan menganggap *IP attacker* adalah bagian dari Networknya misal 192.xx.xx.xx yaitu IP type C. IP Spoofing terjadi ketika seorang *attacker* 'mengakali' *packet routing* untuk mengubah arah dari data atau transmisi ke tujuan yang berbeda. Paket untuk routing biasanya di transmisikan secara transparan dan jelas sehingga membuat *attacker* dengan mudah untuk memodifikasi asal data ataupun tujuan dari data. Teknik ini bukan hanya dipakai oleh *attacker* tetapi juga dipakai oleh para *security profesional* untuk men-*tracing* identitas dari para *attacker*.

.

e) ICMP flood

Melakukan ekploitasi sistem agar dapat membuat suatu target *client* menjadi *crash* yang dilakukan oleh penyerang. Sehingga menjadi *crash* karena diakibatkan oleh pengiriman sejumlah paket yang besar kearah targer *client. Exploting* sistem ini dilakukan dengan mengirimkan suatu perintah *ping* dengan tujuan *broadcast* atau *multicast* di mana si pengirim dibuat seolah-olah adalah target *client.* Semua pesan balasan dikembalikan ke target *cient.* Hal inilah yang membuat target *client* menjadi *crash* dan menurunkan kinerja jaringan. Bahkan hal ini dapat mengakibatkan *denial of service.*

f) UDP Flood

Pada dasarnya mengkaitkan dua sistem tanpa disadarinya. Dengan cara spoofing, *User Datagram Protocol* (UDP) flood attack akan menempel pada servis UDP chargen di salah satu mesin, yang untuk keperluan "percobaan" akan mengirimkan sekelompok karakter ke mesin lain, yang di program untuk meng-*echo* setiap kiriman karakter yang di terima melalui *service chargen*. Karena paket UDP tersebut di spoofing antara ke dua mesin tersebut, maka yang terjadi adalah banjir tanpa henti kiriman karakter yang tidak berguna antara ke dua mesin tersebut. Untuk menanggulangi UDP flood, kita dapat men-disable semua servis UDP di semua mesin di jaringan, atau yang lebih mudah memfilter pada firewall semua servis UDP yang masuk.

g) Base Orifice

Base Orifice adalah sebuah alat bantu administrasi komputer dari jarak jauh yang dapat digunakan untuk mengontrol keluarga sistem operasi Microsoft Windows, yang dikembangkan oleh kelompok peretas professional Cult of the Dead Cow. Back Orifice dirilis pertama kali untuk platform Windows NT pada tahun 1997. Namanya merupakan pelesetan dari Microsoft BackOffice Server. Pada tahun 1999, grup yang sama merilis versi baru, yang disebut sebagai Back Orifice 2000 atau sering disebut **BO2K**. Meskipun pada dasarnya alat bantu ini merupakan salah satu bentuk dari *Trojan horse*, yang dapat digunakan untuk mendapatkan akses dan kontrol penuh terhadap mesin

target, program ini menawarkan banyak fitur, khususnya untuk mengendalikan sistem operasi Windows NT. Tampilan yang digunakannya sangatlah mudah dan sederhana, sehingga para peretas pemula pun dapat menggunakannya.

Untuk mendapatkan sistem yang benar-benar aman bukan hanya membenahi di jaringan external tetapi harus juga membenahi jaringan internal yang sesuai dengan kebijakan keamanan jaringan.

BAB III

DESAIN TOPOLOGI IDS

IDS merupakan suatu sistem yang memiliki kemampuan untuk menganalisa data secara *realtime* dalam mendeteksi, mencatat (*log*) dan menghentikan penyalahgunaan dan penyerangan. IDS merupakan *security tools* yang dapat digunakan untuk menghadapi aktivitas *hackers*. IDS ini mampu memberikan peringatan kepada administrator apabila terjadi suatu serangan atau penyusupan di dalam jaringan, bahkan peringatan itu dapat pula menunjukkan alamat IP dari sebuah sistem penyerang.

3.1. PERANCANGAN SISTEM

Perancangan sistem yang akan digunakan untuk merancang suatu sistem yang dapat mendeteksi adanya penyusup ataupun serangan yaitu *Intrusion Detection System*, yang sebelumnya membutuhkan *tools* atau komponen yang diperlukan untuk membangun sistem tersebut yang nantinya akan bekerja sama untuk mendapatkan hasil yang maksimal. Walaupun sebenarnya IDS tersebut sudah dapat mendeteksi penyusup di dalam jaringan hanya dengan menggunakan *tools* snort dan winpcap tetapi dengan hanya menggunakan snort dan winpcap tersebut sulit bagi administrator untuk dapat menganalisa alert maupun logs, sehingga untuk mendapatkan IDS yang secara maksimal dapat bekerja yaitu dibutuhkan komponen-komponen sebagai berikut :

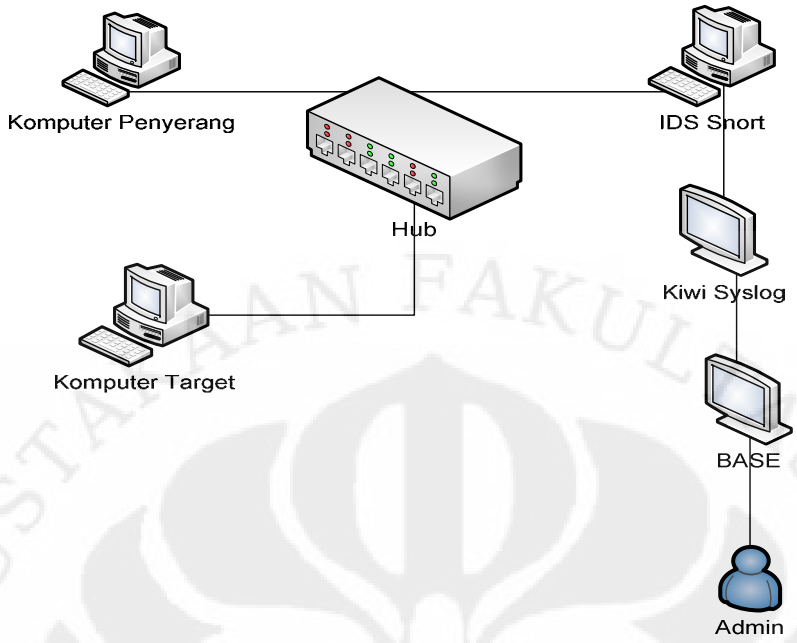
1. Snort 2_8_6 installer
   http://dl.snort.org/snort-current/Snort_2_8_6_Installer.exe
2. WinPcap 4.1.1
   http://www.winpcap.org/install/default.htm
3. Kiwi Syslog Server 9.0.3
   http://kiwisyslog.com/kiwi-syslog-server-download/
4. BASE 3.1
   http://www.brothersoft.com/base-download-170762.html
5. ADODB 504a
   http://adodb.sourceforge.net/

6. PHPlot 5.1.2
   http://sourceforge.net/projects/phplot/
7. Pear Image Graph 0.7.2
   http://pear.php.net/package/Image_Graph/download
8. XAMPP 1.7.3
   http://www.softpedia.com/get/Internet/Servers/Server-Tools/XAMPP.shtml

Apabila *tools* yang diinginkan sudah dapat terpenuhi, maka IDS yang akan dibangun adalah IDS yang dapat juga menyimpan alert dalam database dan IDS pun lebih *user friendly*

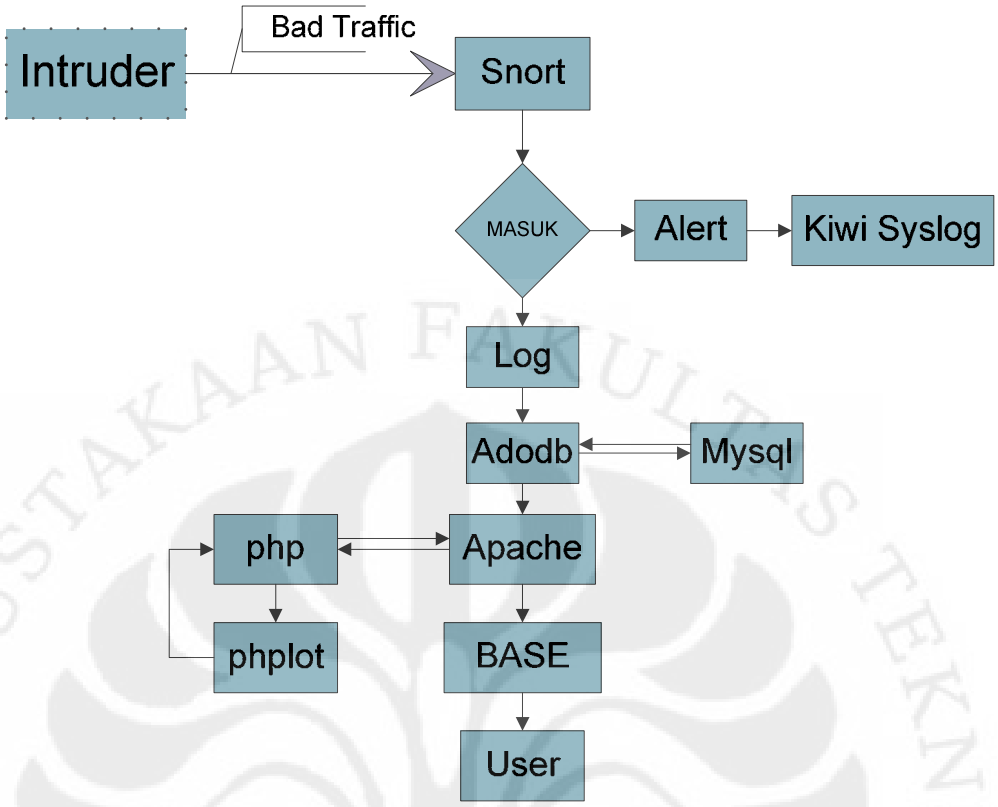
Untuk membangun sebuah IDS pada sistem Microsoft Windows memerlukan beberapa komponen yang perlu diintegrasikan menjadi satu kesatuan sistem.

Jika sudah mendapatkan semua program yang diinginkan, maka dengan mudah sistem IDS dapat dikonfigurasikan sesuai dengan keinginan. IDS ini nantinya dapat berfungsi sebagai pendeteksi adanya serangan terhadap server maupun client. Serangan ini dapat berupa *Denial of Service (DOS), port scanning,* dan *ip scan.* Serangan yang terjadi ini nantinya akan dimunculkan di dalam sebuah program yaitu Kiwi Syslog dan dimunculkan dalam sebuah interface berbasis web yaitu BASE. Gambar 3.1 merupakan rancangan sederhana dari IDS yang dibuat :



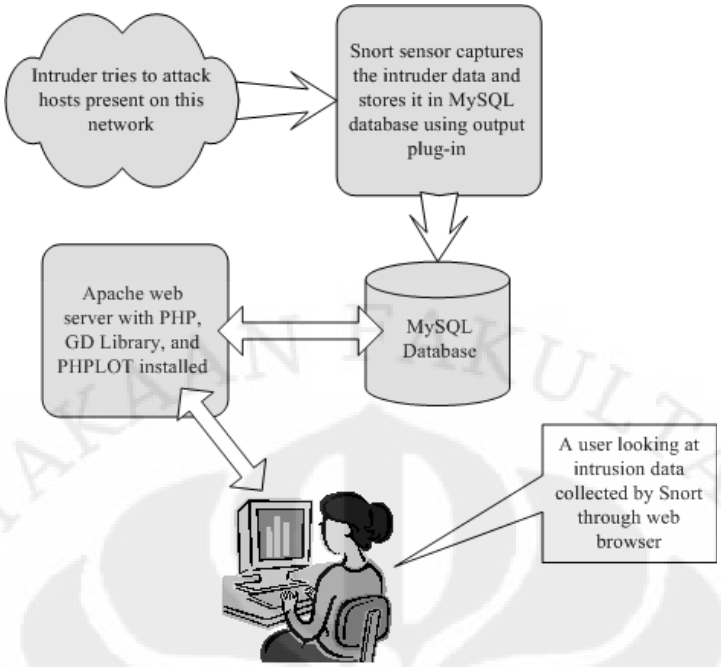
Gambar 3.1 Data Flow Diagram IDS Server Level 0

Pada jaringan tersebut digunakan *device* penghubung yaitu dengan menggunakan Hub, sedangkan untuk server jaringan menggunakan Windows 7, serta database MySQL sebagai media penyimpanan data pada server. Prinsip kerja jaringan diatas yaitu komputer penyerang mencoba untuk melakukan penyerangan terhadap komputer *client*, dimana komputer penyerang nantinya akan melewati sebuah hub yang kemudian serangan tersebut akan terdeteksi oleh komputer server yang telah dipasangi IDS snort. Komputer penyerang yang telah terdeteksi oleh snort, akan ditampilkan di Kiwi Syslog sebagai *alert* terhadap serangan yang terjadi, kemudian serangan tersebut akan disajikan pada sebuah interface berbasis BASE yang akan menampilkan dalam bentuk grafik yang dapat dimengerti oleh admin.



Gambar 3.2 Data Flow Diagram IDS Server Level 1

Sistem IDS yang akan dibangun seperti ditampilkan dalam Gambar 3.2. dengan menggunakan komponen WinPCap, Snort, MySQL database, Apache, PHP, BASE, dan phplot. Data ditangkap dan dianalisis oleh Snort. Snort kemudian menyimpan data ini dalam database MySQL menggunakan *database output plug-in*. *Apache web server* membutuhkan bantuan dari BASE, PHP, dan paket phplot untuk menampilkan data ini dalam *browser* ketika *administrator* terhubung ke Apache. Lalu administrator dapat membuat berbagai jenis *query* pada *form* yang ditampilkan di halaman web untuk menganalisis, arsip, dan grafik.



Gambar 3.3 Blok diagram IDS yang terdiri dari Snort, MySQL, Apache, PHP, dan phplot.[5]

3.2 INSTALLASI DAN KONFIGURASI IDS SOFTWARE

Snort merupakan IDS yang *Open source* yang terdiri dari beberapa komponen yang saling bekerja sama. Bagi pengguna yang terbiasa dengan lingkungan *Graphical User Interface* (GUI) maka akan mengalami sedikit kesulitan dalam menggunakan IDS Snort ini karena memang snort ini merupakan *software* yang masih berbasis pada *command-line*, maka diperlukan beberapa *software* pihak ketiga yang memberikan GUI untuk Snort, misalnya *IDScenter* untuk Microsoft Windows, dan BASE yang berbasis PHP yang dapat diakses melalui web browser. Snort dapat diperoleh atau didownload pada situs resminya yaitu http://www.snort.org. Snort ini dapat diimplementasikan dalam jaringan yang *multiplatform.* Berikut ini adalah cara menginstallasi Snort dan Rules Snort : [4]

1) Download Snort
   http://www.snort.org/downloads/
2) Setelah mendapatkan *installer* untuk snort, maka lakukan proses instalasi dengan men-*duoble* klik ikon pada snort tersebut.

3) Menjalankan Snort yaitu dengan perintah

   c:\snort\bin\snort -iX -s -l c:\snort\log\ -c c:\snort\etc\snort.conf (gantilah X dengan *Device Interface number*)

   tetapi sebelum menjalan snort tersebut, maka membutuhkan beberapa langkah lagi yaitu dengan mengkonfigurasikan *rules snort* agar snort dapat bekerja dengan baik. Rules snort tersebut dapat diperoleh pada http://www.snort.org/snort-rules/?#rules, tetapi sebelum men*download* rules snort tersebut maka terlebih dahulu harus melakukan registrasi terhadap situs snort yaitu http://www.snort.org

4) Setelah berhasil men*download* rules snort tersebut, ekstraklah *snort rules* ke direktori *C:\snort*. Proses akan berhasil ditandai dengan adanya beberapa rules berformat *.rules* pada direktori *C: \snort\rules*.

5) Walaupun rules snort tersebut sudah terekstrak dengan baik di folder snort, maka perlu melakukan beberapa penambahan maupun pengeditan pada snort.conf dengan Notepad++. Pengeditan maupun penambahan ini bermanfaat untuk *mengenerate* rules snort ke dalam Kiwi Syslog agar *alert* yang tertangkap dari adanya serangan di dalam jaringan dapat terbaca oleh Kiwi Syslog. Berikut ini adalah beberapa yang harus dilakukan pengeditan maupun penambahan pada snort.conf yaitu :

   `C:/Snort/etc/snort.conf`

   - Arahkan direktori rule berada.

     ```
     # Path to your rules files (this can be a relative
     path)
     # Note for Windows users: You are advised to make
     this an absolute path,
     # such as: c:\snort\rules
     var RULE_PATH c:\snort\rules
     ```

   - Logging database ke database MySQL

     ```
     # See the README.database file for more
     information about configuring
     # and using this plugin.
     output database: alert, mysql, user=monik
     password=jakarta dbname=snort_log host=localhost
     ```

```
output database: log, mysql, user=monik
password=jakarta dbname=snort_log host=localhost
```

- Arahkan direktori classification sebagai berikut :

```
# Note for Windows users: You are advised to make
this an absolute path,
# such as: c:\snort\etc\classification.config
#
include c:\snort\etc\classification.config
```

- Arahkan direktori reference sebagai berikut :

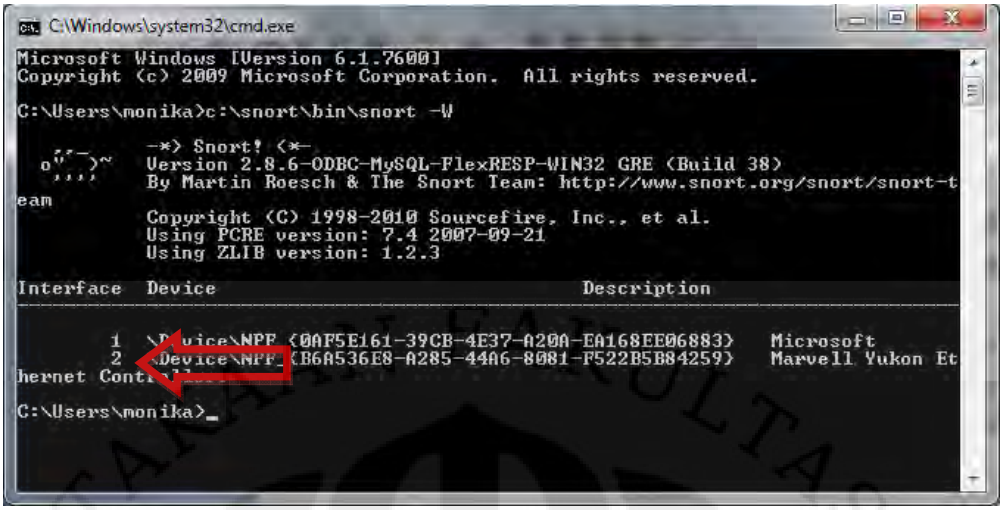
```
# Include reference systems
# Note for Windows users: You are advised to make
this an absolute path,
# such as: c:\snort\etc\reference.config
#
include c:\snort\etc\reference.config
```

Setelah melakukan konfigurasi seperti diatas, snort akan mendeteksi *alert* dari beberapa *host network* karena di *snort.cof* dapat mengeset *var HOME_NET any.*

Setelah mengkonfigurasikan beberapa perintah pada snort.conf maka simpan dan tutup file tersebut. Salin file ini ke C:\snort\etc dan timpa file yang sudah ada.

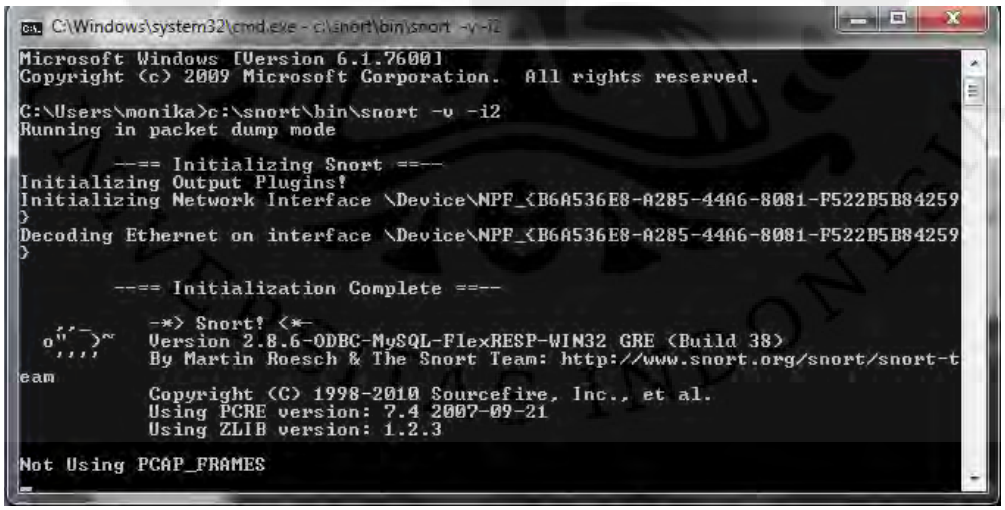
6) Setelah menyimpan semuanya, maka perlu memverifikasi operasi snort agar dapat mengetahui apakah snort dapat berjalan dengan baik. Berikut ini berupa *command Prompt* dan juga gambar dari perintah *command Prompt* tersebut.

```
c:\snort\bin\snort -W
```



Gambar 3.4 Verifikasi Operasi Snort

Setelah melakukan verifikasi terhadap operasi snort, maka sekarang jalankan kembali *command prompt* `c:\snort\bin\snort -v -i2`. Setelah beberapa detik maka snort akan menampilkan " *Not Using PCAP_FRAMES* ", sehingga snort sekarang dapat berjalan dan akan memberitahukan jika ada rules yang dipicu. Berikut ini adalah gambar atau tampilan dari *command prompt* :



.

Gambar 3.5 Initializing Snort

Untuk memicu alert snort tersebut maka perlu dilakukan ping ke alamat IP yang dituju agar lebih mengetahui apakah snort mulai dapat berjalan dengan baik. Apabila semua sudah dapat berjalan dengan baik, maka sekarang sudah dapat menjalankan snort.

7) Menjalankan Snort

Buka *command prompt* dengan perintah :

```
c:\snort\bin\snort -i2 -s -l c:\snort\log\ -c
c:\snort\etc\snort.conf
```

Jika sudah memasukkan informasi dengan benar, maka sekarang dapat menjalankan snort tersebut di administrator atau dapat membuat *shortcut* di desktop dengan nama file snort, kemudian klik kanan dan jalankan di administrator, maka akan tampil seperti Gambar 3.6 dibawah ini :

Gambar 3.6 Tampilan Snort yang Berjalan dengan Baik

Gambar yang telah muncul pesan di atas maka proses inisialisasi snort telah berjalan dengan baik, dari perintah di atas snort berjalan dalam mode *instrusion detection system,* maka jika terjadi serangan, snort akan memberikan peringatan atau *alerting.* Alerting ini dapat ditampilkan oleh Kiwi Syslog.

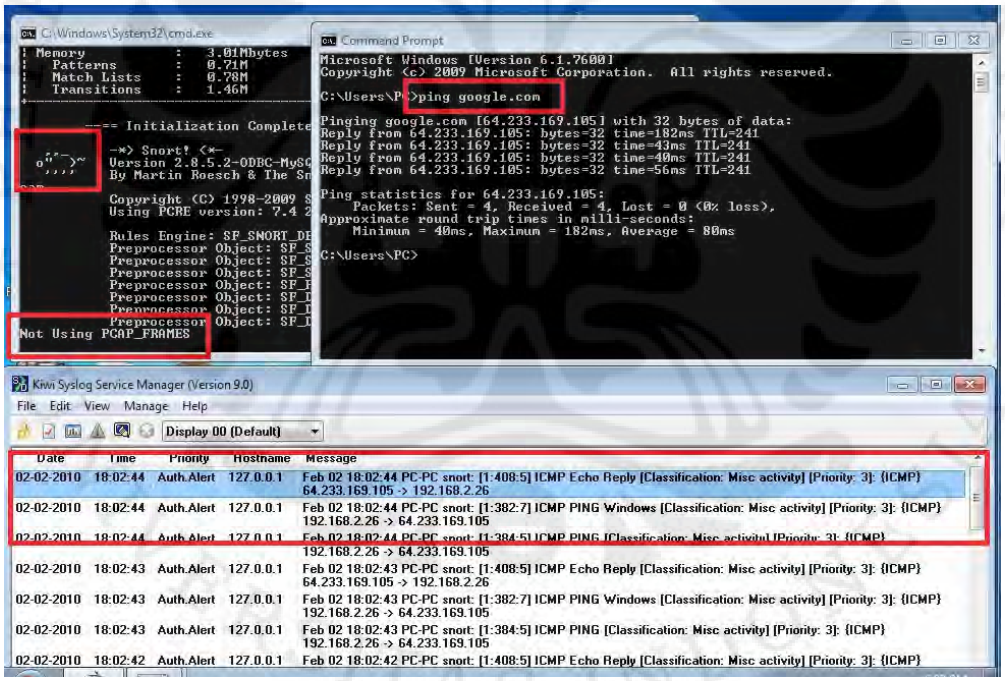
3.3 PENGOPERASIAN KIWI SYSLOG SEBAGAI ALERT DARI SNORT

Kiwi Syslog Server merupakan sebuah freeware Syslog Server untuk Windows. Kiwi Syslog ini dapat berfungsi sebagai log, *displays*, *alert*, dan melakukan tindakan lainnya di pesan syslog dan *SNMP traps*, dan dapat ditangkap dari host seperti firewall, router, switch,hub dan syslog lainnya yang aktif.

Cara untuk *mengenerate* agar Kiwi Syslog dapat terbaca yaitu dapat dikonfigurasikan pada snort.conf :

```
alert udp any any <> any 1:10000 (sid:10000001;)
alert tcp any any <> any 8080 (sid:10000002;)
```

Jika sudah dikonfigurasikan dengan baik, maka snort akan *mengenerate* kiwi syslog server untuk membaca adanya serangan yang masuk pada jaringan. Gambar 3.7 adalah keluaran *alert* snort di Kiwi Syslog :



Gambar 3.7 Contoh Keluaran Alert Snort di Kiwi Syslog

3.4 KONFIGURASI DATABASE DAN BASE

Modul report yang sudah banyak digunakan dan telah terintegrasi dengan baik dengan snort yaitu BASE yang digunakan untuk mengelola data-data *security event.* Berikut ini adalah beberapa keuntungan dari BASE yaitu

- Log-log yang sulit untuk dibaca akan menjadi mudah untuk dibaca.
- Data-data dapat dicari sesuai dengan kriteria tertentu.

Namun untuk menginstall BASE ini dibutuhkan beberapa *software* pendukung yaitu MySQL, PHP, dan Apache. Oleh karena itu untuk mendapatkan semua software tersebut, maka dapat digunakan dengan memakai XAMPP.

3.4.1 Installasi Xampp

Xampp merupakan *web development tool* yang terdiri dari *Apache, MySQL, PHP*, dan beberapa *add-on* yang langsung dirangkap menjadi satu, seperti : *PROFTPD FTP Server* (versi linux), *Filezilla FTP Server* (versi windows), *Webalizer, phpMyAdmin, SQLite*, dll. Beberapa *add-on* yang terpisah seperti *Perl* dan *Tomcat* juga disediakan oleh xampp. Xampp sendiri memiliki kemudahan dengan tersedianya kontrol panel yang mudah pada tampilannya.

3.4.1.1 Mysql

Mysql adalah database yang digunakan dan diinstall pada sistem berbasis Windows atau sistem operasi lain yang mendukung database Mysql. Database MyQSL ini nantinya akan digunakan untuk menyimpan *Alert* IDS. Ada beberapa alasan dalam pemilihan Mysql sebagai program database yaitu :

- Sifatnya yang *open source* dan murah.
- Cukup stabil pada *hardware* dengan spesifikasi yang relatif rendah.
- Untuk administrasi dan *maintenance* sistem database dibuat suatu *interface* berbasis web yang dibuat dengan bahasa pemrograman PHP.

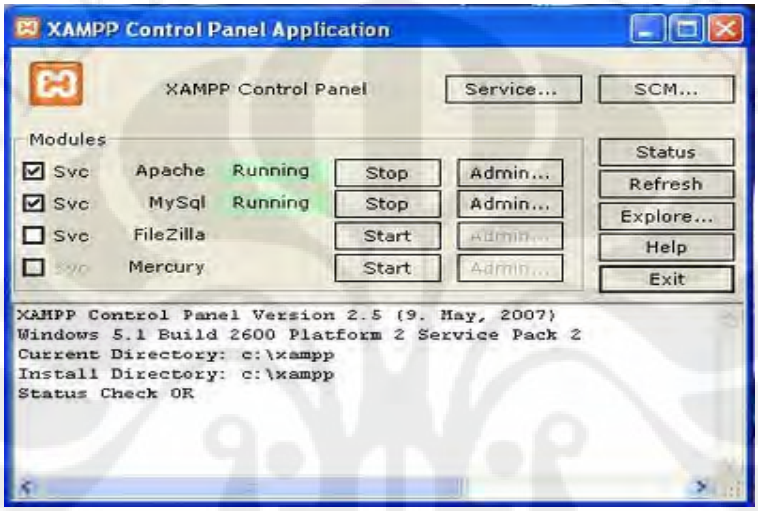
3.4.1.2 PHP (Personal Home Page)

PHP merupakan bahasa pemrograman berbasis web. Bahasa ini mempunyai kelebihan yaitu kompabilitasnya dengan berbagai macam jenis database, dukungan dengan berbagai macam jenis sistem operasi. PHP biasanya digunakan bersamaan dengan database Mysql. Mysql dengan PHP adalah dua hal yang tidak dapat dipisahkan. Fungsi dari PHP ini nantinya akan digunakan untuk menampilkan *alert* yang dihasilkan oleh snort. *Alert* tersebut nantinya akan ditampilkan dengan menggunakan BASE.

3.4.1.3 Web server Apache

Web server yang akan digunakan adalah web server apache. Webserver tersebut nantinya akan diintegrasikan bersama-sama dengan PHP.

Setelah melakukan installasi XAMPP tersebut maka jalankan Apache dan juga MySQL seperti pada Gambar 3.8 dibawah ini :



Gambar 3.8 XAMPP Option

3.4.1.4 Membuat database

Setelah selesai melakukan instalasi Xampp, langkah selanjutnya adalah membuat database pada MySQL yang akan digunakan snort untuk menyimpan *alert*, database yang akan dibuat diberi nama *snort_log* dan *snort_archive*. Berikut ini adalah proses pembuatan database yaitu :

1) Buka *command prompt* dengan mengetikkan *cmd* kemudian tekan *OK*.
2) Kemudian ketik beberapa perintah untuk dapat masuk ke Mysql serta membuat dua database yaitu *snort_log* dan *snort_archive*. Berikut ini adalah perintah yang dituliskan pada *command prompt* :

- Start MySQL

```
c:\user\monik>cd c:\xampp\mysql
```

```
c:\xampp\mysql>cd bin
```

- Login to MySQL

```
c:\xampp\mysql\bin>mysql -u root -p
```

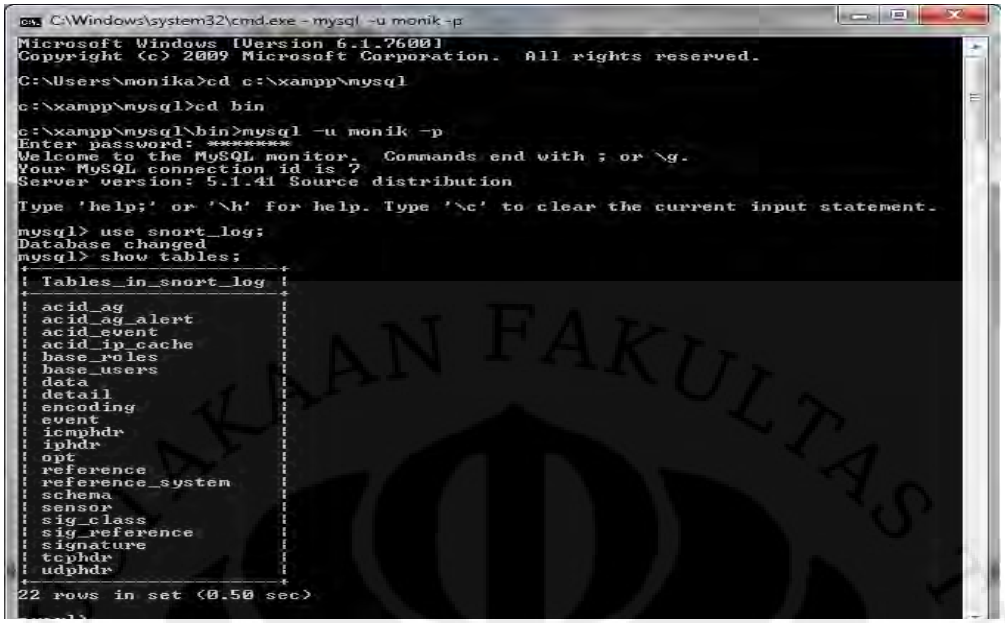
- Membuat Database

```
mysql> create database snort_log;
Query OK, 1 row affected <0,00 sec>
mysql> create database snort_archive;
Query OK, 1 row affected <0,00 sec>
```

3) Setelah berhasil membuat database langkah selanjutnya adalah membuat tabel pada kedua database tersebut, snort sudah menyediakan beberapa *schema* database untuk berbagai tipe *platform database* seperti MySQL, MSSQL, POSTGRESQL, ORACLE, terletak pada direktori *C:\snort\schemas*. Dari *schema* tersebut dapat langsung di*compile* dengan menggunakan perintah sebagai berikut :

- `C:\user\monik>cd c:\xampp\mysql`
- `C:\xampp\mysql>cd bin`
- `C:\xampp\mysql\bin>mysql -D snort_log -u root -p < C:\xampp\snort\schemas\create_mysql`
- `Enter password: jakarta`
- `C:\xampp\mysql\bin>mysql -D snort_archive -u root -p < C:\xampp\snort\schemas\create_mysql`
- `Enter password: jakarta`

4) Untuk mengecek apakah kompilasi telah berhasil gunakan perintah berikut untuk melihat hasilnya :



Gambar 3.9 Pembuatan Database Berhasil

3.4.2 Installasi dan Konfigurasi BASE

- **Install ADODB**

ADODB, sebuah *library* abstraksi untuk menggabungkan PHP ke berbagai database seperti MySQL dan PostgresSQL. ADODB dapat diperoleh pada situs http://adodb.sourceforge.net. ADODB ini nantinya akan dipasang didalam konfigurasi BASE yang akan dijelaskan nanti.

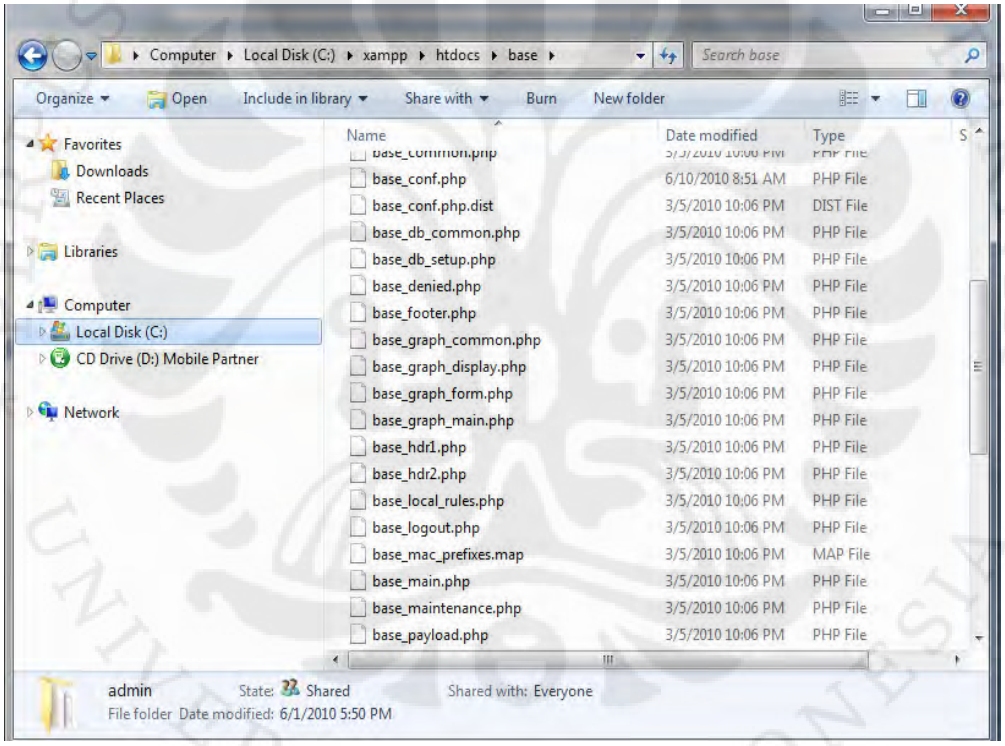
- **Install Pear**

Pear Image Graph adalah sebuah *software* yang menyediakan satu set yang dapat menciptakan grafik/plot berdasarkan data (*numeric*). Bentuk dari grafik ini yaitu berupa *bar, pie,* dan *line*. *Image Graph* kompatibel dengan PHP4 dan PHP5. Berikut ini adalah beberapa perintah yang digunakan untuk menginstall *Pear Image Graph :*

- `C:\xampp\php>cd PEAR`
- `C:\xampp\php\PEAR>install Image_Color`
- `C:\xampp\php\PEAR>install Image_Canvas`
- `C:\xampp\php\PEAR>install Image_Graph`

- **Install BASE**

BASE merupakan *PHP based analysis engine* yang berfungsi untuk mencari dan mengolah database dari *alert network security* yang dibangkitkan oleh perangkat lunak pendeteksi intrusi (IDS). Dapat diimplementasikan pada sistem yang mendukung PHP seperti linux, BSD, Solaris dan OS lainnya. BASE adalah perangkat lunak yang *open-source* dan didistribusikan dibawah lisensi GPL. Apabila telah men*download* BASE maka ekstrak file ke direktori `C:\Xampp\htdocs\base` yang terlihat pada Gambar 3.10 dibawa ini :



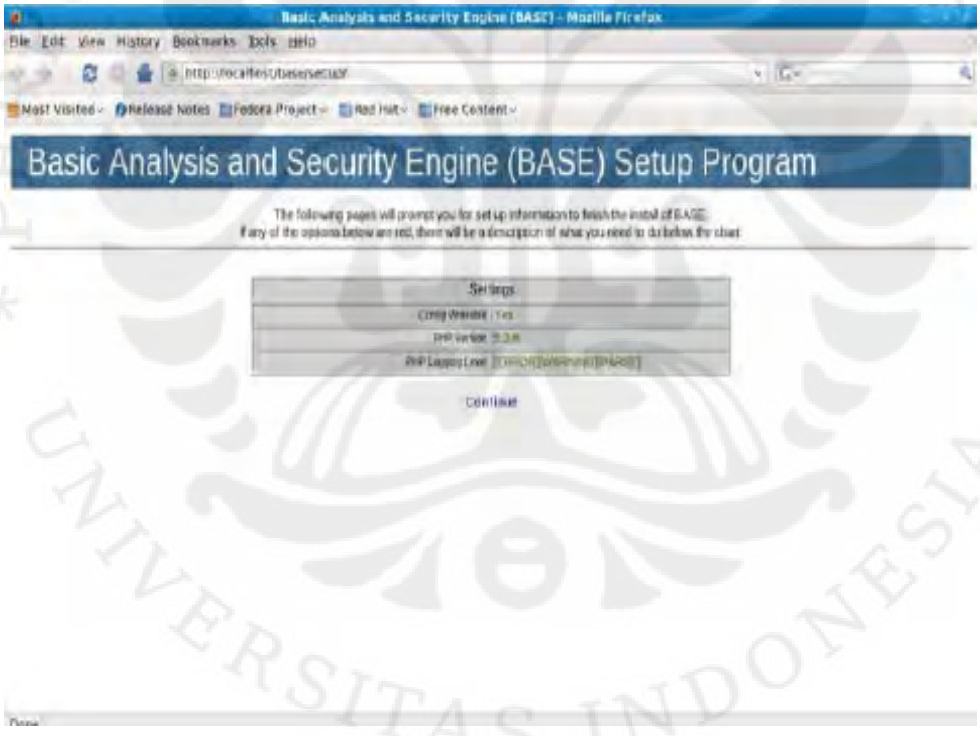
Gambar 3.10 Direktori BASE (Basic Analysis and Security Engine)

Kemudian selanjutnya adalah melakukan konfigurasi BASE dengan mengedit beberapa baris file base_conf.php menggunakan *notepad++* sebagai berikut :

```
$DBlib_path = "c:\adodb";
$alert_dbname = "snort_log";
$alert_host = "localhost";
```

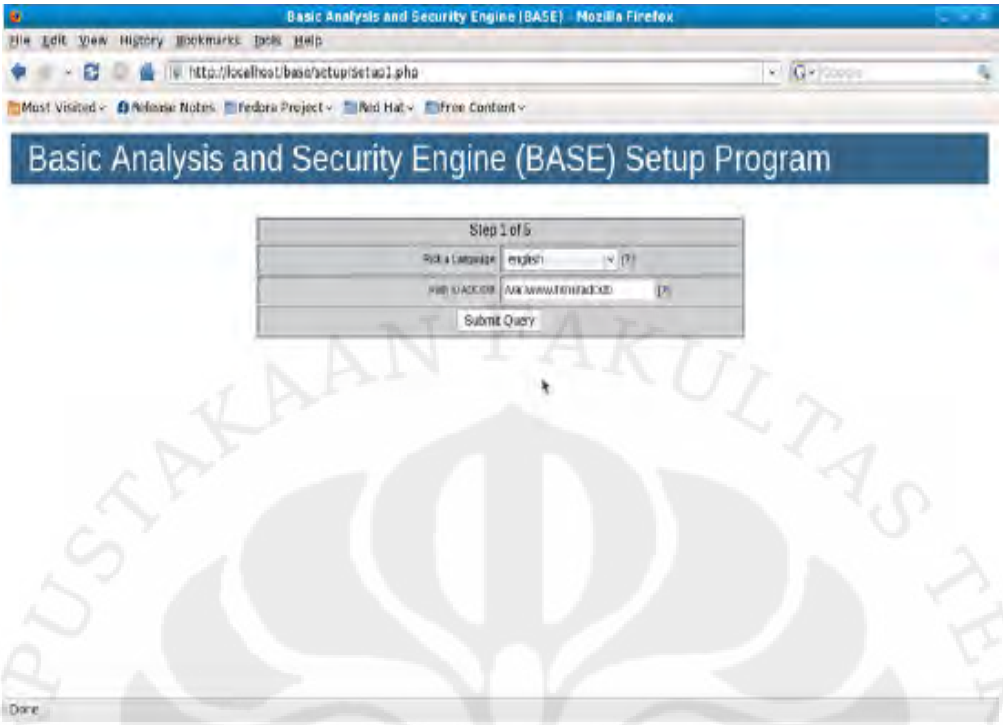
```
$alert_port = "3306";
$alert_user = "monik";
$alert_password = "jakarta";

$archive_dbname = "snort_archive";
$archive_host = "localhost";
$archive_port = "3306";
$archive_user = "monik";
$archive_password = "jakarta";

$ChartLib_path = "c:\phplot";
```

Setelah selesai melakukan konfigurasi simpan file tersebut. Buka *web browser* dan arahkan ke http://localclient/base/setup/ maka akan keluar tampilan seperti Gambar 3.11:



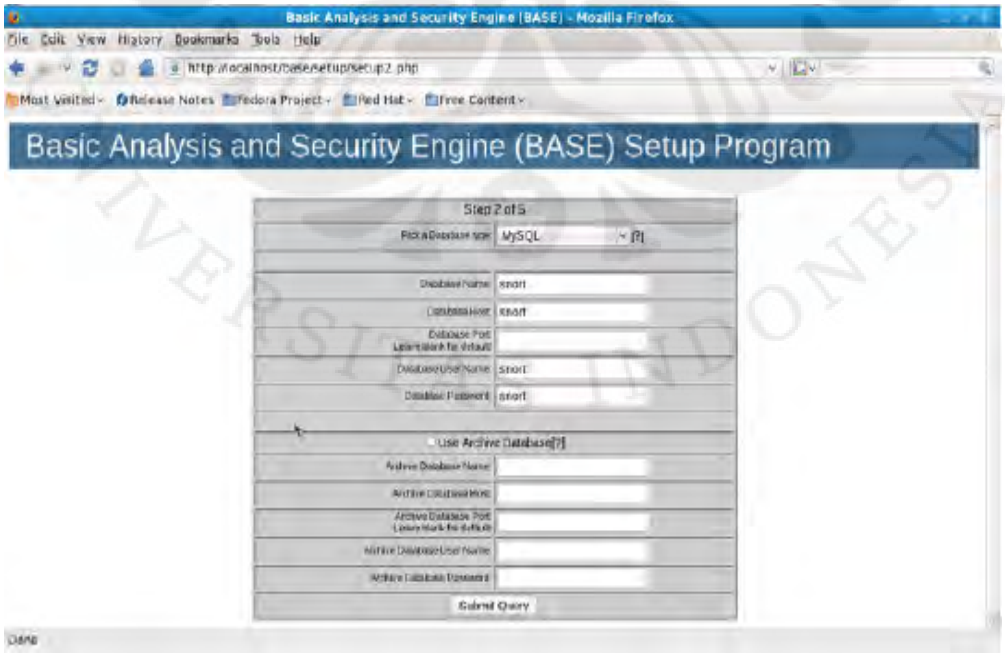
Gambar 3.11 Setup BASE

Pastikan tidak ada kesalahan/error pada settingan tersebut. Kemudian klik *continue* untuk mengisi path letak dari adodb seperti pada Gambar 3.12:



Gambar 3.12 Letak Path ADODB

Setelah itu masukkan konfigurasi MySQL seperti nama database, nama host, dan password. Seperti pada Gambar 3.13:



Gambar 3.13 Konfigurasi MySQL

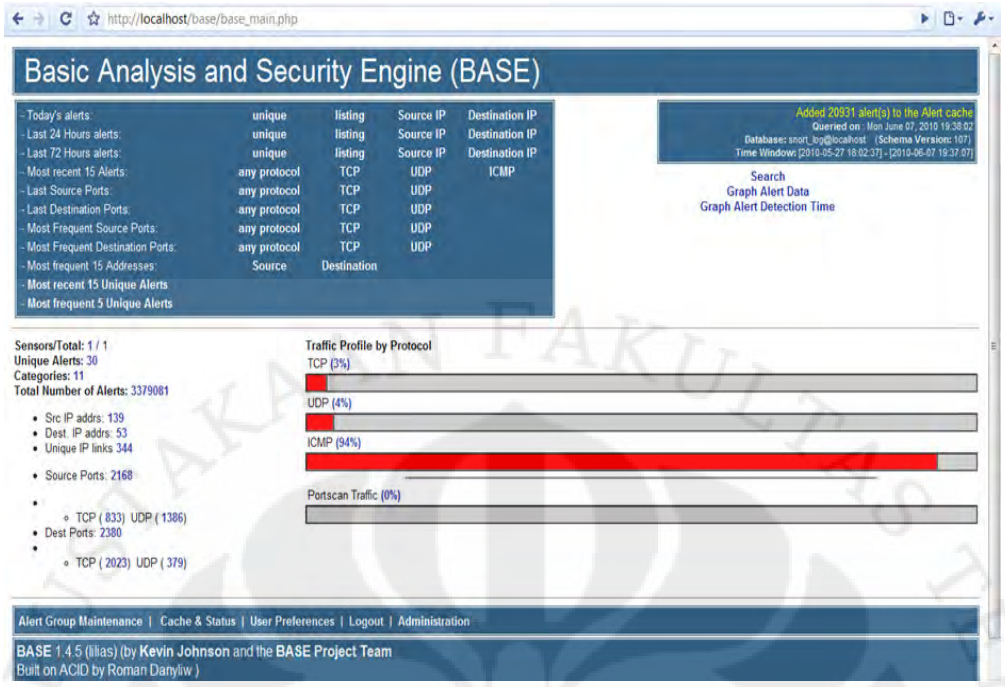
Langkah selanjutnya menambah *table* BASE ke dalam database snort, klik *Create BASE AG*

Gambar 3.14 Penambahan Tabel

Apabila tidak terjadi error dalam proses, maka akan muncul pesan

```
"Successfully created acid_ag",
"Successfully created acid_ag_alert",
"Successfully created acid_ip_cache",
"Successfully created acid_event".
"Successfully created base_roles".
"Successfully created base_user".
```

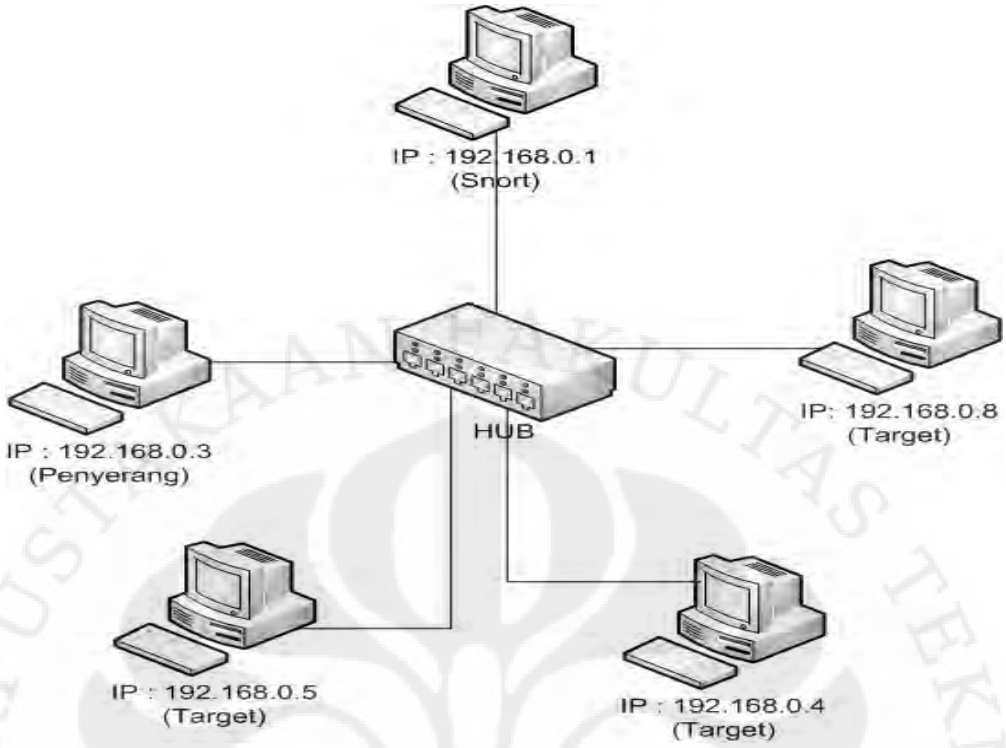
Database ini yang kemudian akan ditambahkan ke dalam database snort di MySQL. Snort akan memonitor jaringan dan akan menghasilkan *alert*, *alert* tersebut akan ditampilkan oleh BASE. Konfigurasi BASE selesai kemudian klik *"Main Page"* untuk masuk ke menu utama. Pada Gambar 3.15 dapat ditunjukkan tampilan awal dari BASE.



Gambar 3.15 BASE dengan alert snortnya

3.5 DISAIN JARINGAN

Pada skripsi ini, jaringan yang akan digunakan adalah jaringan sederhana yang terhubung dengan menggunakan sebuah Hub. Untuk mempermudah pengelompokkan maka komputer diberi IP Address. Sistem IDS snort inilah yang dipasang pada komputer dengan sistem operasi Windows 7. Pada komputer IDS ini yang dapat melakukan pendeteksian apabila terjadi gangguan atau serangan di dalam jaringan. Berikut ini adalah Gambar 3.16 dari disain jaringan IDS snort.



Gambar 3.16 Disain Jaringan IDS Snort

BAB IV

PENGUJIAN DAN ANALISA

Pada bagian ini akan dilakukan pengujian sistem yang sudah dibuat berdasarkan perancangan pada bab sebelumnya. Pengujian sistem dilakukan dengan melakukan beberapa serangan dan untuk mengetahui apakah IDS dapat bekerja dengan baik.

4.1 METODE DAN SKENARIO PENGUJIAN

Pengujian IDS pada skripsi ini dilakukan dengan dua metode untuk menguji apakah sistem dapat berfungsi dengan baik dan juga memiliki tingkat *reliability* yang sesuai. Dua metode tersebut yaitu :

1. *Functionality Test*
2. *Response Time*

Pengujian pada skripsi ini yaitu menggunakan skenario yang diinginkan yang dapat menganalisa data atau serangan dengan menggunakan 1 *client*, 2 *client*, dan 3 *client. Functionality Test* ini nantinya akan menguji apakah sistem IDS tersebut dapat berfungsi dengan baik yang sesuai dengan skenario yang diinginkan, untuk menghitung *response time* maka digunakan *software Wireshark. Response time* ini akan dihitung mulai dari terjadinya serangan sampai IDS memberikan respon dengan mengirimkan *alerting* ke Kiwi Syslog, sehingga untuk skenario yang pertama yaitu

4.1.1 Functionality Test

Functionality test bertujuan untuk menguji apakah sistem IDS ini dapat berfungsi dengan baik dan juga sesuai dengan kriteria yang diinginkan. Kriteria yang diinginkan tentu saja dapat mendeteksi ketika terdapat adanya serangan di dalam jaringan maka sistem IDS akan memberikan *alerting* yang kemudian mengirimkannya ke kiwi syslog . Pada *functionality test* akan dilakukan beberapa pengujian yang menggunakan tipe serangan yang berbeda yaitu *Network Surveying* yang menggunakan pengujian pada :

1. *IP Scan*
2. *Port Scanning*

Dan juga tipe dari *Enumeration Test* yaitu dilakukannya *Flooding*.

Karena ketika ingin melakukan sebuah serangan di dalam jaringan, maka pertama kali yang akan dilakukan yaitu mencari *ip* serta *port* yang terbuka agar dapat mengetahui *ip* dan *port* apa saja yang dapat diserang. Dalam skenario *functionality test* ini dilakukan terlebih dahulu yaitu mencari *ip* dan *port* yang terbuka, jadi apabila *functionality test* ini untuk *ip* dan *port* sudah dapat diketahui, maka scenario ini sesuai dengan yang diinginkan.

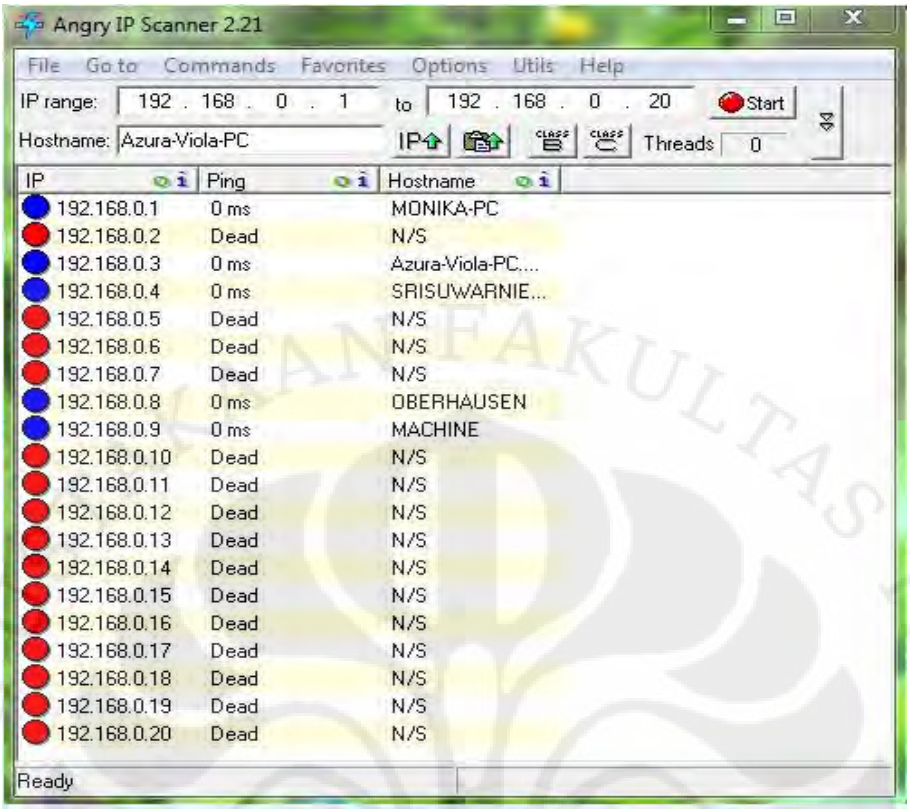
4.2 PERHITUNGAN DAN ANALISA

Perhitungan dan analisa dilakukan pada setiap skenario. Parameter yang dihitung adalah *Functionality Test* dan R*esponse Time*. Perhitungan pertama akan dilakukan pada skenario 1 yaitu *IP Scan* berdasarkan *functionality test.*

4.2.1 Functionality Test

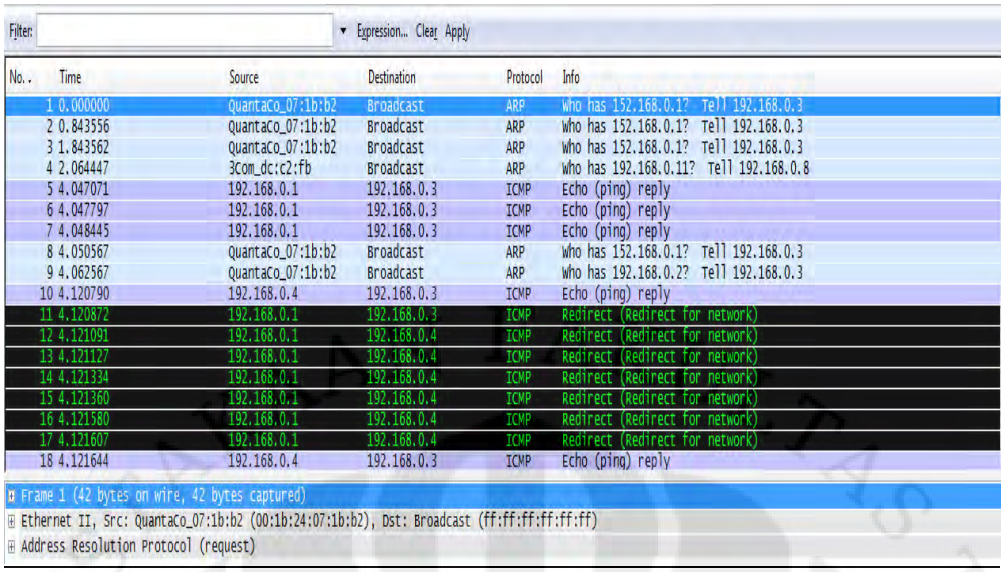
4.2.1.1 IP Scan

Ip Scan ini adalah suatu aplikasi yang dilakukan untuk melakukan suatu proses *scanning*/penelusuran IP pada sebuah jaringan internet. Tentu saja jaringan ini dapat berupa LAN (Local Area Network), MAN (Metropolitan Area Network), dan WAN (Wide Area Network), sehingga dengan ip scan ini dapat mengetahui adanya suatu *ip* atau *user* yang berada di dalam jaringan. Ip scan ini didapatkan dengan menggunakan sebuah *software* yang mampu mendeteksi adanya *ip* yang aktif di dalam jaringan yaitu dengan memakai *software Angry IP Scanner. Angry IP Scanner* merupakan sebuah *tools* yang digunakan untuk mencari *IP* yang hidup atau aktif dari *range IP* yang diinginkan. Selain itu *Angry IP Scanner* juga dapat melakukan pendeteksian port yang terbuka atau pun tertutup dari IP yang aktif, sehingga dengan menggunakan *software* ini dapat mencari target yang akan diserang. Gambar 4.1 adalah hasil dari percobaan yang menggunakan *software Angry IP Scanner* :



Gambar 4.1 Hasil IP Scan di Dalam Jaringan IDS

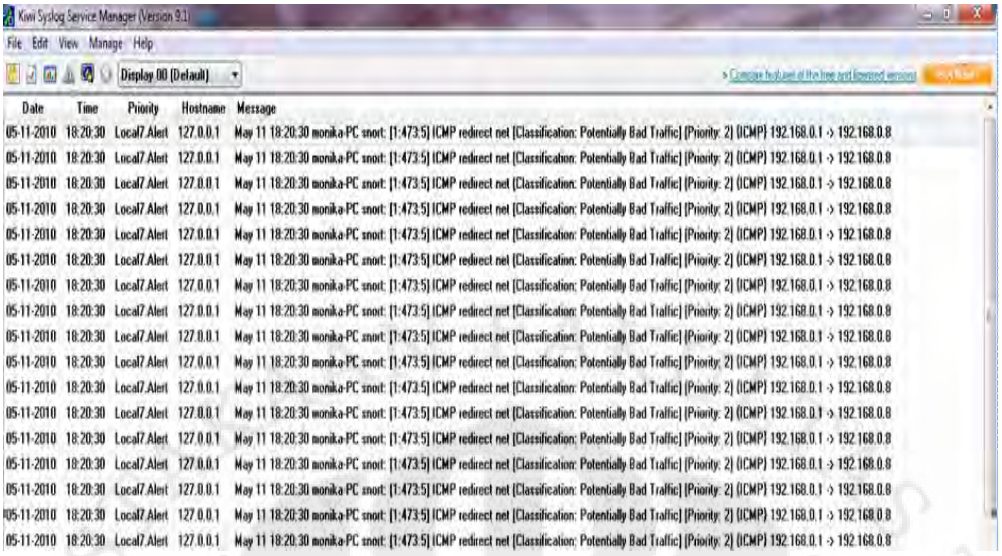
Hasil dari Gambar 4.1 menerangkan bahwa *IP range* yang dipasang yaitu berkisar antara 192.168.0.1 sampai dengan 192.168.0.20. Namun gambar tersebut memperlihatkan bahwa lingkaran yang berwarna biru adalah *IP* yang hidup atau aktif sedangkan lingkaran yang berwarna merah yaitu *IP* yang sedang tidak aktif. *IP* yang aktif tersebut yang nantinya akan dijadikan target untuk diserang. Sistem IDS ini digunakan pada komputer yang mempunyai *IP address* 192.168.0.1 yang dijadikan sebagai *gateway.* Gambar 4.2 adalah hasil tangkapan *wireshark* :



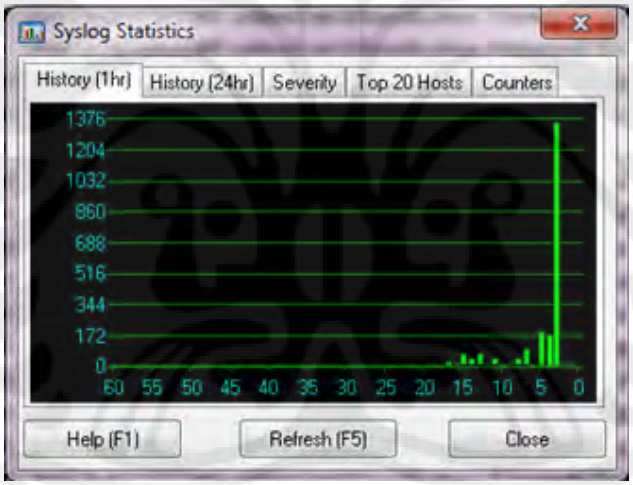
Gambar 4.2 Hasil Capture IP Scan

Hasil dari Gambar 4.2 tersebut menunjukan apabila menjalankan *IP Scan* yang berkisar antara 192.168.0.1 sampai dengan 192.168.0.20 maka *wireshark* dapat melakukan *monitoring* terhadap jaringan sehingga *wireshark* akan mencari paket mana saja yang akan *mereply* dari paket ICMP, maka itu dianggap sebagai IP yang sedang hidup atau aktif. Namun apabila alamat IP yang tidak aktif maka *wireshark* akan memberi tanda dengan kata *who has.*

Selain itu *IP Scan* ini dapat di deteksi oleh kiwi syslog yang memberikan *alerting* pada paket ICMP, dapat dikatakan bahwa *IP Scan* ini sama dengan apa yang ada pada *software Angry IP Scanner, Wireshark,* dan juga *Kiwi Syslog.* Gambar 4.3 adalah hasil *capture* dan garfik dari kiwi syslog :



Gambar 4.3 Hasil Capture Kiwi Syslog



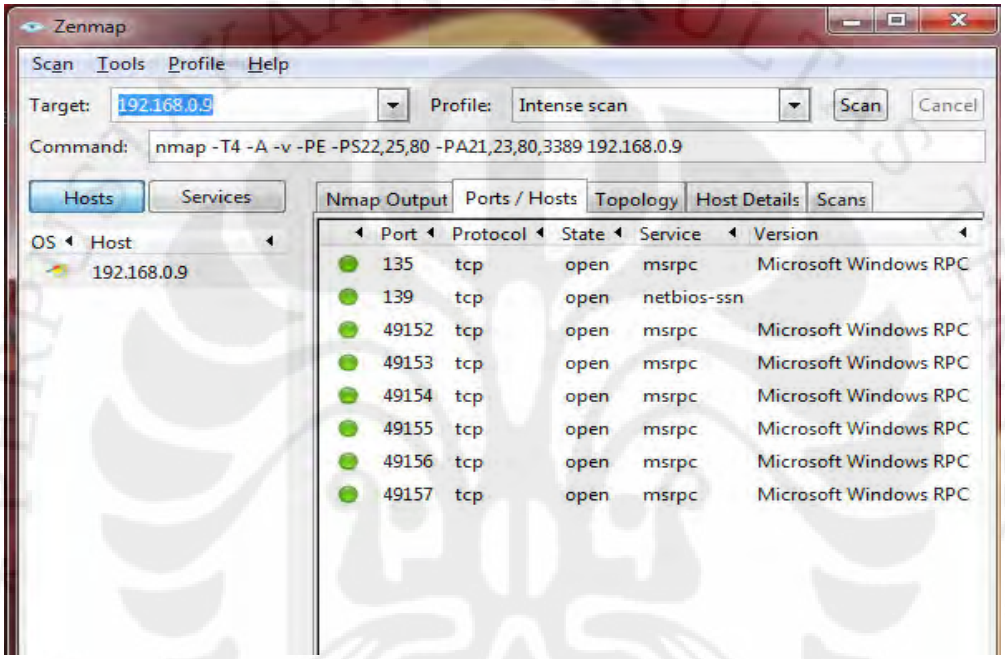
Gambar 4.4 Hasil Grafik IP Scan

Pada Gambar 4.3 menjelaskan suatu serangan pada jaringan tersebut telah terdeteksi oleh Kiwi Syslog sebagai suatu *Potentially Bad Trafic* yang termasuk pada *priority* 2 yang artinya cukup membahayakan bagi jaringan tersebut. *Source IP* ini yaitu berasal dari IP 192.168.0.1 menuju IP 192.168.0.8.

Pada Gambar 4.4 menjelaskan bahwa Kiwi Syslog juga mampu memperlihatkan pergerakkan *alert* yang tertangkap pada jaringan berupa grafik. Grafik ini juga menjelaskan jumlah *alert* terhadap waktu selama 60 menit.

4.2.1.2 Port Scan

Merupakan suatu proses untuk mencari dan membuka port pada suatu jaringan komputer. Hasil *scanning* tersebut akan didapatkan letak kelemahan sistem tersebut. Pengujian ini dilakukan dengan menggunakan alamat IP sebagai target yaitu 192.168.0.9. Gambar 4.5adalah hasil *port scanning* dengan alamat IP 192.168.0.9 :



Gambar 4.5 Hasil Capture Port Scanning 192.168.0.1

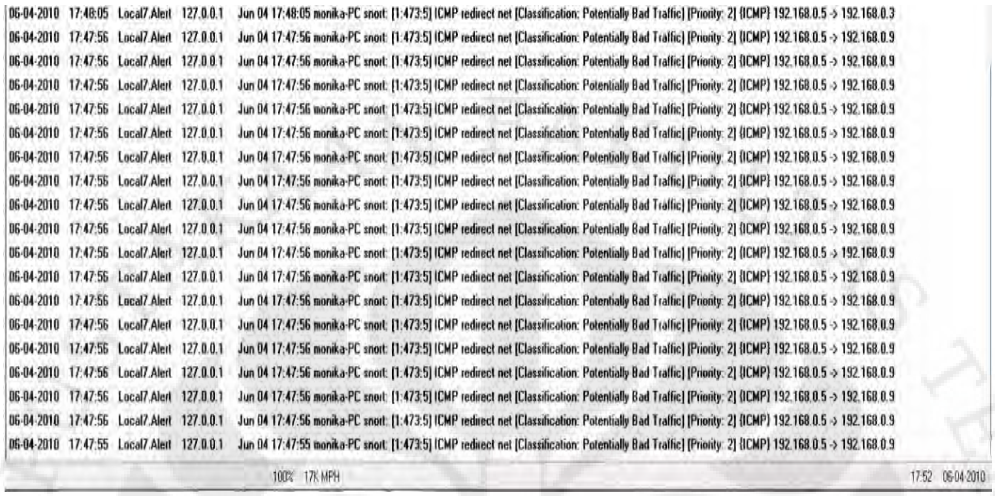
Gambar 4.5 tersebut menjelaskan bahwa pada alamat IP 192.168.0.9 terdapat beberapa port yang terbuka, di port yang terbuka inilah yang dapat diserang oleh penyerang. Dibawah ini adalah data-data hasil dari port scanning yang dilakukan oleh *Zenmap* terhadap *port-port* yang terbuka :

```
Initiating OS detection (try #1) against 192.168.0.9
NSE: Script scanning 192.168.0.9.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 17:47
Completed NSE at 17:48, 22.45s elapsed
NSE: Script Scanning completed.
```

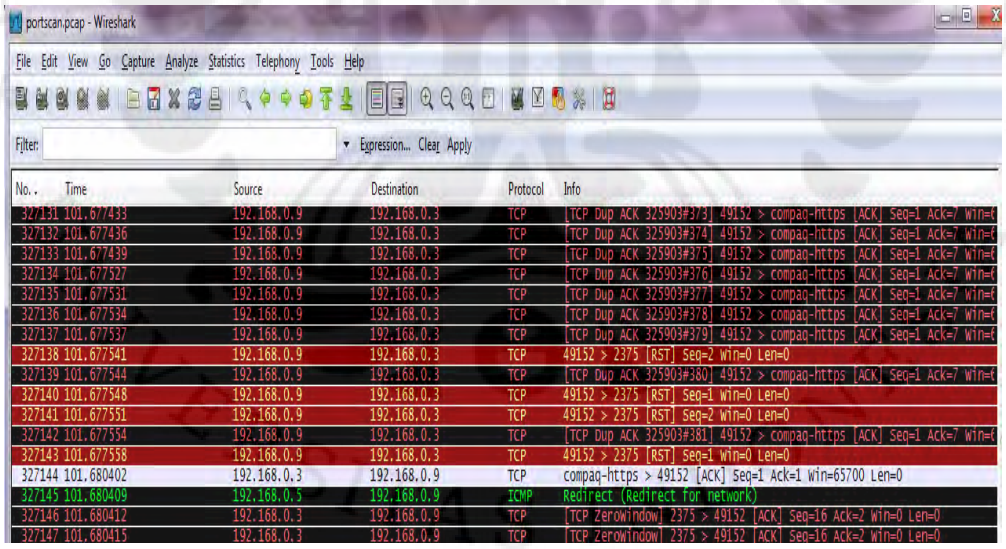
```
Nmap scan report for 192.168.0.9
Host is up (0.059s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE     VERSION
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn
49152/tcp open  msrpc       Microsoft Windows RPC
49153/tcp open  msrpc       Microsoft Windows RPC
49154/tcp open  msrpc       Microsoft Windows RPC
49155/tcp open  msrpc       Microsoft Windows RPC
49156/tcp open  msrpc       Microsoft Windows RPC
49157/tcp open  msrpc       Microsoft Windows RPC
MAC Address: 70:5A:B6:70:F8:5D (Compal Information (kunshan) CO.)
Device type: general purpose
Running: Microsoft Windows Vista|2008|7
OS details: Microsoft Windows Vista SP0 - SP2, Server 2008, or
Windows 7 Ultimate (build 7000)
Uptime guess: 0.178 days (since Fri Jun 04 13:32:13 2010)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=251 (Good luck!)
IP ID Sequence Generation: Busy server or unknown class
Service Info: OS: Windows

TRACEROUTE
HOP RTT      ADDRESS
1   59.14 ms 192.168.0.9

Read data files from: C:\Program Files\Nmap
OS and Service detection performed. Please report any incorrect
results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 105.42 seconds
           Raw packets sent: 1020 (45.578KB) | Rcvd: 33997
(1.417MB)
```

Dari hasil *port scan* dapat mengetahui berbagai informasi mengenai komputer target selain *port* yang terbuka tetapi juga dapat mengetahui Sistem Operasi dan *Mac Address* yang digunakan. *IP Scan* ini dapat terdeteksi pada Kiwi

Syslog yang mampu memberikan *alerting*. Gambar 4.6 adalah tampilan *alerting* dari kiwi syslog :



Gambar 4.6 Hasil Alerting Kiwi Syslog



Gambar 4.7 Hasil Capture Wireshark

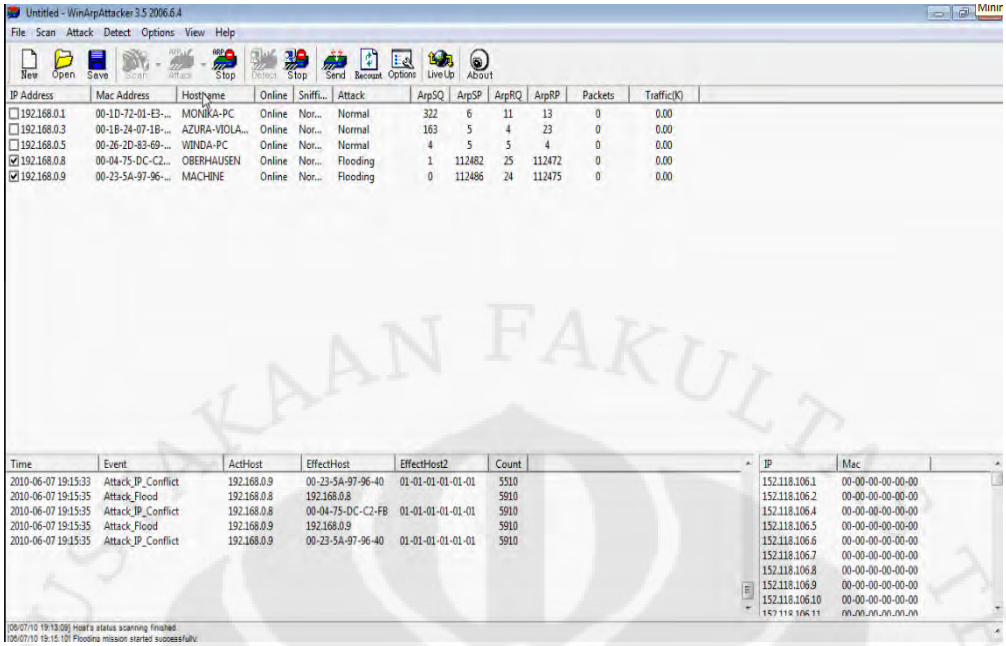
Pada hasil Gambar 4.6 tersebut menunjukkan bahwa Kiwi Syslog mampu mendeteksi adanya *alert* pada jaringan tersebut yang menunjukkan suatu *Potentiality Bad Traffic* pada paket ICMP yang berada pada *Priority* 2, ini berarti bahwa port scan merupakan serangan yang cukup berbahaya bagi jaringan.

Pada hasil Gambar 4.7 menunjukan port yang aktif atau terbuka di jaringan tersebut berada pada protocol TCP yang dapat *dicapture* melalui wireshark.

4.2.1.3 Flooding

Flooding merupakan suatu serangan dengan cara membanjiri *request* atau data ke jaringan dengan tujuan agar jaringan tersebut kebanjiran *request* yang sangat banyak yang mengakibatkan jaringan menjadi lemot atau lambat dan tidak mampu untuk melayani *request* yang sangat banyak tersebut. Flooding ini juga termasuk ke dalam serangan DOS. DOS ini akan menyerang dengan cara mencegah seorang pengguna untuk melakukan akses terhadap sistem atau jaringan yang dituju serta DOS bekerja dengan cara menghabiskan *resource* yang dimiliki oleh komputer tersebut sampai akhirnya komputer tersebut tidak dapat menjalankan fungsinya dengan benar yang secara tidak langsung mencegah pengguna lain untuk memperoleh akses layanan dari komputer yang diserang tersebut.

Pengujian *flooding* ini dilakukan dengan menggunakan *tools* bernama *WinArpAttacker. WinArpAttacker* merupakan salah satu *networking tool* yang memiliki banyak fungsi yang dapat digunakan untuk *sniffing, spoofing* dan *attacking* pada jaringan serta dapat menyerang komputer target dengan *ip conflict, flood* atau melakukan *local ddos*. *WinArpAttacker* sangat berguna bagi Admin jaringan untuk mengetes kelemahan dan cara memperbaiki jaringan yang dikelolanya. Gambar 4.8 adalah tampilan dari *WinArpAttacker :*

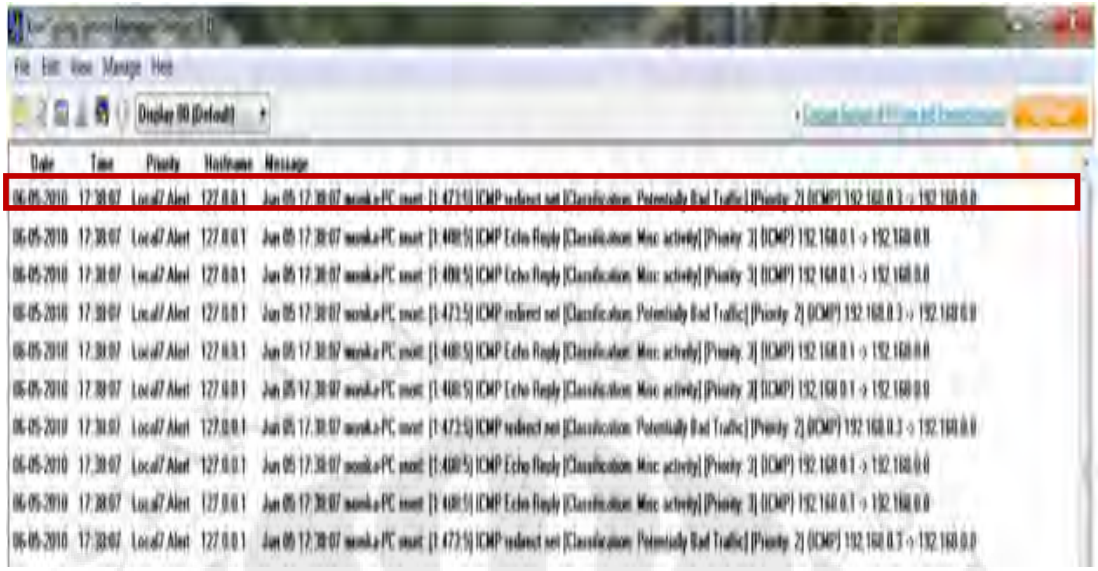


Gambar 4.8 Tampilan WinArpAttacker

Jika program ini dijalankan, maka program akan mengirimkan paket *broadcast* keseluruh alamat IP yang diinginkan.

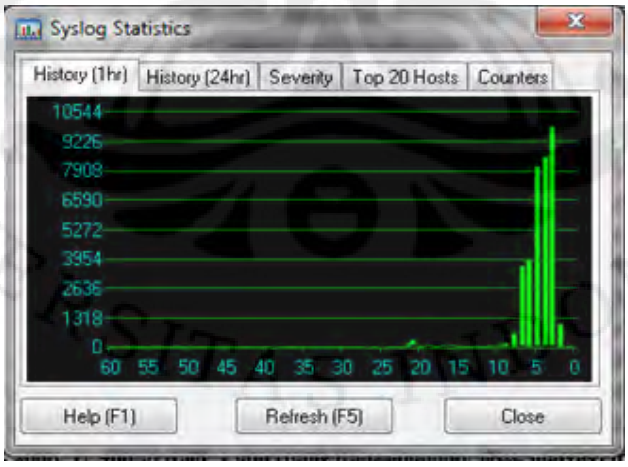
1) Flooding Pada 1 *Client*

Pada skenario ini akan dilakukan serangan *flooding* dengan menggunakan *tools WinArpAttacker 3.50.* Dengan menggunakan *tools* ini maka serangan dapat dimulai dengan membanjiri komputer target yaitu pada satu *client* saja yang mempunyai alamat IP 192.168.0.8. Pemakaian *tools* ini nantinya mampu men-*generate* paket yang menimbulkan IP *Conflict* serta *Request Times Out* pada target. Parameter yang dihitung pertama adalah *Functionality Test*. Gambar 4.9 adalah hasil percobaan yang telah dilakukan oleh Kiwi syslog :



Gambar 4.9 Tampilan Alerting 1 Client Pada Kiwi Syslog

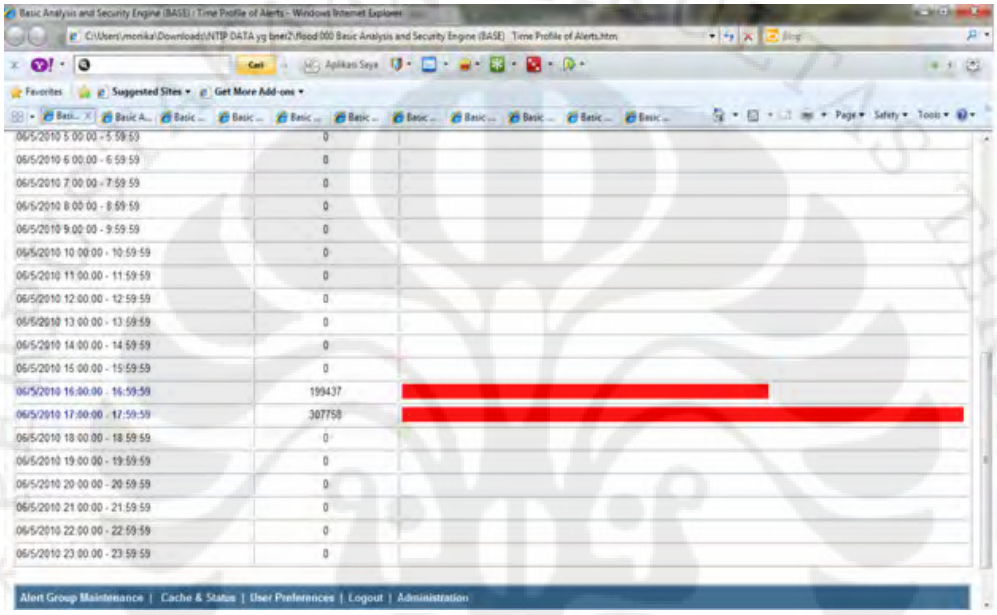
Pada Gambar 4.9 menjelaskan bahwa sistem IDS berhasil mendeteksi adanya serangan pada jaringan yaitu adanya *Potentially Bad Traffic* pada prioritas 2 yang berada pada paket ICMP dari alamat IP 192.168.0.3 menuju ke 192.168.0.8. Gambar 4.10 menjelaskan mengenai grafik yang didapat pada saat terjadi serangan flooding :



Gambar 4.10 Grafik Serangan 1 Client Pada Kiwi Syslog

Grafik ini menjelaskan tentang pergerakan serangan *flooding* di 1 *clien*t terhadap jumlah traffic serangan dan waktu selama 60 menit.

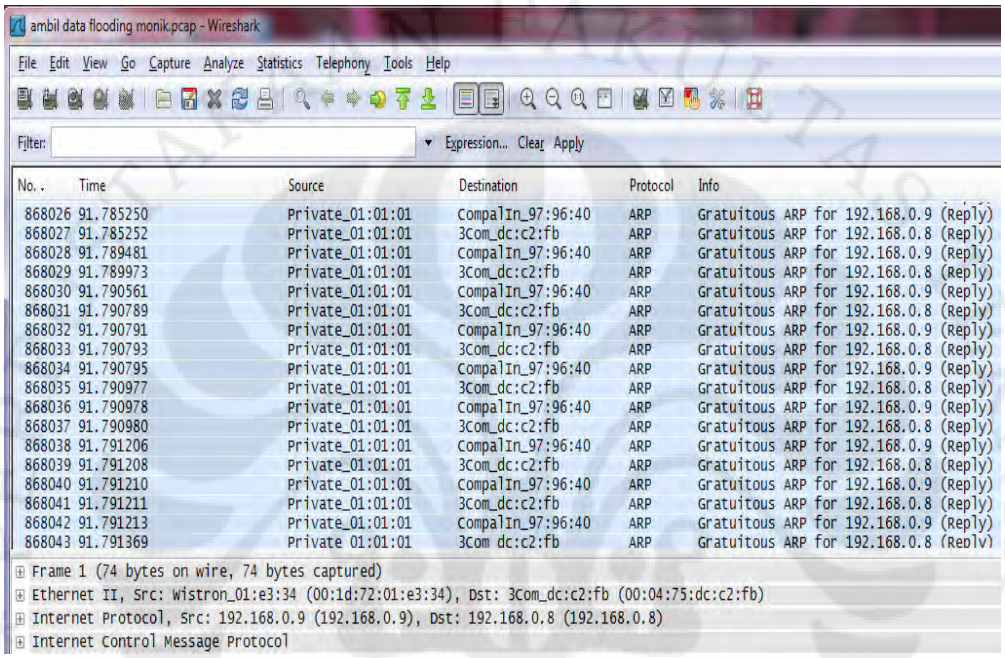
Puncak grafik terhadap traffic serangan tersebut berada pada nilai antara 10544 dan 9226. Grafik tersebut dari 0 menit – 60 menit pergerakkannya semakin turun, hal ini dikarenakan hanya ada 1 client yang memberikan respon terhadap serangan tersebut sehingga pergerakkannya semakin menurun. Selama 60 menit atau 1 jam ini, maka akan diperlihatkan besarnya serangan yang ditangkap oleh BASE. Gambar 4.11 adalah hasil yang ter-*capture* oleh BASE :



Gambar 4.11 Tampilan Jumlah Serangan di BASE

Hasil *capture* serangan dari BASE ini menunjukkan jumlah atau banyaknya serangan yang masuk pada jaringan. Apabila melihat waktu terjadinya serangan pada Kiwi Syslog yaitu pada pukul 17:38:07, maka dapat melihat banyaknya serangan yang terjadi pada pukul 17.00.00 – 17.59.59 yaitu sebesar 307.758 alert. BASE ini memang menyimpan serangan yang terjadi di dalam jaringan yang banyaknya jumlah serangan dapat dilihat berdasarkan waktu selama satu jam penuh. Sehingga apabila ingin melihat banyaknya serangan yang terjadi maka hanya tinggal mencari waktu yang diinginkan. Penggunaan *wireshark* juga dapat memperlihatkan adanya serangan di dalam jaringan dengan melihat terjadinya *Gratuitous ARP*. *Gratuitous ARP* ini yaitu paket-paket yang dikirim adalah paket-paket pengecekan nomor IP yang akan menyebabkan adanya *IP Conflik* yang menyebabkan terjadinya *Request Time Out.* Serangan atau

pembanjiran yang dilakukan oleh *WinArpAttacker* akan mengirimkan atau membanjiri paket ARP ke tujuan. Paket yang dikirim yaitu *Gratuitous ARP* yang menyebabkan terjadinya *IP Conflict* karena paket tujuan tidak mengetahui alamat *IP* yang benar untuk mengirimkan ping ICMP. Gambar 4.12 adalah hasil *capture wireshark* :

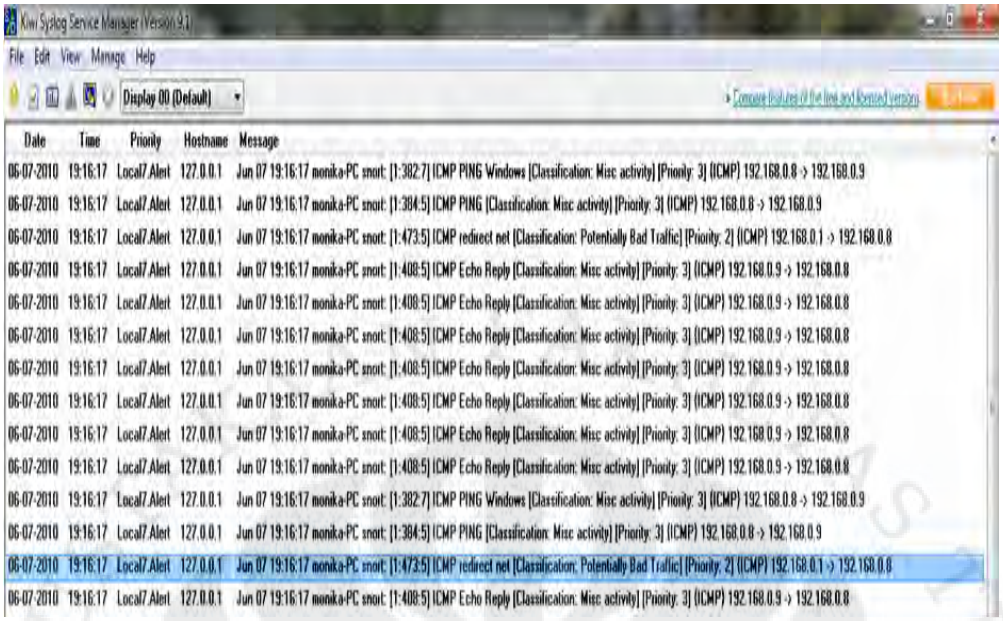


Gambar 4.12 Hasil Capture Wireshark

Banyaknya serangan yang di dapat maka nanti dapat dianalisa dengan membedakan nilai antara serangan terhadap 1 *client, 2 client,* dan 3 *client.*

2) Flooding pada 2 Client

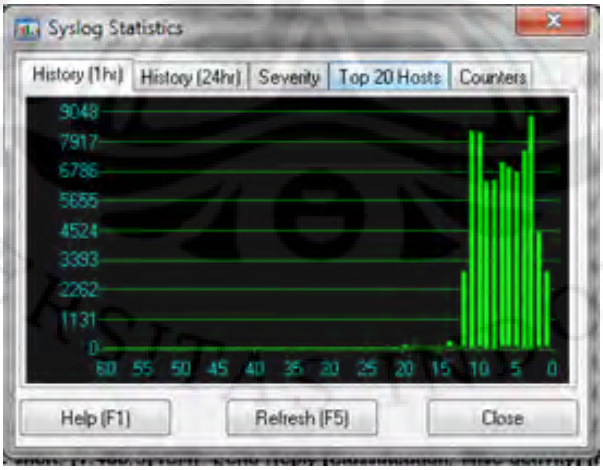
Pada skenario ini akan dilakukan serangan *flooding 2 client* sama halnya seperti pada 1 *client yaitu* dengan menggunakan *tools WinArpAttacker 3.50.* Skenario untuk *flooding* pada 2 *client* yaitu komputer penyerang akan menyerang target 2 *client* sekaligus. Alamat IP yang digunakan untuk diserang yaitu 192.168.0.1 dan 192.168.0.5. Nilai yang sudah di dapat, langsung dibandingkan dengan 1 *client* untuk membuktikan keandalan dari sistem IDS ini. Gambar 4.13 adalah hasil percobaan yang telah dilakukan oleh Kiwi Syslog :



```
Date        Time      Priority      Hostname   Message
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:382:7] ICMP PING Windows [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.8 -> 192.168.0.9
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:384:5] ICMP PING [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.8 -> 192.168.0.9
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:473:5] ICMP redirect net [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 192.168.0.1 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.9 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.9 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.9 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.9 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.9 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.9 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.9 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:382:7] ICMP PING Windows [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.8 -> 192.168.0.9
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:384:5] ICMP PING [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.8 -> 192.168.0.9
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:473:5] ICMP redirect net [Classification: Potentially Bad Traffic] [Priority: 2] {ICMP} 192.168.0.1 -> 192.168.0.8
06-07-2010  19:16:17  Local7.Alert  127.0.0.1  Jun 07 19:16:17 monika-PC snort: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.0.9 -> 192.168.0.8
```

Gambar 4.13 Tampilan Alerting 2 Client Pada Kiwi Syslog

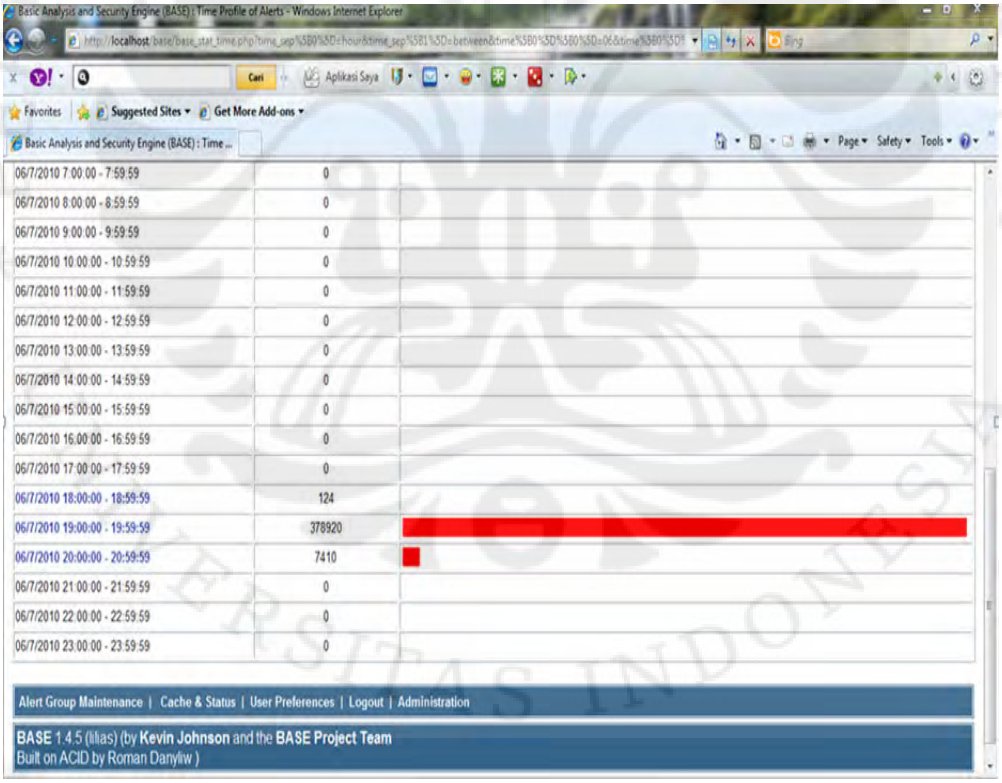
Pada Gambar 4.13 menjelaskan bahwa sistem IDS berhasil mendeteksi adanya serangan pada jaringan yaitu adanya *Potentially Bad Traffic* pada prioritas 2 yang berada pada paket ICMP. Gambar 4.14 akan dijelaskan mengenai grafik yang didapat pada saat terjadi serangan flooding :



Gambar 4.14 Grafik Serangan 2 Client Pada Kiwi Syslog

Grafik ini menjelaskan tentang pergerakan serangan *flooding* di 2 *clien*t terhadap jumlah traffic serangan dan waktu selama 60 menit.

Puncak grafik terhadap traffic serangan tersebut berada pada nilai antara 9048 dan 7917 . Grafik tersebut dari 0 menit – 60 menit pergerakkannya yaitu fluktuatif. Apabila membandingkan grafik 1 *client* maka tinggi *alertnya* semakin turun hal ini disebabkan karena serangan ini terbagi menjadi dua, tetapi apabila melihat banyaknya grafik terhadap 2 *client* maka grafik akan semakin banyak, hal ini disebabkan karena masing-masing *client* memberikan respon terhadap serangan tersebut, sehingga grafiknya semakin banyak apabila dibandingkan dengan grafik 1 *client.* Selama 60 menit atau 1 jam ini, maka akan diperlihatkan besarnya serangan yang ditangkap oleh BASE. Gambar 4.15 adalah hasil yang telah di *capture* oleh BASE :

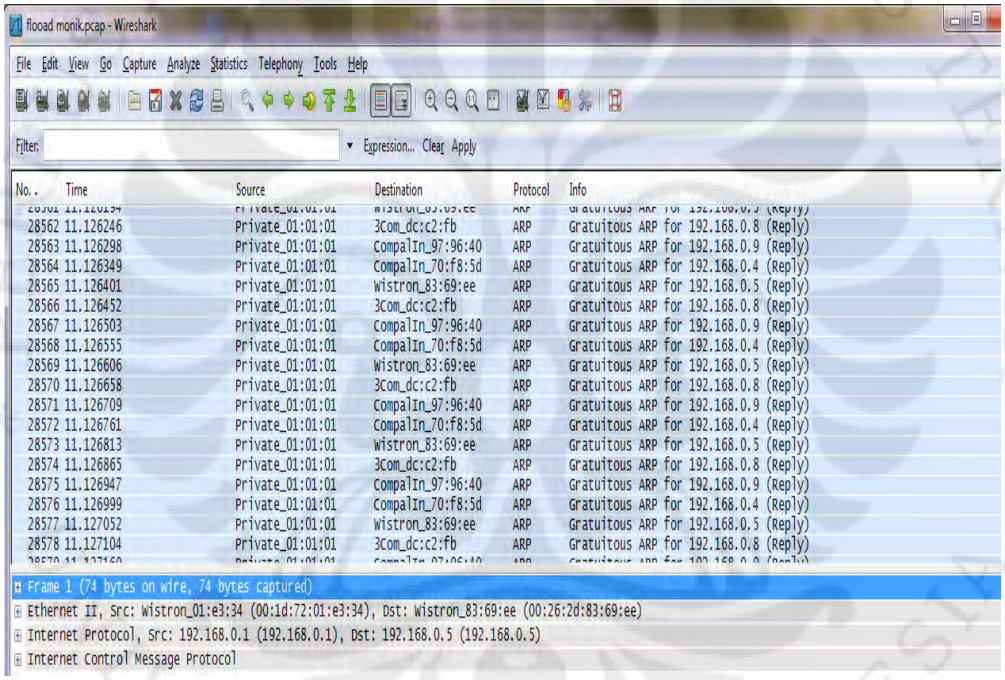


Gambar 4.15 Jumlah Serangan di BASE pada 2 Client

Hasil *capture* serangan dari BASE ini menunjukkan jumlah atau banyaknya serangan yang masuk pada jaringan. Apabila melihat waktu terjadinya

serangan pada Kiwi Syslog yaitu pada pukul 19:16:17, maka dapat melihat banyaknya serangan yang terjadi pada pukul 19.00.00 – 19.59.59 yaitu sebesar 378.920 alert. Apabila melihat jumlah pada 1 *client* yaitu lebih rendah atau sedikit jika dibandingkan dengan 2 *client,* hal ini dikarenakan masing-masing *client* memberikan respon terhadap serangan tersebut sehingga 1 *client* lebih rendah dibandingkan jumlah serangan pada 2 *client.*

Penggunaan *wireshark* juga dapat memperlihatkan adanya serangan di dalam jaringan dengan melihat terjadinya *Gratuitous ARP*. *Gratuitous ARP* ini sudah dijelaskan pada 1*client*. Berikut ini adalah gambar dari *capture wireshark* :

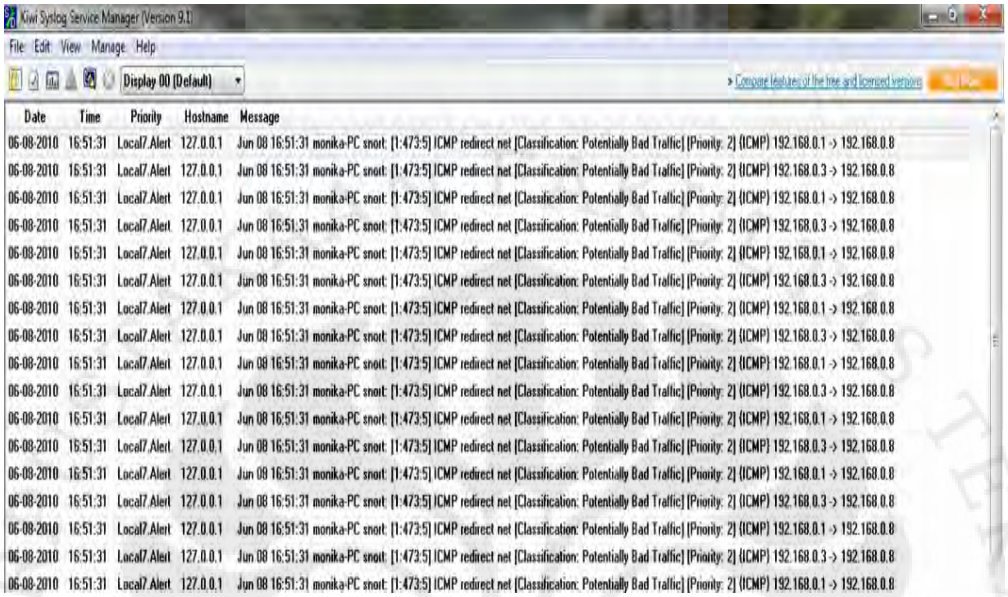


Gambar 4.16 Hasil Capture Wireshark 2 Client

3) Flooding 3 Client

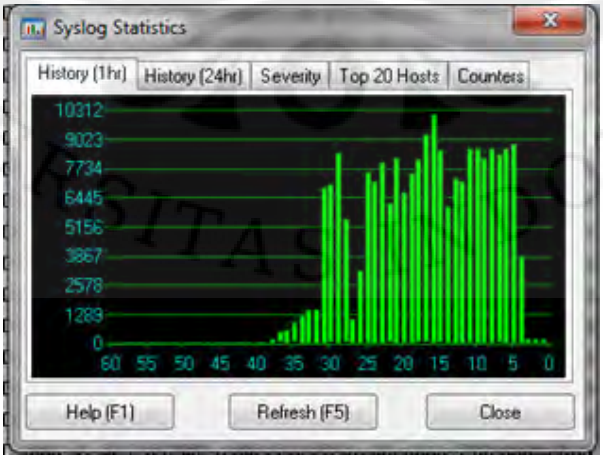
Pada skenario ini akan dilakukan serangan *flooding 3 client* sama halnya seperti pada 1 dan 2 *client yaitu* dengan menggunakan *tools WinArpAttacker 3.50.* Skenario untuk *flooding* pada 3 *client* yaitu PC penyerang akan menyerang target 3 *client* sekaligus. Alamat IP yang digunakan untuk diserang yaitu 192.168.0.8, 192.168.0.9, dan 192.168.0.4. Sehingga apabila sudah mendapatkan nilainya, maka dapat langsung dibandingkan dengan 1 *client* , 2 *client,* dan 3 *client* untuk

membuktikan keandalan dari sistem IDS ini. Berikut ini hasil percobaan yang telah dilakukan :



Gambar 4.17 Tampilan Alerting 3 Client Pada Kiwi Syslog

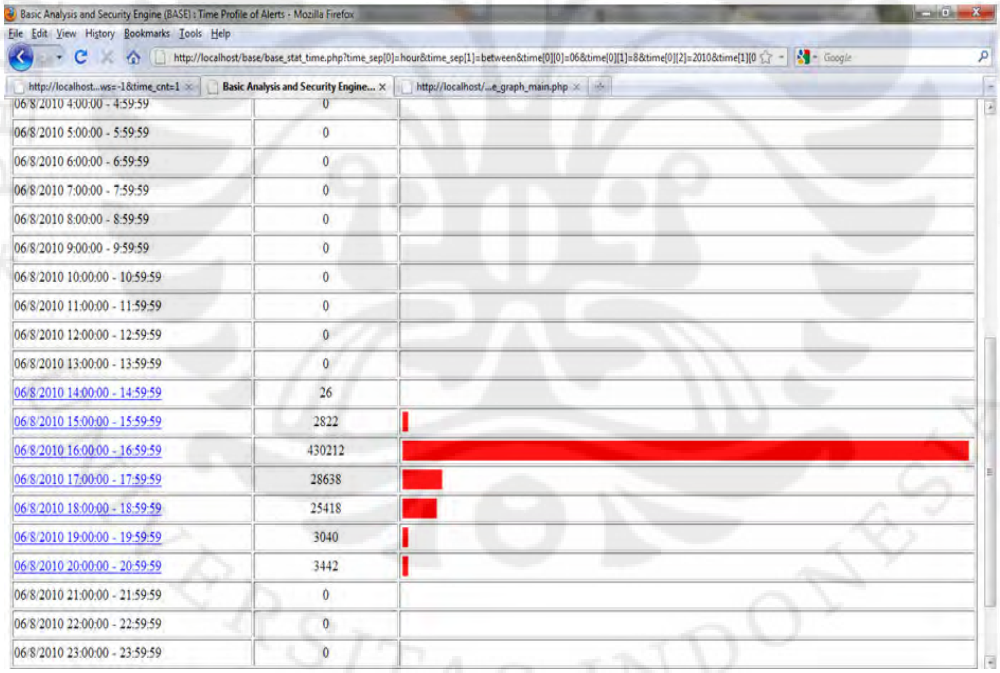
Pada Gambar 4.17 menjelaskan bahwa sistem IDS berhasil mendeteksi adanya serangan pada jaringan yaitu adanya *Potentially Bad Traffic* pada prioritas 2 yang berada pada paket ICMP. Berikut ini akan dijelaskan mengenai grafik yang didapat pada saat terjadi serangan flooding :



Gambar 4.18 Grafik Serangan 3 Client Pada Kiwi Syslog

Grafik ini menjelaskan tentang pergerakan serangan *flooding* di 3 *clien*t terhadap jumlah traffic serangan dan waktu selama 60 menit.

Puncak grafik terhadap traffic serangan tersebut berada pada nilai antara 10312 dan 9023. Grafik tersebut dari 0 menit – 60 menit pergerakkannya yaitu fluktuatif. Apabila membandingkan grafik 1 *client* maka tinggi *alertnya* semakin turun hal ini disebabkan karena serangan ini terbagi menjadi tiga, tetapi apabila melihat banyaknya grafik terhadap 3 *client* maka grafik akan semakin banyak, hal ini disebabkan karena masing-masing *client* memberikan respon terhadap serangan tersebut, sehingga grafiknya semakin banyak apabila dibandingkan dengan grafik 1dan 2 *client.* Selama 60 menit atau 1 jam ini, maka akan diperlihatkan besarnya serangan yang ditangkap oleh BASE. Gambar 4.19 adalah hasil yang telah di *capture* oleh BASE :

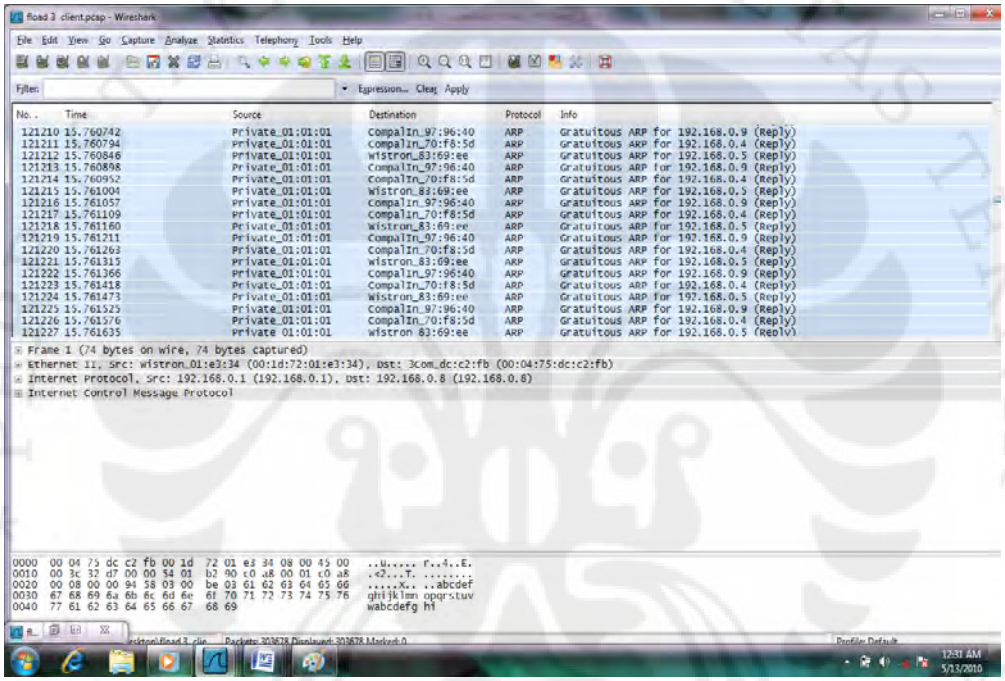


Gambar 4.19 Jumlah Serangan di BASE pada 3 Client

Hasil *capture* serangan dari BASE ini menunjukkan jumlah atau banyaknya serangan yang masuk pada jaringan. Apabila melihat waktu terjadinya serangan pada Kiwi Syslog yaitu pada pukul 16:51:31, maka dapat melihat banyaknya serangan yang terjadi pada pukul 16.00.00 – 16.59.59 yaitu sebesar 430.212 alert. Apabila melihat jumlah pada 1 dan 2 *client* yaitu lebih rendah atau

sedikit jika dibandingkan dengan 3 *client,* hal ini dikarenakan masing-masing *client* memberikan respon terhadap serangan tersebut sehingga 1 dan 2 *client* lebih rendah dibandingkan jumlah serangan pada 3 *client.*

Penggunaan *wireshark* juga dapat memperlihatkan adanya serangan di dalam jaringan dengan melihat terjadinya *Gratuitous ARP*. *Gratuitous ARP* ini sudah dijelaskan pada 1*client*. Gambar 4.20 adalah hasil yang telah di *capture* oleh *wireshark* :



Gambar 4.20 Hasil Capture Wireshark 3 Client

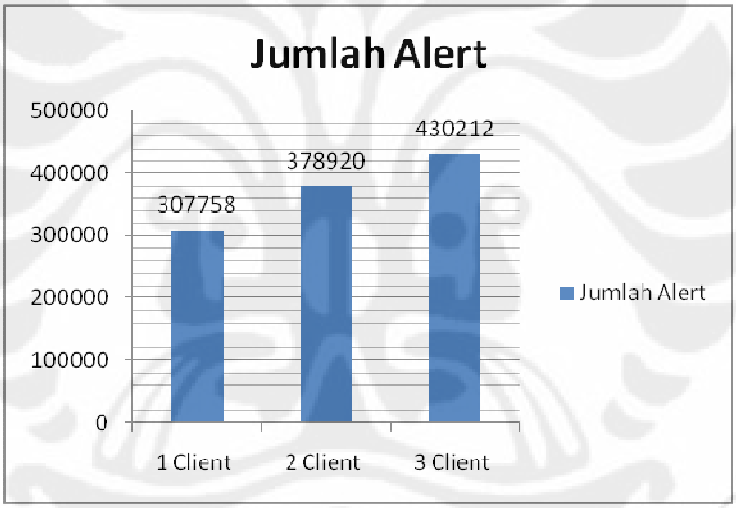
Dari data 1, 2, dan 3 *client* yang telah didapat maka dapat disimpulkan bahwa setiap penambahan jumlah *client* maka akan semakin besar jumlah alert yang didapat. Hal ini dikarenakan setiap *client* memberikan respon terhadap serangan tersebut, sehingga semakin banyak *client* semakin besar pula jumlah *alert* yang didapatkan. Berikut ini untuk mengetahui lebih jelas dapat melihatnya melalui tabel dan grafik berikut ini :

Tabel :

Tabel 4.1 Tabel Jumlah Alert Terhadap Client

Jumlah Client	Jumlah Alert
1 Client	307758
2 Client	378920
3 Client	430212

Grafik :



Gambar 4.21 Grafik Jumlah Alert Berdasarkan Client

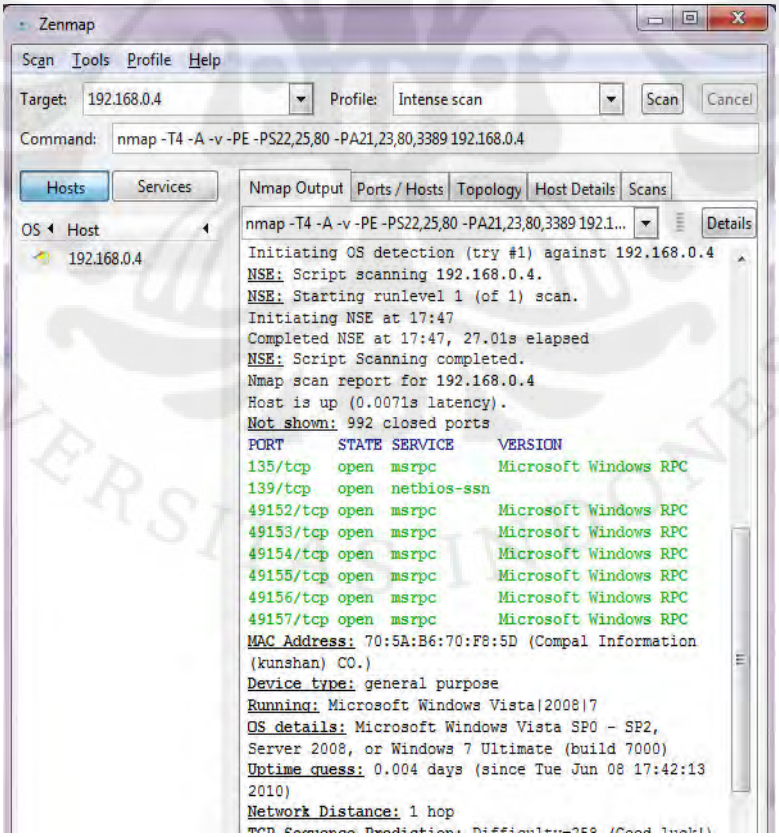
Pada gambar grafik dan tabel di atas maka dapat menunjukan perbandingan antara jumlah *alert* terhadap serangan yang berasal dari 1, 2, dan 3 *client.* Untuk 1 *client* jumlah yang di dapat yaitu 307.758 alert, untuk 2 *client* 378.920 alert, sedangkan untuk 3 *client* yaitu 430.212 alert. Hal ini menunjukkan adanya penaikan terhadap serangan tersebut. Hal ini tentu saja dikarenakan banyaknya jumlah *client* akan mempengaruhi jumlah *alert* yang dihasilkan. Semakin banyak jumlah *client* maka akan semakin banyak jumlah *alert* yang dihasilkan.

4.2.2 Response Time

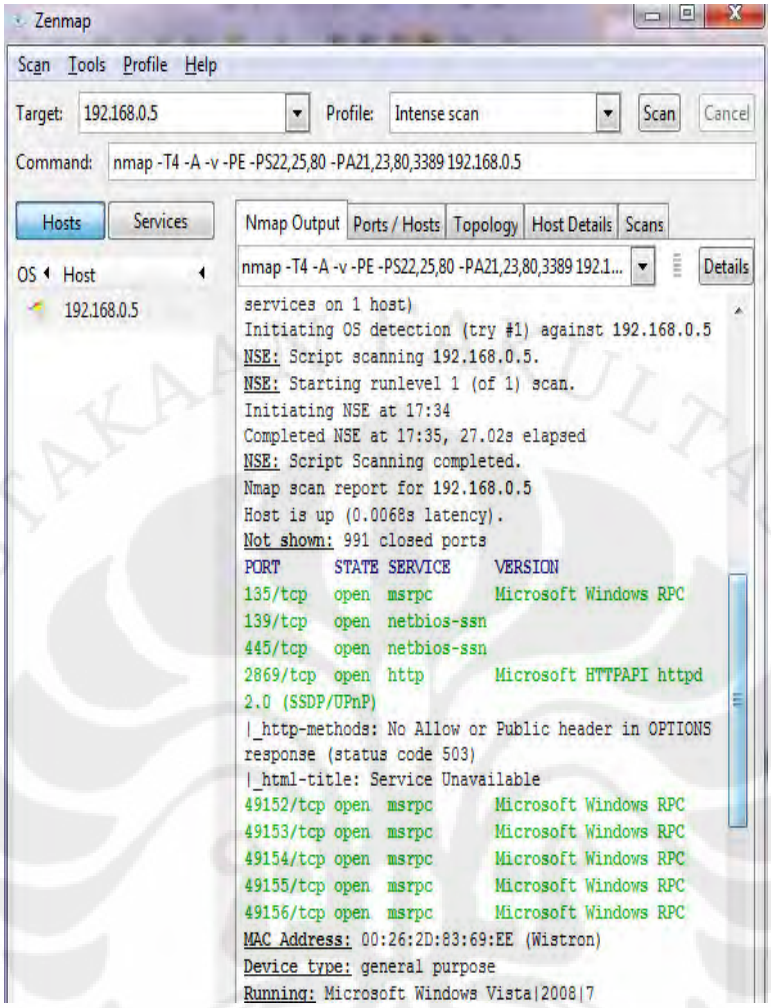
4.2.2.1 Port Scan

Untuk melihat tingkat kehandalan dari suatu sistem IDS maka perlu melihat dari beberapa parameter. Salah satu parameter yang dianggap penting yaitu *response time*. *Response time* adalah waktu yang dibutuhkan untuk merespon sebuah serangan, dimana pada percobaan *response time* dilakukan pada saat serangan dimulai sampai pada saat sistem pertama kali memberi respon. Dibawah ini adalah *response time* dari *port scan* yang dihasilkan dari sistem operasi terhadap waktu dalam merespon serangan.

Pengujian ini dilakukan dengan membandingkan waktu yang dibutuhkan terhadap sistem operasi. Gambar 4.22 akan memperliatkan waktu yang dibutuhkan untuk masing-masing operasi sistem :



Gambar 4.22 Tampilan Waktu untuk Port Scan di Windows Vista



Gambar 4.23 Tampilan Waktu untuk Port Scan di Windows 7

Pada Gambar tersebut dijelaskan bahwa *port scan* yang dilakukan pada sistem operasi Windows Vista yaitu menghasilkan waktu sebesar 27.01s, sedangkan waktu yang dibutuhkan oleh Windows 7 yaitu sebesar 27.02s. Sehingga apabila diperhatikan waktu yang dibutuhkan untuk merespon sebuah serangan yaitu hampir sama. Hal ini disebabkan karena pada saat pengambilan data melakukan percobaan dengan menggunakan cara serial (tidak dijalankan secara bersamaan), sehingga waktu yang dibutuhkan untuk merespon serangan akan tetap sama atau tidak berbeda jauh. Alasan penggunaan sistem operasi Windows 7 yaitu sampai saat ini masalah keamanan, target utamanya masih

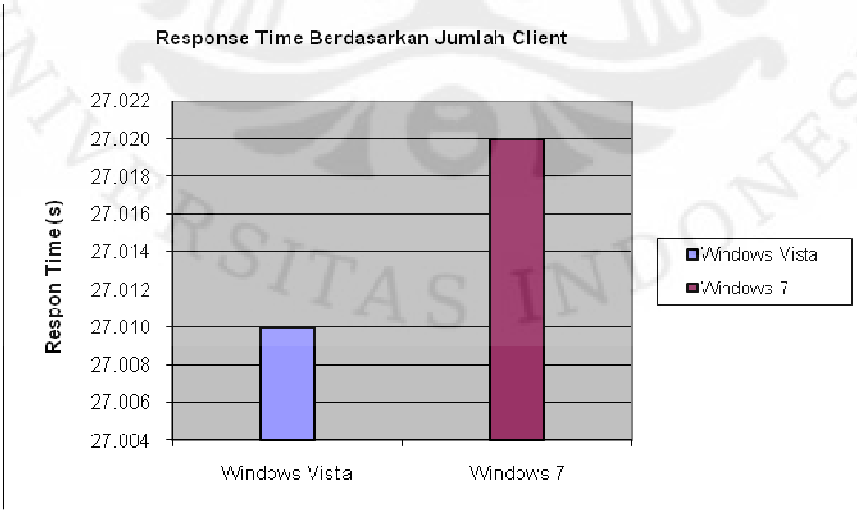
kepada sistem operasi Windows, maka pada skripsi ini dibangun sebuah sistem IDS yang berbasis pada sistem operasi Windows 7, selain itu Windows 7 merupakan satu-satunya sistem hasil perbandingann tes keamanan yang mempunyai *Future Security Center*, dimana *future* ini menyediakan suatu sistem yang dapat memberikan keamanan dan control keamanan apabila *service* tidak dapat berjalan dengan baik, maka sistem akan memberitahukan para pemakainya dan Microsoft akan memberikan solusi secara Online. Berikut ini adalah grafik yang menunjukkan tidak adanya perbedaan waktu yang dilakukan terhadap dua buah operasi sistem :

Tabel :

Table 4.2 Respon Time Terhadap OS

OS	Respon Time
Vista	27.01 s
Seven	27.02 s

Grafik :

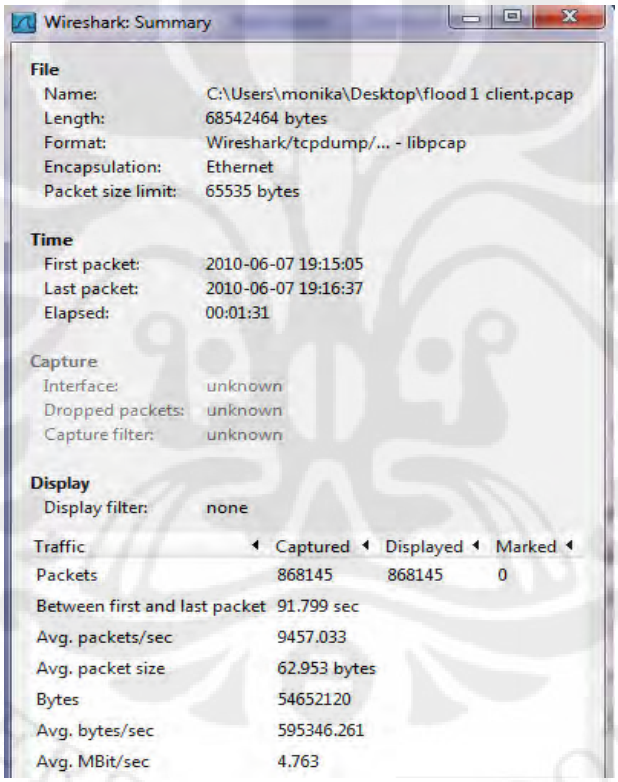


Gambar 4.24 Grafik respon Time Terhadap OS

4.2.2.2 Flooding

Salah satu parameter kehandalan dari suatu sistem IDS juga harus dapat melayani beberapa *client* secara sekaligus. Sehingga untuk menguji ini menggunakan 3 *client* yang melakukan serangan secara bersama-sama yang nantinya dapat melihat *respon time* dari sistem tersebut.

Perhitungan *respon time* ini di dapat dari jumlah paket yang tertangkap oleh *wireshark* terhadap waktu selama *wireshark* menangkap paket tersebut. Gambar 4.25 adalah *capture* dari jumlah paket dan waktu pada 1 *client* yang ada pada *wireshark* :



Gambar 4.25 Jumlah Paket dan Waktu pada 1 Client

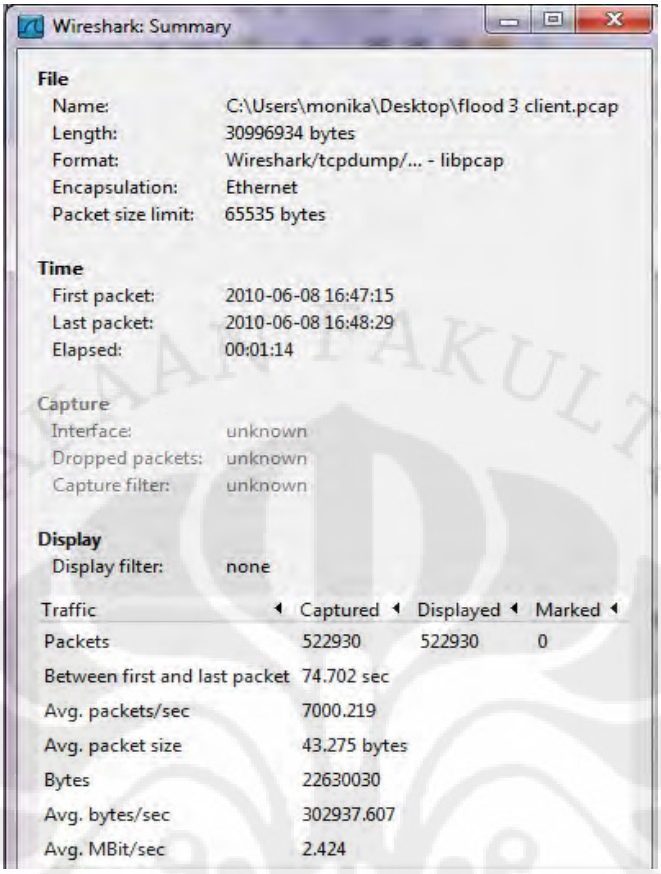
Perhitungan *response time* ini didapat dengan menghitung jumlah paket yang masuk ke dalam *wireshark* terhadap jumlah waktu antara paket pertama yang masuk sampai dengan paket terakhir yang masuk. Sehingga untuk perhitungan *response time* 1 *client* maka di dapat hasil *response time* sebesar 0,000105741 s per paket.

Gambar 4.26 adalah *capture* dari jumlah paket dan waktu pada 2 *client* yang ada pada *wireshark* :

Gambar 4.26 Jumlah Paket dan Waktu pada 2 Client

Perhitungan *response time* ini didapat dengan menghitung jumlah paket yang masuk ke dalam *wireshark* terhadap jumlah waktu antara paket pertama yang masuk sampai dengan paket terakhir yang masuk. Sehingga untuk perhitungan *response time* 2 *client* maka di dapat hasil *response time* sebesar 0,000127213 s per paket.

Gambar 4.27 adalah *capture* dari jumlah paket dan waktu pada 3 *client* yang ada pada *wireshark* :



Gambar 4.27 Jumlah Paket dan Waktu pada 3 Client

Perhitungan *response time* ini didapat dengan menghitung jumlah paket yang masuk ke dalam *wireshark* terhadap jumlah waktu antara paket pertama yang masuk sampai dengan paket terakhir yang masuk. Sehingga untuk perhitungan *response time* 3 *client* maka di dapat hasil *response time* sebesar 0,000142852 s per paket.

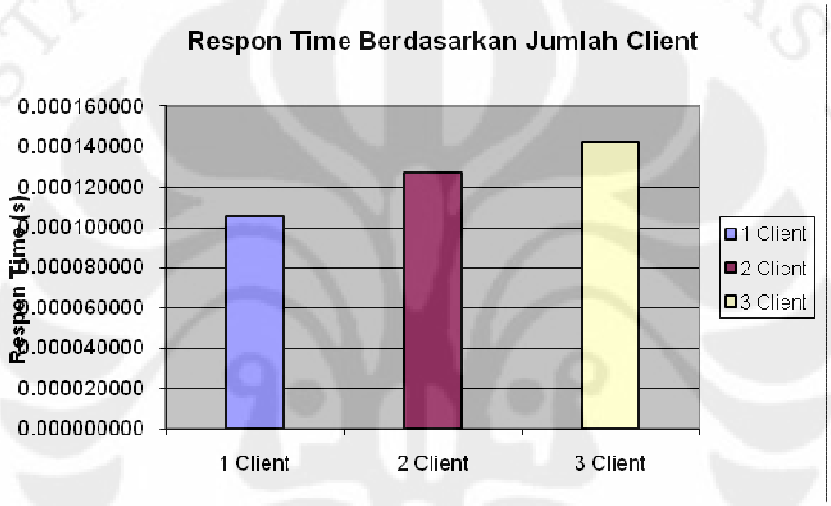
Berikut ini adalah grafik yang menunjukkan hasil *response time* terhadap 1, 2, dan 3 *client.*

Tabel :

Tabel 4.3 Respon Time Terhadap Jumlah Client

Jumlah Client	Respon Time
Client 1	0.000105741 s
Client 2	0.000127213 s
Client 3	0.000142852 s

Grafik :



Gambar 4.28 Grafik Respon Time terhadap Jumlah Client

Dari hasil pengujian dapat dilihat bahwa dengan bertambahnya jumlah *client* akan berpengaruh terhadap performa sistem IDS ini. Hal ini ditunjukkan dengan semakin lamanya *response time* yang dihasilkan oleh sistem IDS tersebut, sehingga ini semua disebabkan karena serangan paket *flooding* yang dikirimkan yaitu paket *broadcast* dan *flooding* bekerja dengan cara membanjiri *request* atau data ke jaringan dengan tujuan agar jaringan tersebut kebanjiran *request* yang sangat banyak. Apabila 2 atau 3 *client* mengirimkan paket *broadcast* secara bersamaan maka jaringan menjadi semakin penuh sehingga menyebabkan kemampuan IDS dalam merespon serangan akan terganggu.

Semakin tinggi angka *response time* atau semakin lambatnya *response time* ini disebabkan karena adanya proses pengiriman paket dalam jumlah yang besar yang berasal dari *client.*

Sehingga dengan bertambahnya *client* yang melakukan serangan secara bersamaan dapat menyebabkan *response time* yang dimiliki sistem IDS semakin lambat.

BAB IV

KESIMPULAN

1. Pada *functionality test,* IDS mampu mendeteksi adanya serangan baik yang berupa *IP Scan, Port Scan,* maupun *Flooding.*
2. Berdasarkan percobaan yang telah dilakukan terjadi kenaikan alert sebesar 23,12 % dari 1 client ke 2 client, 13,54 % dari 2 client ke 3 client, serta 39,79 % dari 1 client ke 3 client. Hal ini menunjukkan adanya penaikan terhadap serangan tersebut. Sehingga dengan bertambahnya jumlah *client* maka semakin banyak pula jumlah *alert* yang dihasilkan.
3. Berdasarkan percobaan yang telah dilakukan terjadi kenaikan response time sebesar 20.31 % dari 1 client ke 2 client, 12,29 % dari 2 client ke 3 client, serta 35,10 % dari 1 client ke 3 client. Hal ini menunjukkan semakin tinggi angka *response time* atau semakin lambatnya *response time* ini disebabkan karena adanya proses pengiriman paket dalam jumlah yang besar yang berasal dari *client,* sehingga dengan bertambahnya *client* yang melakukan serangan secara bersamaan dapat menyebabkan *response time* yang dimiliki sistem IDS semakin lambat.

DAFTAR REFERENSI

[1] Ariyus, Dony, "Intrusion Detection System", 2007

[2] Ariyus, Dony. Istiyanto, Jazi Eko, "Membangun Intrusion Detection System Pada Windows 2003 Server". 2007

[3] Buku8's blog, http://buku8.wordpress.com/2010/04/14/beberapa-jenis-serangan-yang-menyerang-jaringan/, di akses pada tanggal 5 Mei 2010

[4] Gilang, Jaringan Komputer, http://gilang1188.blogspot.com/2009/10/jaringan-komputer.html, di akes 3 Mei 2010

[5] Rehman, Rafeeq Ur. 2003. *Intrusion Detection Systems with Snort*. Prentice HALL. New Jersey.

[6] Ri2M, Network and www.security, http://ftp.labkom.bl.ac.id, di akses pada tanggal 3 Mei 2010

[7] Chris Vespermann, 2003 .Snort, MySQL, Apache, and BASE for Gentoo Linux.