IMPLEMENTASI INTRUSION DETECTION SYSTEM MENGGUNAKAN SNORT, BARNYARD2 DAN BASE PADA SISTEM OPERASI LINUX

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Kelulusan Pada
Program Studi Strata Satu Sistem Komputer di Jurusan Teknik Komputer

Oleh
Rizki Triandini
10210164

Pembimbing
Dr. Yeffry Handoko Putra, M.T

JURUSAN TEKNIK KOMPUTER
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
BANDUNG

2016

LEMBAR PENGESAHAN

IMPLEMENTASI INTRUSION DETECTION SYSTEM MENGGUNAKAN SNORT, BARNYARD2 DAN BASE PADA SISTEM OPERASI LINUX

Oleh

Rizki Triandini
10210164

Telah disetujui dan disahkan di Bandung sebagai Tugas Akhir pada tanggal:

Bandung, 25 Februari 2016

Menyetujui,

Pembimbing,

Dr. Yeffry Handoko Putra, M.T
NIP.4127.70.05.001

Mengetahui,

Dekan Fakultas Teknik dan Ilmu Komputer	Ketua Jurusan Teknik Komputer
Prof. Dr. H. Denny Kurniadie, Ir.M.Sc	Dr. Wendi Zarman, M.Si
NIP.4127.70.015	NIP.4127.70.05.010

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Rizki Triandini

NIM : 10210164

Menyatakan bahwa Tugas Akhir ini adalah hasil karya sendiri dan bukan merupakan duplikasi sebagian atau seluruhnya dari hasil karya orang lain yang pernah dipublikasikan atau yang sudah pernah dipakai untuk mendapatkan gelar di Universitas lain, kecuali pada bagian dimana sumber informasi dicantumkan dengan cara referensi yang semestinya.

Pernyataan ini dibuat agar dengan sebenar-benarnya secara sadar dan bertanggung jawab dan saya bersedia menerima sanksi berupa pembatalan Tugas Akhir saya apabila terbukti melakukan duplikasi terhadap Tugas Akhir yang sudah ada.

Bandung, 17 Februari 2016

Rizki Triandini

ABSTRAK

Intrusion Detection System (IDS) dapat didefinisikan sebagai kegiatan yang bersifat *anomaly, incorrect, inappropriate* yang terjadi di jaringan atau *host*. Dan IDS sendiri adalah sistem keamanan yang bekerja bersama *Firewall* untuk mengatasi *Intrusion*. IDS mampu mendeteksi penyusup dan memberikan respon secara *real time*. Terdapat dua teknik yang digunakan dalam IDS yaitu, NIDS (*Network Based Intrusion Detection System*) dan HIDS (*Host Based Intrusion Detection System*). Pada percobaan kali ini IDS dibagun menggunakan perangkat lunak Snort. Snort merupakan *Open Source Intrusion Detection System* (IDS) yang digunakan untuk pemantauan dan pencegahan terhadap gangguan pada jaringan komputer. Agar mempermudah administrator dalam melihat dan membaca hasil log dari setiap paket data yang masuk atau keluar maka menggunakan Basic Analysis and Security Engine (BASE). Pada percobaan ini PC Server menggunakan sistem operasi Linux Ubuntu 14.04 LTS. Pengujian dilakukan pada Local Area Network dengan topologi Star. Dimana hasil gangguan yang disebabkan DoS dan Port Scanner dapat dikenali oleh Snort IDS dan menampilkan log secara lengkap, baik dari waktu, tanggal kejadian dan sumber IP *Address* dari penggangu.

Kata kunci: Kemanan Jaringan, IDS, Snort, BASE

ABSTRACT

Intrusion Detection System (IDS) is defined as an anomaly, incorrect, inappropriate activity take place in a net or a host. The IDS is a security system that work with Firewall to overcome intrusion. IDS is able to detect an intruder and to response at real time. There are two methods used in IDS, NIDS (Network Based Intrusion Detection System) and HIDS (Host Based Intrusion Detection System). In this research, IDS built using the divace Snort software. Snort is an Open source Intrusion Detection System (IDS) used for the monitoring and prevention of disorders on a computer network. In order to make it easier for the administrator to see and read the log results from each entering or exiting of data paket then use the Basic Analysis and Security Engine (BASE). In this research, the PC Server was operated under Linux Ubuntu 14.04 LTS system. The tests were conducted on Local Area Network with Star Topologi. The result of disruptions coused by DoS and Port Scanner identified by snort IDS and displays the complete log, both of time, date of the incident and the source IP address of the intruder.

Key words: Network Security, IDS, Snort, BASE

KATA PENGANTAR

Puji dan sukur bagi Alah SWT, atas rahmat dan karunianya yang senantia selalu menyertai umatnya, shalawat serta salam taklupa pula kita sampaikan kepada Nabi besar Muhammad SAW, yang telah membawa umatnya dari zaman jailiah ke zaman yang terang benerang seperti saat ini.

Adapun judul dari tugas akhir ini adalah **Implementasi Intrusion Detection System Menggunakan Snort, Barnyard2 dan BASE Pada Sistem Operasi Linux.** Tujuan dari pembuatan laporan ini adalah untuk memenuhi salah satu syarat untuk memperoleh gelar sarjana Strata-1 di jurusan Teknik Komputer, Universitas Komputer Indonesia.

Selama melaksanankan tugas akhir dan penyususan laporan tugas akhir ini, penulis mendapat banyak bantuan dari berbagai pihak, baik material maupun spritual, namun dengan kerendahan hati, penulis mengucapkan terimakasih kepada yang terhormat :

1. Kedua Orang Tua, kakak dan keluarga besar tercinta yang senantiasa tidak henti-hentinya berdoa dan memberikan semangat kepada penulis agar dapat menyelesaikan tugas akhir ini.
2. Bapak Prof. Dr H. Denny Kurniadie, Ir. M.Sc. selaku Dekan Fakultas Teknik dan dan Ilmu Komputer Universitas Komputer Indonesia.
3. Bapak Dr. Wendi Zarman, M.Si. selaku Ketua Jurusan Teknik Komputer Universitas Komputer Indonesia.
4. Bapak Dr. Yeffry Handoko Putra. M.T. selaku pembimbing penulis yang telah banyak memberikan arahan, saran, nasihat, motivasi dan bimbingan kepada penulis selama menyelesaikan tugas akhir ini.
5. Ibu Sri Supatmi, M.T. Selaku dosen wali yang memberikan motivasi penulis dalam penyelesaian tugas akhir ini.
6. Ibu Susmini Indriani L, M.T. selaku penangguang jawab leb komunikasi data yang telah mengizinkan penulis untuk menggunakan lab sebagai tempat pembuatan dan pengujian dari tugas akhir ini.

7. Bapak dan Ibu dosen jurusan Teknik Komputer Universitas Komputer Indonesia yang pernah mengajar dan memberikan ilmunya kepada penulis.
8. Teman-teman seperjuangan di kelas 10-TK4, Aulia Rahmat Sagala, Made Aldi Bunggi, Wahyu Adrian Kusuma, Sofyan Adriansyah, S.Kom, Giri Laya Bahari, S.Kom dan teman-teman yang lainnya.
9. Teman-teman dari Inter Club Indonesia Regional Bandung dan teman-teman dari ICI Adventure yang selalu mendukung penulis untuk tetap semangat dalam mengerjakan tugas akhir ini.
10. Dan kepada Lastribudi, Lela Lailatul, Oxy Firnando, Gian, Filial, yang telah membantu dan memberikan masukan dan semangat selama ini disaat penulis mengalami masalah selama mengerjakan tugas akhir ini.

Akhirnya, Penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi penulis khususnya dan umumnya bagi akademisi lain untuk menambah ilmu pengetahuan.

Bandung, 17 Februari 2016

Penulis

DAFTAR ISI

ABSTRAK ..... **i**
ABSTRACT ..... **ii**
KATA PENGANTAR ..... **iii**
DAFTAR ISI ..... **v**
DAFTAR GAMBAR ..... **vii**
DAFTAR TABEL ..... **viii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Maksud dan Tujuan 2
1.3 Batasan Masalah 2
1.4 Metode Penelitian 2
1.5 Sistematika Penulisan 3
BAB II TINJAUAN PUSTAKA **5**
2.1 Jaringan Komputer 5
2.2 Protokol Jaringan 9
2.3 Open System Interconnection (OSI) 10
2.4 TCP/IP Protocol Suite 13
2.5 Keamanan Jaringan (*Network Security*) 15
2.6 Intrusion Detection System (IDS) 16
2.7 Snort 17
2.8 Basic Analysis and Security Engine (BASE) 20
2.9 Sistem Operasi 20
BAB III PERANCANGAN SISTEM **28**
3.1 Komponen Sistem 29
3.2 Perancangan Sistem 30
3.3 Instalasi Sistem Operasi 31
3.4 Instalasi Paket Pendukung IDS 31
3.5 Instalasi Snort, Barnyard2 dan BASE 32
3.6 Membuat dan Konfigurasi Database Snort 33
3.7 Melakukan konfigurasi Snort, Barnyard2 dan BASE 33

BAB IV PENGUJIAN DAN ANALISA 39
4.1 Persiapan Pengujian.....39
4.2 Tahap-Tahap Pengujian dan Pengambilan Data.....41
4.3 Pengujian dan Pengambilan Data 41
4.4 Analisa 45
BAB V KESIMPULAN DAN SARAN 46
5.1 Kesimpulan.....46
5.2 Saran 46

DAFTAR GAMBAR

Gambar II-1 Topologi Bus 6
Gambar II-2 Topologi Star 7
Gambar II-3 Topologi Ring 8
Gambar II-4 Topologi Mesh 8
Gambar II-5 Lapisan OSI Model 10
Gambar II-6 Lapisan TCP/IP 14
Gambar II-7 Komponen Snort 18
Gambar II-8 Versi Rilis Ubuntu 22
Gambar III-1 Diagram Sistem Secara Umum 28
Gambar III-2 Diagram Alir Perancangan Sistem 30
Gambar IV-1 Tampilan *Network Interface* Pada Server IDS 39
Gambar IV-2 Tampilan Pengujian Koneksi 40
Gambar IV-3 Tampilan Saat Menjalankan Snort IDS 40
Gambar IV-4 Tampilan Running Snort IDS 41
Gambar IV-5 Tampilan Proses DoS 42
Gambar IV-6 Hasil *Log* DoS Pada BASE 42
Gambar IV-7 Hasil *Log* Pada Terminal *Console* 43
Gambar IV-8 Tampilan Proses *Port Scanner* Menggunakan Nmap 43
Gambar IV-9 Log *Port Scanner* Pada *Interface* BASE 44
Gambar IV-10 Log *Port Scanner* Pada Terminal *Console* 44

DAFTAR TABEL

Tabel III-1 Kebutuhan Minimum Perangkat Keras Tablet 29
Tabel III-2 Perangkat Lunak Yang Digunakan 29

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam jaringan komputer, khususnya yang berkaitan dengan keamanan jaringan merupakan hal yang sangat penting untuk diperhatikan, karena akan banyak terjadi hal-hal yang dapat mengganggu keamanan atau kestabilan koneksi jaringan komputer tersebut, baik yang berkaitan dengan *hardware* (pengaman fisik, sumber daya listrik) maupun yang berkaitan dengan *software* (sistem, konfigurasi, sistem akses).

Beberapa contoh masalah yang sudah umum dalam keamanan jaringan adalah adanya gangguan pada sistem yang dapat terjadi karena faktor ketidak sengajaan yang dilakukan oleh pengelola (*human error*), akan tetapi tidak sedikit pula yang disebabkan oleh pihak ketiga yang mencoba mengganggu pengguna jaringan tersebut yang berupaya melakukan perusakan, penyusupan atau penyalahgunaan data maupun sistem. Maka untuk membantu mempermudah administrator jaringan dalam mengelola jaringan tersebut dibutuhkan satu sistem yang dapat membaca dan mengenali setiap paket data yang masuk atau yang keluar secara cepat dan tepat.

Salah satu solusi yang dapat digunakan adalah dengan membangun *Intrusion Detection System* (IDS). IDS sendiri dapat membaca paket-paket data yang masuk maupun yang keluar secara otomatis yang nantinya akan memberikan sebuah laporan atau *log* kepada administrator jaringan.

Dengan memanfaatkan sistem IDS pada suatu jaringan maka kemungkinan adanya gangguan yang disebabkan oleh pihak ketiga terhadap suatu jaringan dapat di meminimalisir, karena nantinya setiap gangguan yang disebabkan akan terdeteksi oleh IDS, dan nantinya IDS akan memberikan peringatan atau alert kepada admin jaringan.

1.2 Maksud dan Tujuan

Maksud dari penelitian ini adalah untuk membangun sistem keamanan jaringan pada sistem operasi Linux Ubuntu Desktop 14.04 LTS sebagai server IDS yang nantinya akan membaca setiap paket-paket data yang masuk maupun yang keluar.

Sedangkan tujuan yang ingin dicapat dari penelitian ini adalah :

1. Membangun sistem IDS pada sistem operasi Linux Ubuntu Desktop 14.04 LTS.
2. Membantu mempermudah dalam mengenali paket data yang masuk dalam jaringan.
3. Mengetahui kapan terjadinya waktu gangguan pada jaringan secara cepat.
4. Dapat mengenali jenis gangguan secara tepat dan jelas.

1.3 Batasan Masalah

Adapun batasan-batasan masalah dalam membangun *Intrusion Detection System* ini :

1. Server Snort IDS dibangun pada sistem operasi Linux Ubuntu Desktop 14.04 LTS.
2. Menggunakan perangkat lunak Snort IDS sebagai pendeteksi gangguan.
3. Menggunakan *Basic Analysis and Security Engine* (BASE) sebagai *interface* untuk pembacaan log gangguan.
4. Pengujian gangguan hanya dilakukan dengan dua cara, yaitu : dengan teknik Denial of Service (DoS) dan *Port Scanner* menggunakan aplikasi Nmap.
5. *Implementasi* sistem hanya dilakukan pada jaringan lokal.

1.4 Metode Penelitian

Metodologi yang digunakan dalam penelitian ini adalah:

1. Studi Pustaka

   Merupakan metode pengumpulan data yang dilakukan dengan cara mencari referensi, membaca, mempelajari buku-buku yang berhubungan dengan sistem *Intrusion Detection System* (IDS).

2. Analisis dan Perancangan Sistem

   Melakukan analisis terhadap masalah-masalah yang ada pada penelitian yang dilakukan dan mengumpulkan perangkat yang akan digunakan dalam pembuatan sistem IDS.

3. Implementasi

   Menerepak sistem yang telah dirancang serta dianalisis sistem tersebut untuk dibuktikan.

4. Pengujian dan Analisa

   Melakukan percobaan dan meng-evaluasi sistem yang telah dirancang dan menganalisa sistem tersebut.

5. Kesimpulan

   Merupakan hasil dari apa yang telah dilakukan selama perancangan sistem sampai uji coba serta analisa yang telah didapatkan dari penelitian ini.

1.5 Sistematika Penulisan

Penelitian ini disusun berdasarkan sistematika penulisan adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan secara singkat mengenai latar belakang, maksud dan tujuan, batasan masalah dan metode penelitian dengan maksud memberikan gambaran tentang isi tugas akhir ini.

BAB II DASAR TEORI

Bab ini membahas tentang berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan untuk membangun sistem.

BAB III PERANCANGAN SISTEM

Bab ini menjelaskan tentang blok-blok sistem yang dirancang serta di implementasikan. Parameter-parameter sistem, blok diagram dan diagram alir sistem.

BAB IV PENGUJIAN DAN ANALISA

Bab ini membahas tentang hasil penelitian dan hasil pengujian dari sistem yang telah dirancang.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari pengujian sistem dan saran yang bermanfaat bagi perbaikan dan perkembangan sistem.

BAB II
TINJAUAN PUSTAKA

2.1 Jaringan Komputer

Merupakan hubungan dari sejumlah perangkat yang dapat saling berkomunikasi satu sama lain. Pada jaringan komputer terdapat dua buah komputer atau perangkat yang saling terhubung satu sama lain dan terjadi proses komunikasi dan transfer paket data di dalamnya.

Berdasarkan klasifikasinya, jaringan komputer dibagi menjadi :

1. Cakupan area geografis
2. Topologi jaringan
3. Arsitektur Komputer

2.1.1 Cakupan Area Geografis

Berdasarkan cakupan area goegrafis, jaringan komputer dibagi menjadi :

1. *Local Area Network* (LAN)

   LAN digunakan untuk menghubungkan komputer yang berada didalam suatu area kecil, misal didalam gedung kantor atau kampus. Jarak antara komputer yang dihubungkan biasa mencapai 5 sampai 10Km. suatu LAN biasanya bekerja pada kecepatan 10 Mbpa sampai 100 Mbps.
2. *Metropolitan Area Network* (MAN)

   MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang lokasinya berjahuan. Jangkauan MAN bias mencapai 10Km sampai beberapa ratus Km. MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps.
3. *Wide Area Network* (WAN)

   WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari satu kota ke kota lain di suatu Negara. Cakupan WAN bias meliputi 100Km sampai 1,000 Km, dan kecepatan antar kota bias bervariasi antara 1,5 Mbps sampai 2,4

Gbps. Dalam WAN, biaya untuk transmisi sangant tinggi, dan biasanya jaringan WAN dimiliki dan di operasikan sebagai suatu jaringan publik.

2.1.2 Topologi Jaringan Komputer

Topologi jaringan adalah gambaran perencanaan hubungan antarkomputer dalam LAN yang umumnya menggunakan kabel (sebagai media transmisi), dengan konektor, *ethernet card*, dan perangkat pendukung lainnya. Ada beberapa jenis topologi yang terdapat pada jaringan komputer, seperti:

1. Topologi Bus

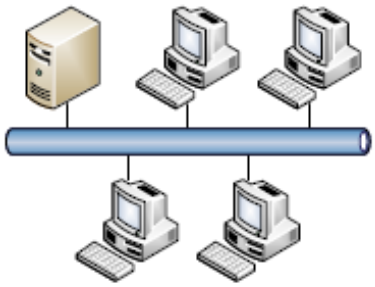
Topologi bus merupakan koneksi bertipe *multipoint,* sebuah bentang kabel tunggal yang panjang akan bertindak sebagai *backbone* yang akan menghubungkan semua perangkat dalam jaringan. *Node-node* terhubung ke kabel *Bus* melalui *drop line* dan *tap. Drop line* adalah suatu konektor yang berjalan antara perangkat dan kabel utama. *Tap* adalah konektor yang juga tersambung pada kabel utama (terhubung langsung pada inti logam kabel).

Kelebihan dari topologi *Bus :*

- *Instalasi* mudah. Kabel *backbone* dapat diletakan ditempat manapun yang paling efisien, kemudian dihubungkan pada *node-node* melalui *drop line.*
- Penggunaan kabel lebih sedikit.

Kekurangan topologi *Bus*:

- *Backbone* rusak maka semua jaringan lumpuh atau tidak berfungsi.
  Sangat sulit untuk mengindentifikasi permasalahan jika jaringan sedang rusak.



Gambar II-1 Topologi Bus

2. Topologi Star

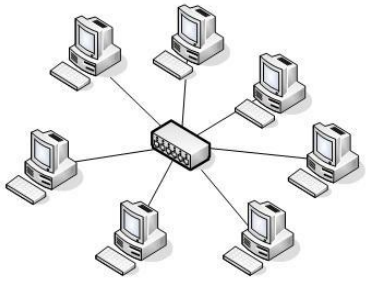
Topologi star adalah topologi yang memiliki pusat berupa *hub* dan *switch* yang berfungsi untuk menghubungkan dan meneruskan pengiriman data antar perangkat.

Kelebihan topologi *Star* :

- Satu *link* yang rusak tidak akan mempengaruhi *link* yang lain.
- Mudah dalam hal *installasi* dan *konfigurasi.*
- Penambahan dan pengurangan perangkat hanya melibatkan perangkat itu sendiri

Kekurangan topologi *Star*:

- Membutuhkan lebih banyak kabel.
- Jika pusat penghubung rusak atau mati semua perangkat yang terhubung tidak dapat melakukan pengiriman atau pengambilan data.



Gambar II-2 Topologi Star

3. Topologi Ring

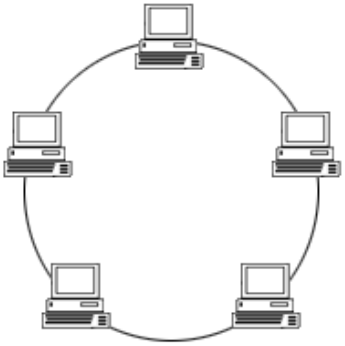
Topologi *Ring* adalah topologi yang setiap perangkat terhubung secara seri dengan yang lain sehingga membentuk satu lingkaran yang tertutup (*loop*).

Kelebihan Topologi *Ring*:

- *Installasi* dan *konfigurasi* relative mudah.
- Didukung oleh banyak perangkat keras dan perangkat lunak

Kekurangan Topologi *Ring*:

- Sulit untuk dikembangkan keselakal jaringan yang besar.
  Jika jaringan utama rusak, maka keseluruhan *segment* akan rusak juga.



Gambar II-3 Topologi Ring

4. Topologi Mesh

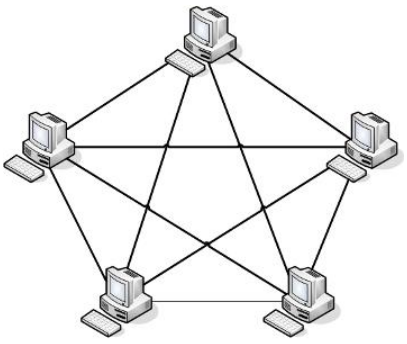
Setiap perangkat terhubung secara *dedicade point-to-point* kesemua perangkat. Dan beban koneksi satu *link* tidak akan menggang *link* lainnya

Kelebihan topologi *mesh:*

- *Dedicade link* menjamin bahwa setiap koneksi hanya membawa beban datanya sendiri. Hal ini menghilangkan masalah *traffic* yang terjadi jika *link* dibagi.
- Kerusakan satu *link* tidak akan menggangu *link* lainnya.
- Data aman karena hanya ada pengirim dan penerima, tidak bisa di *intervensi* oleh user lain.

Kekurangan topologi *mesh*:

- Butuh banyak kabel dan butuh banyak port.
- Mahal dan sulit saat *installasi*.



Gambar II-4 Topologi Mesh

2.1.3 Arsitektur Jaringan Komputer

Berdasarkan arsitekturnya jaringan dibagi menjadi:

1. *Client Server*

   Client Server adalah jaringan komputer yang salah satu komputer difungsikan sebagai server atau komputer yang memberikan pelayanan kepada komputer lain. Server melayani komputer-komputer lain yang disebut client. Layanan yang diberikan bisa berupa akses web, e-mail, file atau yang lainya. *Client Server* banyak dipakai pada *Internet.* Namun LAN atau jaringan lainpun bisa mengimplementasikan *cleint server*.

2. *Peer to Peer*

   Peer-to-peer adalah jaringan komputer dimana setiap komputer bisa menjadi server sekaligus client. Setiap komputer dapat menerima dan memberikan akses dari satu komputer ke komputer lain.

2.2 Protokol Jaringan

Pada jaringan komputer, komunikasi terjadi antara *entitas* dalam sistem yang berbeda. *Entitas* adalah segala sesuatu yang mampu mengirim dan menerima informasi. Namun dua *entitas* tadak dapat langsung mengirim aliran bit satu sama lain, juga tidak dapat langsung saling mengerti. Agar komunikasi dapat dilakukan, *entitas-entitas* tersebut harus memiliki suatu protokol.

Protokol adalah kumpulan aturan yang mengatur komunikasi data. Protokol mengindentifikasi apa itu komunikasi, bagaimana cara berkomunikasi, serta kapan komunikasi dapat dilakukan.

Elemen kunci dari protokol adalah sebagai berikut[1]:

1. *Syntax*

   Istilah *syntax* merupakan struktur atau format data yang digunakan untuk mengkodekan sinyal.

2. *Semantics*

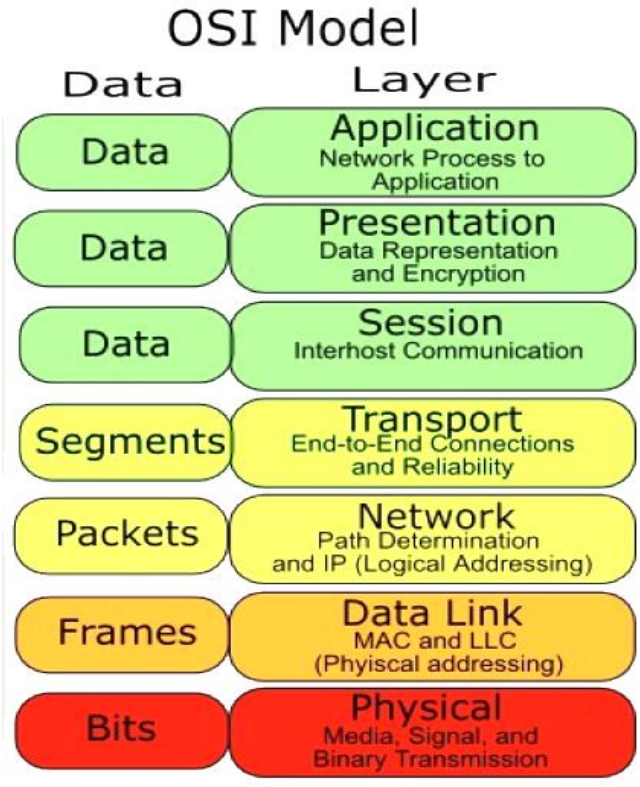
   Semantics digunakan untuk mengetahui maksut dari informasi yang dikirim dan mengoreksi kesalahan yang terjadi dari informasi

3. *Timing*

Istilah *Timing* mengacu pada dua karakteristik kapan data dikirim dan seberapa cepat data tersebut dapat dikirim.

2.3 Open System Interconnection (OSI)

Open System Interconnection (OSI) adalah kumpulan protokol yang memungkinkan dua sistem yang berbeda dapat berkomunikasi tanpa memperhatikan arsitektur yang mendasari sistem tersebut. Tujuan OSI model adalah untuk menunjukan bagai mana cara untuk memfasilitasi dua sistem yang berbeda tanpa memerlukan perubahan logika pada *hardware* dan *software*. OSI model bukan suatu protocol melainkan model untuk memahami dan merancang arsitektur jaringan yang *fleksibel*, kokoh dan mudah di operasikan. OSI model merupakan *framwork* berlapis untuk merancang sistem jaringan yang memungkinakan komunikasi antara semua jenis komputer[1].



Gambar II-5 Lapisan OSI Model

Berikut penjelasan fungsi dari gambar model OSI diatas :

1. *Application layer* (Lapisan Aplikasi)

   Application Layer bertugas untuk menyediakan layanan pada pengguna. Layanan ini memungkinkan pengguna, baik manusia atau *software* untuk

mengakses jaringan. Layer ini menyediakan user *interface* dan dukungan untuk layanan-layanan seperti e-mail, akses remote file dan layanan-layanan lainnya.

2. *Presentation layer* (Lapisan Presentasi)

Presentation Layer berhubungan dengan format data yang digunakan antara dua sistem yang saling bertukar data. *Presentation Layer* memiliki tugas utama yaitu :

- Translasi yang berguna untuk menerjemahkan kode data dari setiap sistem yang berbeda-beda
- *Enkripsi* yang berguna untuk mengubah informasi asli yang akan dipertukarkan menjadi bentuk lain agar aman ketika melalui jalur pengiriman.
- Kompresi yang berguna untuk mengurangi jumlah bit pada informasi yang akan dipertukarkan.

3. *Session Layer* (Lapisan Sesi)

Session Layer bertugas untuk mengelola dialog dan *sinkronisasi.* Tanggung jawab dari *Session Layer* adalah sebagai *Dialog Control* yaitu lapisan sesi yang memungkinkan kedua sistem untuk membangung dialog serta tanggung jawab untuk *Synchronization* pada interaksi antara sistem komunikasi.

4. *Transport Layer* (Lapsisan Transport)

Transport layer bertugas untuk mengirim pesan dari satu proses ke proses yang lain. Proses adalah program aplikasi yang berjalan pada host. Tugas lain *Transport Layer* adalah:

- *Service-point addressing* karena sering menjalankan beberapa program pada waktu yang sama, maka pengirim *source-to-destination* tidak hanya dari satu komputer ke komputer lain, tapi juga dari satu *spesificprocess* pada komputer lain.
- *Segment and reasembly*. Berguna untuk memastikan seluruh pesan yang sampai dibagian penerima dalam keadaan utuh dan sesuai urutan dan melakukan *indentifikasi* serta pergantian paket jika terjadi kehilangan pada saat proses pengiriman.

- *Connection control*. Lapisan *transport* bisa berupa *connectionless* atau *connection-oriented, Transport layer* secara *connectionless* memerlukan setiap segment sebagai paket yang individu dan mengirimnya ke *transport layer* pada tujuan. *Transport layer connection-oriented* akan membangun koneksi terlebih dahulu dengan *transport layer* pada tujuan sebelum paket dikirim. Dan setelah semua data terkirim, koneksi akan dipututskan.
- *Flow Control* dilakukan di *end-to-end,* bukan di *single link*.
- *Error Control* memastikan seluh pesan sampai pada lapisan *transport* di penerima tanpa ada masalah. *Error Control* dilakukan pada proses ke proses, bukan di *single link.*

5. *Netwok Layer* (Lapisan Jaringan)

Network layer bertugas untuk mengirimkan paket-paket individu dari *host sender* ke *host receiver*, sekalipun melewati banyak jaringan (*link*). Tugas lain dari layer ini adalah:

- *Logical addressing,* network layer akan menambahkan header yang berisi IP sumber dan tujuan pada paket yang datang dari layer diatasnya.
- *Routing,* merupakan mekanisme untuk memilih atau mengerahkan paket ketujuan akhir paket tersebut. Perangkat yang melakukan fungsi tersebut adalah *Router*.

6. *Data Link Layer* (Lapisan Data link)

Data Link layer bertugas dalam perpindahan *frame* dari satu *hop* (mode) ke *hop* yang selanjutnya. Tugas lain dari layer ini adalah:

- *Framing,* aliran bit yang diterima akan dibagikan menjadi *frame-frame.*
- *Physical addressing,* data link layer akan memberikan header pada *frame* untuk mendefenisikan *sender* dan *receiver* dari *frame* tersebut.
- *Flow control,* rate data di-produce akan di sesuaikan dengan rate yang bisa diterima oleh *receiver.* Hal ini dilakukan agar *receiver* tidak kewalahan saat menerima data.
- *Error control* untuk mendeteksi dan mengirim ulang frame yang rusak. *Error control* akan mengetahui jika terjadi duplikat pada *frame.*

- *Access control* jika terdapat dua perangkat atau lebih terhubung pada link yang sama, protokol data link layer akan menentukan perangkat mana yang memiliki kontrol atas link tersebut pada waktu tertentu.

7. *Physical Layer* (Lapisan Fisik)

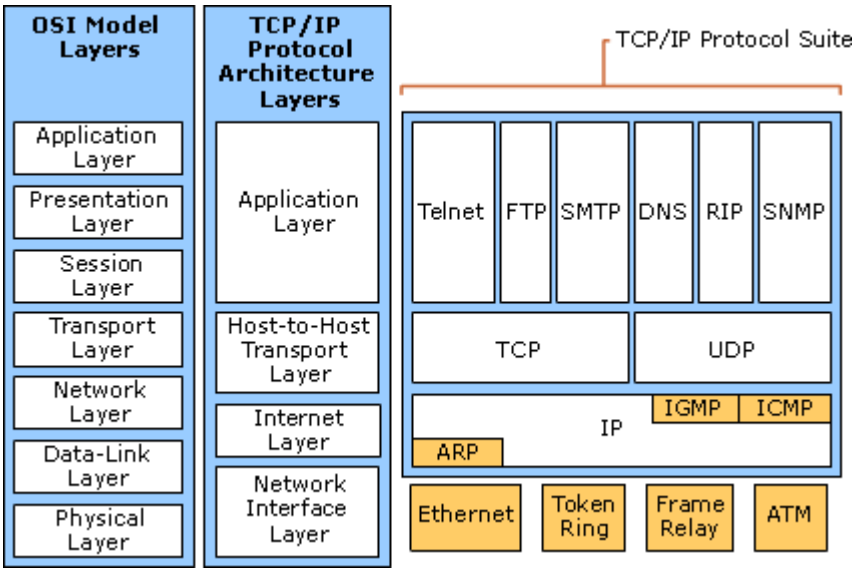
Physical Layer bertugas untuk perpindahan bit-bit tunggal dari satu hop (node) ke hop yang selanjutnya. Layer ini juga menentukan prosedure dan fungsi yang harus dijalankan oleh perangkat fisik dan antarmuka agar transmisi dapat dilakukan. *Physical Layer* juga mengelola hal-hal berikut:

- Karakteristik fisik dari antarmuka dan media perantara.
- Representasi bit data pada layar ini terdiri dari aliran bit (0 atau 1) tanpa *interpretasi*. Agar dapat ditransmisikan, bit-bit ini harus di-*encode* menjadi sinyal listrik atau optik.
- Data rate atau *transmission* rate adalah jumlah bit yang dikirim setiap satu waktu.
- *Sinkronisasi* bit, bit level dan bit rate *sender* dan *receiver* harus sama.
- *Line configuration* layer ini juga memperhatikan koneksi perangkat ke media. Pada konfigurasi point-to-point, koneksinya *dedicated link*. Pada *konfigurasi multipoint*, link di-share pada beberapa perangkat.
- Topologi fisik yaitu menggambarkan secara fisik atau nyata bagaimana perangkat-perangkat terhubung satu sama lain hingga menjadi suatu jaringan.
- *Transmission mode* menentukan arah transmisi dari perangkat yang sedang melakukan pertukaran data. *Simplex* yaitu hanya satu perangkat yang bisa mengirimkan data dan yang lainnya hanya menerima. *Half-Duplex* yaitu dua perangkat bisa mengirim dan menerima data tapi tidak dalam waktu yang bersamaan. *Full-Duplex* yaitu setiap perangkat bisa mengirim dan menerima data dalam waktu yang bersamaan.

2.4 TCP/IP Protocol Suite

TCP/IP protocol suite dikembangkan sebelum model OSI. Namun lapisan-lapisan pada TCP/IP tidak sama dengan lapisan-lapisan yang dimiliki oleh OSI.

Protocol TCP/IP hanya diabuat atas lima lapisan, yaitu: *pyshical, data link, network, transport* dan *application.*



Gambar II-6 Lapisan TCP/IP

Berikut penjelasan dari setiap lapisan protocol TCP/IP :

1. *Aplication layer* (Lapisan Aplikasi)

   Lapisan aplikasi pada TCP/IP setara dengan tiga lapisan yang terdapat pada model OSI. Bentuk data pada lapisan ini biasa disebut dengan *message.* Komunikasi pada lapisan ini juga sama seperti lapisan transport yaitu *end-to-end.* Beberapa protocol yang ada pada lapisan ini sebagai berikut:

   - HTTP (*Hyper Text Transfer Protocol*) yang digunakan untuk layanan web.
   - Telnet yang berfungsi untuk melakukan remote pada suatu host
   - FTP (*File Transfer Protocol*) yang digunakan untuk keperluan *file sharing.*

2. *Transport* Layer (Lapisan Transport)

   Terdapat perbedaan utama pada lapisan jaringan dan lapisan transport. Setiap node memerlukan lapisan transport. Dilapaisan ini pula terdapat protokol yang bisa digunakan untuk pengiriman data yaitu *User Datagram Protocol* (UDP) dan *Transmission Control Protocol* (TCP).

3. *Internet Layer* (Lapisan Internet)

   Berfungsi untuk menyediakan *Routing* dan pembuatan paket menggunakan teknik *encaptulation*. *Internet layer* memiliki tugas utama untuk memilih

rute dalam sebuah jaringan. Selain itu layer ini juga bertugas untuk melakukan *paket swictching*.

4. *Network Layer* (Lapisan jaringan)

   Berfungsi untuk menyediakan *frame-frame* data yang akan dikirim kemedia jaringan. Layer ini bertugas mengatur semua hal yang diperlukan oleh sebuah paket IP.

2.5 Keamanan Jaringan (*Network Security*)

Keamanan jaringan merupakan hal yang sangat penting untuk diperhatikan dalam sebuah jaringan komputer (*Network Securty*). Adanya keamanan jaringan maka resiko yang tidak diinginkan akan dapat dikurangi, dengan demikian pada *user* atau pengguna akan merasa aman saat bekerja di jaringan komputer yang mereka gunakan.

Sebuah jaringan dengan desain dasar yang aman memerlukan beberapa penerapan aturan dasar yang sederhana yang perlu disesuaikan dengan kontek yang berbeda, seperti berikut :

- Menutup port IP yang tidak terpakai.
- Memusatkan titik *control traffic.*
- Periksa semua jalur akses dan desain multilayer dalam kasus jaringan yang kompleks.

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

1. *Confidentiality*

   Adanya data-data yang penting yang biasanya tidak boleh diakses oleh seseorang, sehingga dilakukan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. Biasanya *confidentiality* ini berhubungan dengan informasi yang diberikan kepihak lain.

2. *Integrity*

   Pesan yang disampaikan ke penerima tetap orisinil yang tidak diragukan keasliannya, tidak dimodifikasi selama dalam pengiriman sumber ke penerimanaya.

3. *Availability*

Dimana *user* yang mempunyai hak akses diberikan akses tepat pada waktunya, biasanya ini berhubungan dengan ketersedian informasi atau data ketika dibutuhkan.

Keamanan jaringan komputer sendiri bertujuan untuk mengantisipasi resiko pada jaringan komputer, berupa bentuk ancaman fisik maupun logik, baik langsung (*direct*) ataupun tidak langsung (*indirect*) menggangu aktivitas yang sedang berlangsung dalam jaringan komputer. Secara umum, terdapat 3 hal dalam konsep keamanan jaringan, yakni :

1. Resiko (*Risk*): resiko disini adalah untuk menyatakan besarnya kemungkinan gangguan yang muncul terhadap jaringan.
2. Ancaman (*Threat*): ancaman disini merupakan kemungkinana gangguan yang muncul terhadap jaringan.
3. Kerapuhan sistem (*Vulnerability*): ancaman disini menyatakan kelemahan-kelemahan pada sistem yang memungkinkan terjadinya gangguan.

2.6 Intrusion Detection System (IDS)

IDSmerupakan sistem untuk mendeteksi adanya *intrusion* yang dilakukan oleh *intruder* atau penyusup di dalam jaringan. IDS sendiri sangat mirip seperti alarm, apabila IDS mencatat adanya suatu serangan atau gangguan dalam jaringan, maka IDS akan memperingati administrator jaringan tersebut. IDS (I*ntrusion Detection System*) dapat didefenisikan sebagai kegiatan yang bersifat *anomaly*, *incorrect*, *inappropriate* yang terjadi dijaringan atau host. Dan IDS sendiri adalah sistem keamanan yang bekerja bersama *Firewall* untuk mengatasi *Intrusion*[2].

Berikut beberapa komponen yang terdapat pada IDS [4]:

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

Apabila ada aktivitas yang dianggap mencurigakan di dalam jaringan makan IDS akan mendeteksi dan mencatat semua aktivitas yang mencurigakan tersebut.

Namun IDS ini tidak dapat melakukan tindakan atau pencegahan jika terjadi serangan atau penyususpan didalam jaringan.

Namun IDS mempunyai peran yang cukup membantu dalam hal yang berkaitan dengan keamanan jaringanan, diantaranya :

a. Secara aktif mengamati segala macam kegiatan yang mencurigakan.
b. Memeriksa *audit logs* dengan sangat cermat dan seksama.
c. Mengirim *alert* kepada administrator saat adanya serangan yang terdeteksi.
d. Memberikan laporan segala macam kerentanan yang ditemukan.

Namun kembali lagi kepada kemampuan IDS yang terbatas, yang hanya bisa mencatat dan memberikan laporan kepada administrator, karena IDS tidak dapat menghentikan serangan tersebut.

Dilihat dari kemampuan mendeteksi serangan dalam jaringan, maka IDS dibagi menjadi dua, yaitu [3]:

1. Network-Based IDS (NIDS)

Network-Based Intrusion Detection System (NIDS) akan melakukan *capture paket data* yang masuk pada jaringan LAN atau Wireless dan mengumpulkan paket-paket data tersebut, serta melakukan analisa terhadapt paket-paket tersebut apakah paket normal atau paket serangan.

2. Host-Based IDS (HIDS)

Host-based Intrusinon detection system (HIDS) hanya melakukan pemantauan pada perangkat komputer tertentu dalam jaringan. IDS dapat melihat kedalam sistem dan aplikasi file log untuk mendeteksi aktivitas penyusup. HIDS hanya akan berkerja apabila terjadi serangan dan akan memberikan laporan atau peringakatan kepada administrator secara *real time.*

2.7 Snort

Snort merupakan *open source intrusion detection system* (IDS) yang bisa di konfigurasi sebagai *intrusion prevention system* (IDS) untuk pemantauan dan pencegahan pada keamanan komputer.

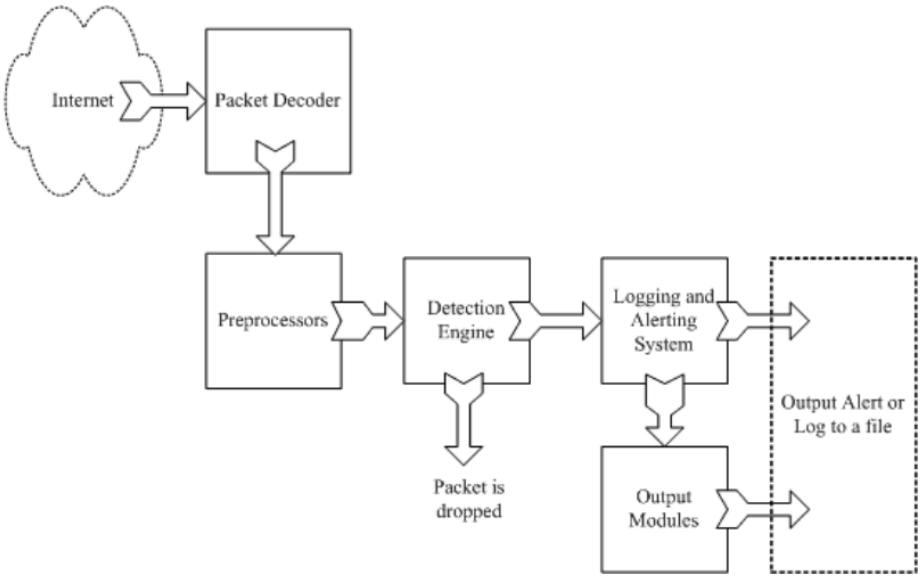
Ketika bekerja sebagai IDS, Snort dapat mendeteksi serangan atau paket data yang masuk kedalam jaringan komputer dan memperingatkan administrator jaringan untuk mengambil tindakan pencegahan. Jika kita menngkonfigurasi Snort

sebagai IPS maka Snort dapat mencegah serangan atau paket data yang berbahaya yang masuk kedalam jaringan, dengan cara memodifikasi IP Table.

Snort secara logis dibagi menjadi beberapa komponen. Komponen-komponen ini bekerja sama untuk mendeteksi serangan tertentu dan untuk mengasilkan *output* dengan format yang dibutuhkan pada *system detection*[4].

Berikut beberapa komponen utama pada Snort :

1. Paket decoder
2. Preprocessors
3. Detection Engine
4. Logging and Alerting System
5. Output modules



Gambar II-7 Komponen Snort

2.7.1 Paket Decoder

Paket Decoder berfungsi untuk mengambil paket dari berbagai jenis *network interface* dan mempersiapkan paket yang akan di *preprocessed* atau dikirim ke *Detection Engine*.

2.7.2 Preprocessors

Preprocessor adalah komponen atau plug-in yang dapat digunakan dengan Snort untuk mengatur atau memodifikasi paket data sebelum *detection engine*

melakukan beberapa operasi untuk mengetahui apakah paket yang digunakan oleh penyusup. Beberapa preprocessor juga melakukan deteksi dengan mencari anomaly dalam header paket dan menghasilkan *alert*. *Preprocessor* sangat penting untuk setiap IDS untuk mempersiapkan paket data yang akan dianalisa terhadapat aturan dalam *detection engine*.

2.7.3 Detection Engine

Detection Engine adalah bagian paling penting dari Snort. *Detection Engine* bertanggung jawab untuk mendeteksi jika ada aktivitas *intrusi* dalam sebuah paket. *Detection Engine* menggunakan *rule* Snort. Untuk tujuan ini, rule dibaca dengan struktur data internal atau dimana mereka cocok dengan semua paket. Jika sebuah paket cocok dengan rule apapun, maka tindakan yang tepat akan dilakukan atau diambil tetapi jika tidak paket akan dibuang.

Detection Engine adalah *time-critical* bagian dari Snort. Tergantung pada seberapa kuat mesin adan berapa banyak aturan telah ditetapkan, mungkin diperlukan jumlah waktu yang berbeda untuk merespon paket yang berbeda, jika lalulintas (*traffic*) di jaringan terlalu tinggi, maka ketika Snort NIDS bekerja dalam model ini, mungkin ada beberapa paket yang di drop dan mungkin tidak mendapatkan real-time respone yang benar.

Beberapa faktor beban pada *detection engine* :

- Jumlah rule
- Kekuatan mesin dalam menjalankan Snort
- Kecepatan internet bu yang dugunakan dalam mesin Snort.
- Load pada jaringan.

2.7.4 Logging and Alerting System

Tergantung pada apa yang *detection engine* temukan dalam sebuah paket, paket digunakan untuk mencatat aktivitas atau menghasilkan peringatan. Nantinya log akan disimpan dalam bentuk file teks sederhana.

2.7.5 Output Modules

Output module atau plug-in dapat melakukan operasi yang berbeda tergantung pada administrator dalam menyimpan output yang dihasilkan oleh *logging* dan *system alert* dari Snort. Pada dasarnya modile ini mengontrol jenis putput yang dihasilkan oleh logging dan memperingatkan system.

2.8 Basic Analysis and Security Engine (BASE)

BASE merupakan *interface* yang akan digunakan untuk menganalisa *output* dari Snort. BASE bekerja dengan menggunakan *query* pada *log database* Snort yang akan disimpan kedalam *database mysql server*. Kemudian akan menampilkan kedalam bentuk web sehingga *log database* tersebut mudah untuk di analisa.

2.9 Sistem Operasi

Suatu program yang mengontrol eksekusi program aplikasi dan berfungsi sebagai media *interface* antara pengguna komputer dengan hardware komputer. System operasi memiliki tiga tujuan :

1. Kemudahan : sistem operasi membuat komputer menjadi lebih mudah dipakai
2. Efesiensi : sistem operasi memungkinkan sumber daya system komputer untuk digunakan dengan cara yang efesien
3. Kemampuan berkembang : sistem operasi harus disusun sedemikian rupa sehingga memungkinkan pengambang yang efektif, pengujian, dan penerapan fungsi-fungsi system yang baru tanpa menggangu layanan yang telah ada.

2.10 Linux

Linux awalnya merupakan varian dari UNIX untuk arsitektur IBM. Versi awal dituliskan oleh Linux Torvalds, seorang mahasiswa ilmu komputer dari Firlandia. Torvalds memposting versi awal Linux di Internet tahun 1991. Karena Linux gratis dan sumber kodenya tersedia, menjadi alternative awal untuk workstation UNIX seperti yang ditawarkan oleh Sun Microsystem, Digital Equipment Corp (sekarang menjadi Compaq), dan Silicon Graphich.

2.10.1 Ubuntu

Ubuntu diambil dari Bahasa Afrika kuno yang berarti *Humanity to Others* atau rasa peri kemanusiaan terhadapat sesama. Tujuan dari distribusi Linux Ubuntu adalah membawa semangat yang terkandung didalam Ubuntu ke dunia perangkat lunak. Ubuntu merupakan varian atau turunan dari Debian, yang merupakan salah satu distro Linux tertua selain Red Hat dan Slackware.

Ubuntu dibentuk berdasarkan gagasan yang terdapat dalam filosofi Ubuntu, yaitu perangkat lunak harus tersedia secara gratis dan tidak ada biaya lisensi, perangkat lunak harus dapat digunakan dalam Bahasa lokal masing-masing dan untuk orang-orang yang mempunyai keterbatasan fisik, serta bersifat serta bersifat open source sehingga pengguna memiliki kebebasan untuk mengubahnya sesuai dengan kebutuhan komputasi mereka.

Kebebasan inilah yang membuat Ubuntu berbeda dari pesaingnya misalnya Microsoft Windows yang bersifat *proprietary* atau berlisensi. Ubuntu dapat diperoleh secara gratis apabila anda memintanya langsung kepada Canonical Ltd. Selain itu, Ubuntu juga memberikan kebebasan untuk memodifikasi Ubuntu agar menjadi distro Linux yang di inginkan. Bahkan, dapat menamai sendiri versi Ubuntu hasil modifikasi tersebut.

2.10.2 Rilis Ubuntu

Setiap rilis Ubuntu memiliki keunikan pada nama kode dan nomor versi yang dikeluarkan. Misalnya nama diambil dari nama binatang dan diikuti dengan nomor versi berdasarkan tahun dan bulan dari rilis. Setiap tahun umumnya akan ada 2 rilis Ubuntu, biasanya pada bulan April dan Oktober. Edisi Ubuntu biasanya di rilis sedikitnya dalam 3 edisi yaitu *Desktop, Alternate* dan *Server*. Perbedaan mendasar ketiganya adalah kegunaan dan cara instalasi. Edisi Desktop digunakan untuk komputer personal dengan instalasi mode grafis. Ubuntu Alternate juga digunakan untuk komputer personal namun instalasi menggunakan mode teks. Ubuntu Server digunakan untuk komputer server dengan instalasi teks. Berikut versi-versi Ubuntu dari awal rilis sampai sekarang :

Versi	Kode nama	Tanggal Perilisan	Didukung sampai	
			Desktop	Server
4.10	Warty Warthog	20-10-2004	2006-04-30	
5.04	Hoary Hedgehog	08-04-2005	2006-10-31	
5.10	Breezy Badger	13-10-2005	2007-04-13	
6.06 LTS	Dapper Drake	01-06-2006	2009-07-14	2011-06-01
6.10	Edgy Eft	26-10-2006	2008-04-25	
7.04	Feisty Fawn	19-04-2007	2008-10-19	
7.10	Gutsy Gibbon	08-10-2007	2009-04-18	
8.04 LTS	Hardy Heron	24-04-2008	2011-05-12	2013-04
8.10	Intrepid Ibex	30-10-2008	2010-04-30	
9.04	Jaunty Jackalope	23-04-2009	2010-10-23	
9.10	Karmic Koala	29-10-2009	2011-04-30	
10.04 LTS	Lucid Lynx	29-04-2010	2013-04	2015-04
10.10	Maverick Meerkat	10-10-2010	2012-04-10	
11.04	Natty Narwhal	28-04-2011	2012-10-28	
11.10	Oneiric Ocelot	13-10-2011	2013-05-09	
12.04 LTS	Precise Pangolin	26-04-2012	2017-04	
12.10	Quantal Quetzal	18-10-2012	2014-05-16	
13.04	Raring Ringtail	25-04-2013	2014-01-27	
13.10	Saucy Salamander	17-10-2013	2014-07-17	
14.04 LTS	Trusty Tahr	17-04-2014	2019-04	
14.10	Utopic Unicorn	23-10-2014	2015-07	
15.04	Vivid Vervet	23-04-2015	2016-01	

Warna	Arti
Merah	Sudah tidak didukung
Hijau	Masih didukung
Biru	Rilis masa depan

Gambar II-8 Versi Rilis Ubuntu

- **Ubuntu 4.10 (Waty Wathog)**

  Inilah cikal bakal Ubuntu, yang diluncurkan tanggal 20 Oktober 2004. Kode namanya adalah Warty Warthog dengan nomor kode 4.10 yang muncul dengan logo sedikit aneh yaitu babi hutan. Rilis ini dilanjutkan dengan komitmen untuk secara simultan melakukan update security untuk jangka waktu 18 bulan sesuai dengan rangkaian Ubuntu Security Notices pada saat itu. Tampilan desktop yang indah membuat Ubuntu 4.10 menjadi idola di rumah, sekolah dan bisnis hampir diseluruh belahan dunia. Pada sat itu

masih belum banyak Linux Desktop berbasis grafis yang memiliki tampilan interface menarik. Beberapa fitur lainnya adalah instalasi cepat dan sederhana, ketersediaan paket secara online, sudah terintegrasi baik dengan pengguna laptop, Firefox 0.9, Evolution 2.0, OpenOffice.org 1.1.2 dan juga dukungan untuk hardware. Server Ubuntu 4.10 bisa diinstalasikan dengan mode minimalis.

- **Ubuntu 5.04 (Hoary Hedgehog)**

  Kode nama Hoary Hedgehog adalah rilis Ubuntu kedua yang dikeluarkan dengan kode nomor 5.04 pada tanggal 8 April 2005. Icon yang digunakan adalah binatang landak. Pada versi ini merupakan awal mula dari Kubuntu, yaitu distro Linux Ubuntu yang berbasiskan KDE sebagai desktop environment. KDE yang digunakan pada Kubuntu ini adalah versi 3.4, sedangkan Ubuntu sendiri menggunakan basis GNOME versi 2.10. Pada versi ini terdapat fasilitas baru untuk melakukan update yang dapat berjalan pada modus background. Prooses booting pada versi ini sudah lebih cepat dibanding versi pendahulu. Pun demikian versi ini lebih banyak mengenali hardware pada komputer. Versi ini juga mendukung instalasi melalui media USB serta men-support program kompresi bzip2, autentifikasi Apt dan juga sudah adanya versi Live CD yang memungkinkan anda menggunakan Ubuntu tanpa perlu di-install terlebih dahulu.

- **Ubuntu 5.10 (Breezy Badger)**

  Rilis ketiga dari Ubuntu ini menggunakan binatang jenis luwak sebagai icon-nya. Distro yang dirilis pada 13 Oktober 2005 ini hadir dengan GNOME sebagai desktop *environment* dan didalamnya sudah terpasang program aplikasi OpenOffice versi 2.0 beta 2. Untuk memperkaya content-nya, hadir juga EDUbuntu, yaitu jenis baru dari Ubuntu yang berfokus dalam dunia pendidikan. EDUbuntu berisi aplikasi-aplikasi edukatif dan dapat digunakan oleh siswa untuk sarana belajar. Didalamnya terdapat aplikasi matematika, fisika, kimia, serta aplikasi edukatif lainnya. Fitur-fitur tambahan pun banyak ditemui pada versi ini seperti hadirnya usplash, yaitu loading bar pada saat proses Ubuntu. Fitur lainnya berupa dukungan untuk menulis audio CD, tambahan *driver hardware* dari *third party*, dukungan

penuh untuk printer dan *scanner Hewlett-Packard*, kompatibilitas keyboard dan mouse Bluetooth, serta kernel 64-bit untuk Power PC.

- **Ubuntu 6.06 (Dapper Drake)**

  Rilis keempat muncul pada tanggal 1 Juni 2006 dengan nama Dapper Drake dan berkode nomor 6.06. Icon versi ini adalah binatang bebek. Versi ini muncul dengan basis GNOME dan KDE. Keluarnya *Dapper Drake* menandai semakin bertambahnya jumlah pengguna Ubuntu di Indonesia. Bersamaan dengan momen ini bermunculan pula tempat download lokal di Indonesia. Fitur terbaru yang disertakan adalah cara instalasi melalui Live CD secara cepat, sistem start-up yang singkat pada saat login, menu yang simple, proses shutdown dengan visual grafis, dukungan terhadap video playback dan pre install dengan paket aplikasi openOffice 2.0.2. Dari versi ini lahir jenis baru yaitu Xubuntu. Perbedaannya dengan Ubuntu, Kubuntu dan EDUbuntu adalah desktop environment yang digunakan Xubuntu merupakan XFCE versi 4.3.90.1. Desktop environment ini relatif lebih ringan dibandingkan dengan GNOME atau KDE.

- **Ubuntu 6.10 (Edgy Eft)**

  Pada tanggal 26 Oktober 2006 rilis kelima Ubuntu Edgy Eft muncul. Ubuntu bermaskot kadal berwarna oranye ini memiliki peningkatan performa pada start-up dan shutdown speed serta terintegrasi dengan Tomboy dan F-Spot. Fitur dan content yang dibenamkan antara lain kernel 2.16.17, Evolution 2.8.0, Firefox 2.0 dan OpenOffice 3.0.4. Desktop environment menggunakan GNOME2.16, KDE 3.5.5 dan XFCE 4.3.99 RC1. Pada versi ini terdapat sedikit keistimewaan karena banyaknya aplikasi tambahan diantaranya beryl, xgl dan compiz. Aplikasi-aplikasi add on ini menjadikan tampilan desktop Ubuntu semakin memukau. Ubuntu 6.10 mencuri perhatian orang-orang dan digunakan sebagai pilihan alternatif bagi sistem operasi Microsoft Windows

- **Ubuntu 7.04 (Feisty Fawn)**

  Ubuntu generasi ke-6 ini dirilis pada tanggal 19 April 2007 dengan nama Feisty Fawn, yang menggunakan icon seekor kancil afrika. Didalamnya telah terdapat kernel Linux 2.6.20, desktop GNOME 2.18, KDE 3.5.6 dan

XFCE 4.4. Namun Ubuntu 7.04 ini meminta penggunaan memori RAM minimal 256 MB agar bisa dijalankan di komputer. Pada versi ini mulai digunakan edisi Alternate. Edisi ini digunakan jika anda ingin meng-install dengan waktu yang lebih singkat dibandingkan dengan menggunakan grafis. Fitur terbarunya adalah tool migrasi Windows yang memudahkan integrasi antara aplikasi Windows dan Ubuntu jika anda menggunakan dual booting. Fitur lainnya berupa kemudahan instalasi media codec, serta plug dan play jaringan menggunakan Avahi.

- **Ubuntu 7.10 (Gutsy Gibbon)**

  Pada tanggal 15 Oktober 2007 Ubuntu teranyar dengan nama Gutsy Gibbon dirilis oleh Canonical Ltd. Versi ini menggunakan hewan siamang sebagai maskot icon-nya. Keunggulan pada rilis ini antara lain adalah adanya dukungan add-on atau plugin untuk web browser firefox, wizard untuk dukungan printer, WiFi, kamera digital, IPod dan piranti digital lainnya. Versi ini juga sudah mendukung akses ke system file NTFS secara default. Di versi sebelumnya, tidak memiliki kemampuan untuk membaca system file NTFS. Efek 3D juga disetel secara default jika semua persyaratan minimumnya sudah terpenuhi. Untuk mendukung keamanan jaringan disediakan Apparmor dari Novell. Pun ditambahkan desktop search, switch user yang cepat, dan dukungan untuk thin client atau biasa disebut LTSP (Linux Terminal Server Project).

- **Ubuntu 8.04 (Hardy Heron)**

  Pada 24 April 2008, Ubuntu terbaru muncul dengan kode nama Hardy heron. Ubuntu 8.04 berlambang burung bangau ini mensyaratkan memory RAM minimal sebesar 384 MB. Bertambahnya konsumsi RAM yang semakin besar dikarenakan semakin ciamiknya tampilan grafis dan juga untuk dukungan aplikasi-aplikasi yang digunakan. Hardy Heron merupakan versi LTS (Long Term Support) yaitu paket yang dimiliki oleh Ubuntu 8.04 yang akan didukung selama jangka waktu 3 tahun untuk varian desktop dan 5 tahun untuk varian server. Dukungan ini juga dapat anda temui pada Ubuntu Dapper Drake. Pada versi ini juga muncul istilah Wubi, yaitu installer Ubuntu di dalam OS Windows. Anda tidak perlu memasang

Ubuntu ke dalam harddisk komputer. NFS (*Network File System*) juga sudah didukung di dalam versi ini.

- **Ubuntu 8.10 (Intrepid Ibex)**

  Ubuntu 8.10 Intrepid Ibex dirilis pada tanggal 30 Oktober 2008. Ubuntu ini dapat dijalankan dengan minimal RAM sebesar 256 MB asalkan kapasitas RAM tidak di-share dengan kartu grafis. Ubuntu 8.10 dirancang untuk gaya hidup digital karena banyak fitur-fitur yang mendukung untuk hal ini. Fitur yang dimiliki Ubuntu 8.10 antara lain adalah dukungan jaringan 3G dengan menggunakan kabel ataupun WiFi. Pada rilis ini juga banyak modem-modem dan juga handphone untuk ber-internet ria sudah dikenali dan anda tidak perlu melakukan pengaturan yang rumit. Intrepid Ibex juga mendukung instalasi melalui live USB.

- **Ubuntu 9.04 (Jaunty Jackalope)**

  Jaunty Jackalope dirilis tanggal 23 April 2009 dengan menggunakan maskot icon kelinci bertanduk. RAM minimum yang diperlukan untuk menjalankan versi ini sebesar 256 MB, namun disarankan menggunakan ukuran yang lebih besar. Keunggulan Ubuntu 9.04 ini salah satunya adalah speed booting hanya sekitar kurang lebih 25 detik. Dukungan 3G dan wireless juga disertakan dari versi sebelumnya. Paket OpenOffice yang digunakan adalah versi 3.0. Aplikasi-aplikasi seperti Ubuntu Tweak yang memudahkan anda mengatur sistem hingga manajemen paket dari third party sudah tersedia untuk ubuntu ini. Desktop yang digunakan adalah GNOME 2.26. Perbedaan utama rilis ini dengan versi lain adalah system file Ext4 yang merupakan file sistem terbaru dari Linux setelah Ext3. File sistem ini lebih tahan uji dan stabil karena merupakan penyempurnaan dari system file Ext3.

- **Ubuntu 9.10 (Karmic Koala)**

  Karmic Koala merupakan rilis kandidat yang sesuai jadwal keluar pada tanggal 29 Oktober 2009, dengan lambang Koala. Ubuntu rilis kali ini menyertakan GNOME 2.28.1, KDE 4.3.1, OpenOffice 3.1.1 dan kernel Linux 2.6.31. Karmic Koala mengganti arsitektur akselerasi grafis X.Org server EXA menjadi UXA karena adanya masalah pada driver video Intel tersebut. Pada versi ini, aplikasi-aplikasi yang disertakan semakin lengkap

diservarian baru maupun penambahan fitur dan performa. Karmic Koala sudah mampu mengakomodasi semua kebutuhan komputerisasi user secara total.

- **Ubuntu 10.4 (Lucid Lynx)**

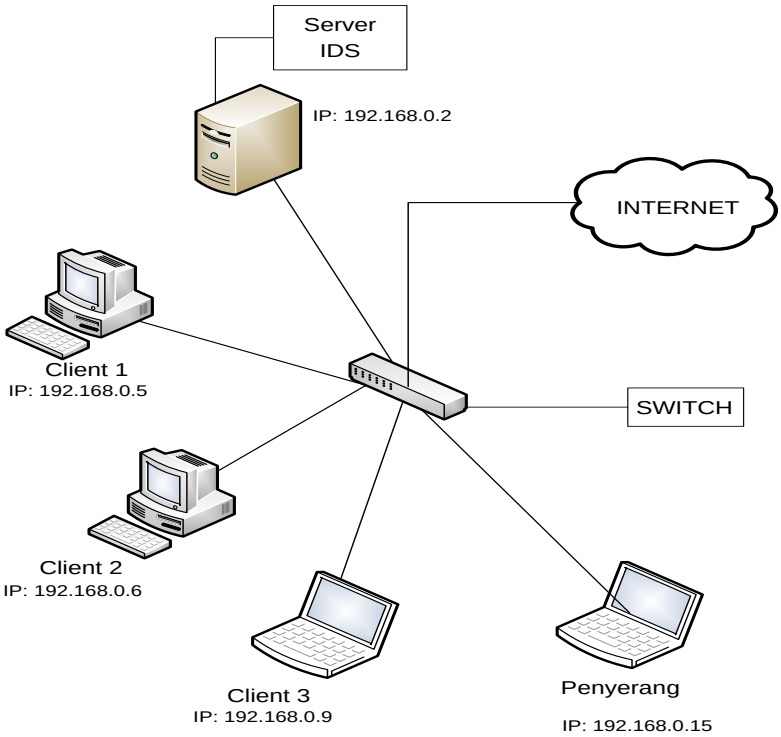
  Pada tanggal 27 April 2010, Canonical Ltd mengumumkan rilis Ubuntu 10.4 Lucid Lynx. Kelebihan versi Ubuntu bermaskot kucing gunung ini antara lain boot speed akan terasa lebih cepat pada hampir semua komputer, dan terasa sangat cepat pada komputer berbasis SSD seperti netbook. Aplikasi ubuntu One yang disertakan pada versi ini berguna pada peningkatan integrasi desktop untuk online services serta sharing file dan folder via internet dengan lebih cepat dan mudah. Kernel Linux yang digunakan Lucid Lynx adalah versi 2.6.32-16.25 based on 2.6.32.9. Desktop environment menggunakan GNOME dan KDE. Selain itu dibenamkan pula HAL Removal yang membuat akselerasi dan boot Ubuntu semakin cepat. Indikator-indikator baru pada desktop juga banyak bermunculan pada versi ini.

- **Ubuntu 10.10 (Maverick Meerkat)**

  Canonical Ltd merilis versi ini pada tanggal 10 Oktober 2010. Ubuntu 10.10 memiliki dukungan multi-touch. Hal ini dimungkinkan melalui uTouch Ubuntu 1.0 dan multi-touch stack. Dengan Ubuntu 10.10, berbagai aplikasi GTK mendapat dukungan gesture-based scrolling. Selain itu, document viewer Ubuntu pun disempurnakan Canonical agar mendukung interaksi yang lebih baik. Untuk teknologi multi-touch baru, Canonical bekerja sama dengan X.Org dan komunitas kernel Linux, untuk menambahkan dukungan fitur yang belum ada atau memperbaiki driver yang telah ada. Tampilan graphical Maverick Meerkat sangat menawan, akselerasi ke menu-menunya sangat mudah dan cepat diseranimasi yang memukau. Diperkirakan versi ini akan menjadi batu lompatan generasi Linux Ubuntu selanjutnya menggunakan graphical sentuh serta dukungan penuh untuk multimedia mobile.

BAB III
PERANCANGAN SISTEM

Pada Bab ini menjelaskan tentang perancangan sistem *Intrusion Detection System* pada jaringan lokal. Topologi yang digunakan pada sistem ini adalah topologi Star.



Gambar III-1 Diagram Sistem Secara Umum

Berikut Keterangan dari sistem yang terlihat pada gambar III.1 :

1. Pada PC server IDS terdapat satu *ethernet card* dengan IP 192.168.0.2.
2. PC Server IDS dan client dihubungkan oleh sebuah *switch* agar semua perangkat dapat berkomunikasi.
3. Untuk PC penyerang terdapat pada satu jaringan lokal dengan alamat IP 192.168.0.15
4. Untuk monitoring dilakukan pada PC client atau PC Server IDS

3.1 Komponen Sistem

Untuk membangun *Intrusion Detection System*, diperlukan beberapa perangkat keras dan perangkat lunak yang akan digunakan.

3.1.1 Perangkat Keras

ada bagian ini akan menjelaskan perangkat kersa yang dibutuhkan untuk server PC IDS. Adapun perangkat yang dibutuhkan adalah sebagai berikut :

Tabel III-1 Kebutuhan Minimum Perangkat Keras Tablet

No	Perangkat Keras	Keterangan	Jumlah
1	Desktop Switch TP-LINK TL-SF1005D	Sebagai media penghubung	1 unit
2	PC Desktop	OS Ubuntu Desktop 14.04 LTS. Sebagai server Snort IDS	1 unit
3	Notebook Laptop Toshiba Satellite	OS Windows 8.1 64bit.	1 unit

3.1.2 Perangkat Lunak

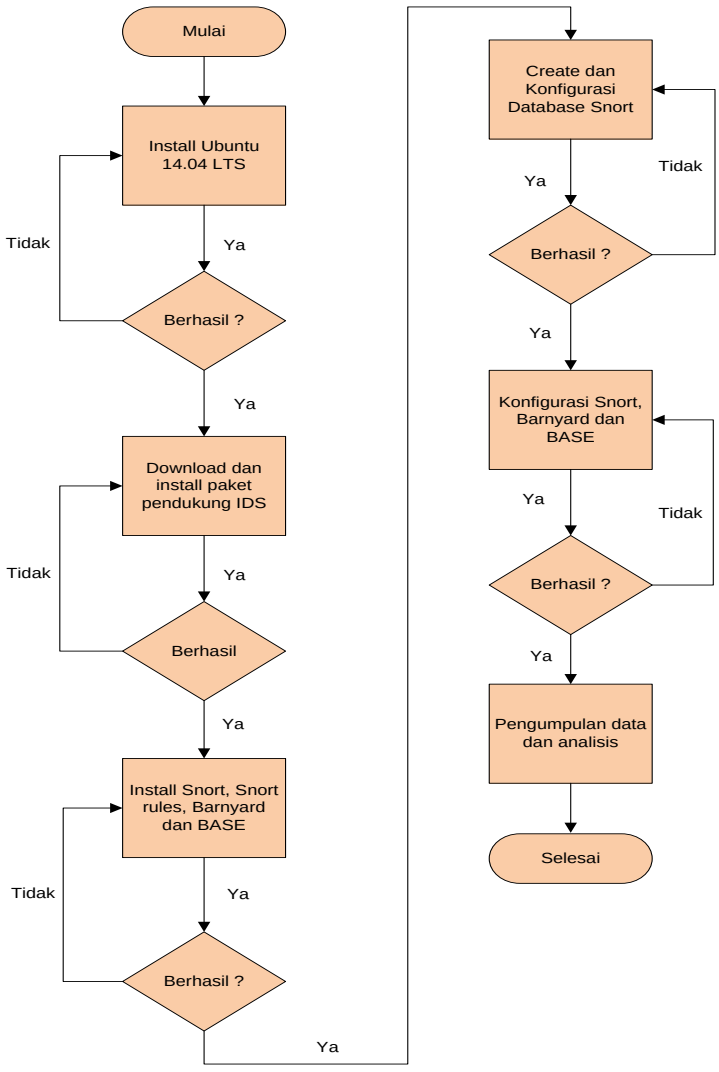
Pada bagian ini akan menjelaskan perangkat lunak yang digunakan dalam pembangunan Server Snort IDS. Adapun perangkat yang digunakan adalah sebagai berikut :

Tabel III-2 Perangkat Lunak Yang Digunakan

No	Nama	Keterangan
1	OS Ubuntu Desktop 14.04 LTS	Sebagai OS dari Server yang akan digunakan
2	Snort	Sebagai Program pendukung dari Sistem IDS
3	OS Windows 8.1	Sebagai OS untuk melakukan pengujian
4	Nmap	Sebagai Program pengujian

3.2 Perancangan Sistem

Agar sistem *Intrusion Detection System* (IDS) dapat berjalan dengan baik, diperlukan beberapa proses instalasi, seperti instalasi system operasi yang akan digunakan dan instalasi paket-paket pendukung yang dibutuhkan, seperti : Snort, Barnyard2 dan BASE. Proses-proses tersebut dapat diliat pada diagram alir dibawah ini.



Gambar III-2 Diagram Alir Perancangan Sistem

Berdasarkan pada gambar III.2 dapat dijelaskan sebagai berikut :

1. Pada peroses pertama yaitu diawali dengan instalsi sistem operasi pada PC. Sistem operasi yang digunakan pada PC adalah Linux Ubuntuk Desktop

14.04 LTS. Setelah selesai dan berhasil mengistal selanjutnya akan menginstal paket-paket pendukung untuk sistem.

2. Selanjutnya download dan install paket-paket pendukung dari IDS tersebut, seperti : Apache, PHP dan Mysql. Jika berhasil lanjut ketahap berikutnya.
3. Tahap berikutnya adalah installasi Snort, Barnyard2 dan BASE. Sebagai inti dari sistem yang akan kita bangun. Setelah berhasil melakukan installasi tersebut dilanjutkan ketahap berikut.
4. Membuat *database* dari snort, agar nantinya semua log dan sistem yang kita bangun bisa menyimpan semua kejadian kedalam database, setelah berhasil mebuat database lanjut ke proses berikutnya yaitu mengkonfigurasi database.
5. Setelah selesai membuat dan mengkonfigurasi database dilanjutkan ke konfigurasi Snort, Barnyard2 dan BASE.
6. Setelah semua proses selesai, tahap terakir adalah melakukan pengujian pada sistem dan pengumpulan data serta menganalisis.

3.3 Instalasi Sistem Operasi

Sistem Operasi yang digunakan pada PC Server IDS adalah Linux Ubuntuk Desktop 14.04 LTS. Sedangkan untuk *client* menggunakan sistem operasi windows 8.1 dan windows 7. Instalasi sistem operasi Linux dapat dilakukan dengan beberapa media, diantaranya melalui CDROM dan Flasdisk, pada instalasi kali ini akan digunakan Flaskdisk sebagai mana terlapir pada lampiran A.

3.4 Instalasi Paket Pendukung IDS

Untuk instalasi paket pendukung IDS seperi, Apache, PHP dan Mysql maka kita perlu untuk men-download paket tersebut dengan dengan cara, membuka Terminal console pada Ubuntu atau dengan cara menekan Ctrl + Alt + T, setelah jendela terminal muncul kita bisa menjalankan perintah berikut :

```
# sudo apt-get install apache2 libapache2-mod-php5 mysql-server mysql-common mysql-client php5-mysql libmysqlclient-dev php5-gd php-pear libphp-adodb php5-cli libwww-perl libnet1 libnet1-dev libpcre3 libpcre3-dev autoconf libcrypt-ssleay-perl libtool libssl-dev build-essential automake gcc make flex bison git
```

3.5 Instalasi Snort, Barnyard2 dan BASE

Setelah paket-paket pendukung tadi selesai di instal kita lanjutkan dengan menginstal Snort, Barnyard2 dan BASE untuk sistem yang akan kita bangun nanti.

Untuk proses awal instal Snort, kita ketikan perintah berikut di terminal Ubuntu :

```
# sudo apt-get install snort
```

Pada saat proses menginstal Snort akan keluar jendela untuk memasukan alamat IP yang akan kita lindungi. Untuk alamat IP yang akan di lindungi kita bisa memasukan alamat IP dari topologi yang kita bangun atau dengan mengetikan any.

Setelah selesai install Snort, dilanjutkan dengan mengintsall Barnyard2, dengan cara mengetikan perintah :

```
# cd /usr/local/src
# git clone  http://github.com/firnsy/barnyard2.git barnyard2
# cd barnyard2
# autoreconf -fvi -I ./m4
# ./configure -with-mysql -with-mysql-libraries=/usr/lib/i386-linux-gnu
# make
# make install
# cp etc/barnyard2.conf /usr/local/snort/etc
# mkdir /var/log/barnyard2
# chmod 666 /var/log/barnyard2
# touch /var/log/snort/barnyard2.waldo
# chown snort.snort /var/log/snort/barnyard2.waldo
```

Selanjutnya instal BASE yang nantinya akan di gunakan sebagai *interface* dari log serangan. Untuk instal BASE bisa menjalankan perintah berikut pada terminal Ubuntu :

```
# cd /var/www/html
# wget http://sourceforget.net/projects/secureideas/files/BASE/base-
  1.4.5/base-1.4.5.tar.gz
# tar zxvf base-1.4.5.tar.gz
# mv base-1.4.5 base
```

3.6 Membuat dan Konfigurasi Database Snort

Setelah selesai meng-install Snort, Barnyard2 dan BASE, dilanjut dengan membuat dan konfigurasi database untuk Snort, yang nantinya akan digunakan untuk menyimpan log dari setiap paket yang dibaca oleh IDS. Untuk membuat database tersebut dengan perintah:

mysql –u root –p (isi password untuk masuk kedalam database)

mysql > create database snortdb;

mysql> grant create, insert, select, delete, update on snortdb.* to snortuser@localhost identified ‘password’;

mysql > flush privileges;

mysql > exit;

3.7 Melakukan konfigurasi Snort, Barnyard2 dan BASE

Setelah selesai membuat database untuk Snort, dilanjutkan dengan mengkonfigurasi Snort, Barnyard2 dan BASE agar bisa berjalan dengan database.

3.7.1 Konfigurasi Snort

vim /etc/local/snort/etc/snort.conf

Mengubah value pada snort.conf

var WHITE_LIST_PATH /usr/local/snort/rules

var BLACK_LIST_PATH /usr/local/snort/rules

dynamicpreprocessor directory /usr/local/snort/lib/snort_dynamicprocessor/

dynamicengine /usr/local/snort/lib/snort_dynamicengine/libsf_engine.so

dynamicdetection directory /usr/local/snort/lib/snort_dynamicrules

output unified2: filename snort.log, limit 128

3.7.2 Konfigurasi Barnyard2

vim /usr/local/snort/etc/barnyard2.conf

Mengubah value seperti dibawah ini :

config reference_file: /usr/local/snort/etc/reference.config

config classification_file: /usr/local/snort/etc/classification.config

config gen_file: /usr/local/snort/etc/gen-msg.map

config sid_file: /usr/local/snort/etc/sid-msg.map

config hostname: localhost

config interface: eth0

output database: log, mysql, dbname=snortdb user=snortuser password=inter host=localhost

3.7.3 Konfigurasi BASE

Untuk mengkonfigurasi BASE, ketikan perintah dibawah ini pada terminal Ubuntu :

vim base_conf.php

Setelah itu isi setingan seperti dibawah ini :

$BASE_urlpath = '/base';

$DBlib_path = '/var/www/html/adodb5';

$alert_dbname = 'snortdb';

$alert_host = 'localhost';

$alert_port = '';

$alert_user = 'snortuser';

$alert_password = 'password'; (password sesuai dengan konfigurasi database)

Setelah semuanya berhasil di lakukan, restart Apache server dengan menjalankan perintah :

sudo service apache2 restart

3.8 Snort Rules

Untuk meng-edit dan mengaktifkan beberapa rules yang ada pada Snort, ketikan perintah berikut pada terminal *Console.* Sudo vim /etc/snort/rules

Berikut adalah beberapa contoh Rules dari Snort :

Vim /etc/snort/rules/snmp.rules (untuk rules snmp)

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP missing community string attempt"; content:"|04 00|"; depth:15; offset:5; reference:bugtraq,2112; reference:cve,1999-0517; classtyp$

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP null community string attempt"; content:"|04 01 00|"; depth:15; offset:5; reference:bugtraq,2112; reference:bugtraq,8974; reference$

alert udp $EXTERNAL_NET any -> $HOME_NET 161:162 (msg:"SNMP community string buffer overflow attempt"; content:"|02 01 00 04 82 01 00|"; offset:4; reference:bugtraq,4088; reference:bugtra$

alert udp $EXTERNAL_NET any -> $HOME_NET 161:162 (msg:"SNMP community string buffer overflow attempt with evasion"; content:" |04 82 01 00|"; depth:5; offset:7; reference:bugtraq,4088; re$

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP public access udp"; content:"public"; reference:bugtraq,2112; reference:bugtraq,4088; reference:bugtraq,4089; reference:cve,1999-05$

alert tcp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP public access tcp"; flow:to_server,established; content:"public"; reference:bugtraq,2112; reference:bugtraq,4088; reference:bugtraq$

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP private access udp"; content:"private"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:bugtraq,7$

alert tcp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP private access tcp"; flow:to_server,established; content:"private"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtr$

alert udp any any -> 255.255.255.255 161 (msg:"SNMP Broadcast request"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; reference:cve,2002$

alert udp any any -> 255.255.255.255 162 (msg:"SNMP broadcast trap"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; reference:cve,2002-00$

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP request udp"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; reference:cve,2002-0$

alert tcp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP request tcp"; flow:stateless; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; refe$

alert udp $EXTERNAL_NET any -> $HOME_NET 162 (msg:"SNMP trap udp"; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; reference:cve,2002-0013$

alert tcp $EXTERNAL_NET any -> $HOME_NET 162 (msg:"SNMP trap tcp"; flow:stateless; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-0012; referen$

alert tcp $EXTERNAL_NET any -> $HOME_NET 705 (msg:"SNMP AgentX/tcp request"; flow:stateless; reference:bugtraq,4088; reference:bugtraq,4089; reference:bugtraq,4132; reference:cve,2002-001$

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP PROTOS test-suite-req-app attempt"; content:"0&|02 01 00 04 06|public|A0 19 02 01 00 02 01 00 02 01 00|0|0E|0|0C 06 08|+|06 01 02 0$

alert udp $EXTERNAL_NET any -> $HOME_NET 162 (msg:"SNMP PROTOS test-suite-trap-app attempt"; content:"08|02 01 00 04 06|public|A4|+|06|"; reference:url,www.ee.oulu.fi/research/ouspg/proto$

vim /etc/snort/rules/icmp/rules (untuk Rules icmp)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP ISS Pinger"; itype:8; content:"ISSPNGRQ"; depth:32; reference:arachnids,158; classtype:attempted-recon; sid:465; rev:3;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP L3retriever Ping"; icode:0; itype:8; content:"ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"; depth:32; reference:arachnids,311; classtype:atte$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Nemesis v1.1 Echo"; dsize:20; icmp_id:0; icmp_seq:0; itype:8; content:"|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|"; depth:32; reference:arachnids,284; classtype:misc-activity; sid:1$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP icmpenum v1.1.1"; dsize:0; icmp_id:666 ; icmp_seq:0; id:666; itype:8; reference:arachnids,450; classtype:attempted-recon; sid:471$

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP redirect host"; icode:1; itype:5; reference:arachnids,135; reference:cve,1999-0265; classtype:bad-unknown; sid:472; rev:4;)

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP redirect net"; icode:0; itype:5; reference:arachnids,199; reference:cve,1999-0265; classtype:bad-unknown; sid:473; rev:4;)

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP superscan echo"; dsize:8; itype:8; content:"|00 00 00 00 00 00 00 00|"; classtype:attempted-recon; sid:474; rev:4;)

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute ipopts"; ipopts:rr; itype:0; reference:arachnids,238; classtype:attempted-recon; sid:475; rev:3;)

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP webtrends scanner"; icode:0; itype:8; content:"|00 00 00 00|EEEEEEEEEEEE"; reference:arachnids,307; classtype:attempted-recon; si$

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Source Quench"; icode:0; itype:4; classtype:bad-unknown; sid:477; rev:2;)

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Broadscan Smurf Scanner"; dsize:4; icmp_id:0; icmp_seq:0; itype:8; classtype:attempted-recon; sid:478; rev:3;)

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING speedera"; itype:8; content:"89|3A 3B|<=>?"; depth:100; classtype:misc-activity; sid:480; rev:5;)

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP TJPingPro1.1Build 2 Windows"; itype:8; content:"TJPingPro by Jim"; depth:32; reference:arachnids,167; classtype:misc-activity; si$

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING WhatsupGold Windows"; itype:8; content:"WhatsUp - A Netw"; depth:32; reference:arachnids,168; classtype:misc-activity; sid:4$

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING CyberKit 2.2 Windows"; itype:8; content:"|AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA|"; depth:32; reference:arachnids,1$

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Sniffer Pro/NetXRay network scan"; itype:8; content:"Cinco Network, Inc."; depth:32; classtype:misc-activity; sid:484; rev:4;)

#alert icmp any any -> any any (msg:"ICMP Destination Unreachable Communication Administratively Prohibited"; icode:13; itype:3; classtype:misc-activity; sid:485; rev:4;)

#alert icmp any any -> any any (msg:"ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited"; icode:10; itype:3; classtype:misc-activity; sid:486;$

#alert icmp any any -> any any (msg:"ICMP Destination Unreachable Communication with Destination Network is Administratively Prohibited"; icode:9; itype:3; classtype:misc-activity; sid:48$

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP digital island bandwidth query"; content:"mailto|3A|ops@digisle.com"; depth:22; classtype:misc-activity; sid:1813; rev:5;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Detection of a Denial of Service Attack"; dsize:>800; reference:arachnids,246; classtype:denial-of-service; sid:499; rev:4;)

BAB IV
PENGUJIAN DAN ANALISA

Bab ini membahas cara dan hasil pengujian sistem Snort IDS yang nantinya akan di tampilkan pada BASE, dalam hal ini yang menjadi objek penelitian adalah server Snort IDS, yaitu dengan mengirimkan paket gangguan kepada server Snort IDS dan hasil dari. Tujuannya adalah untuk mengetahui kelebihan dan kekurangan sistem yang telah dibuat apakah bekerja sesuai dengan apa yang direncanakan dan bekeja secara efektif.

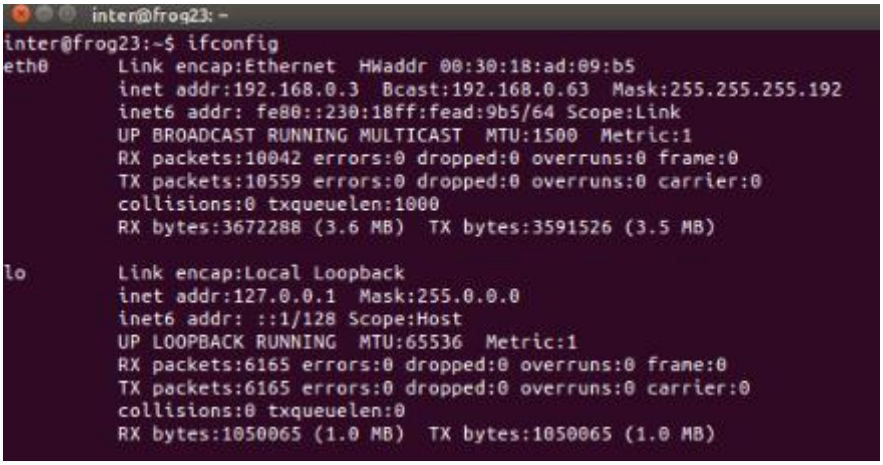
4.1 Persiapan Pengujian

Sebelum pengujian dilakukan ada beberapa tahap yang harus diperiksa sebagaimana dilampirkan pada bagian lampiran yaitu :

1. Memeriksa *interface* dan IP pada Server IDS

Sebelum pengujian dilakukan sebaiknya memeriksa *network interface* dan IP Address pada server IDS. Untuk melakukan pemeriksaan tersebut bisa menggunakan perintah :

```
#ifconfig
```

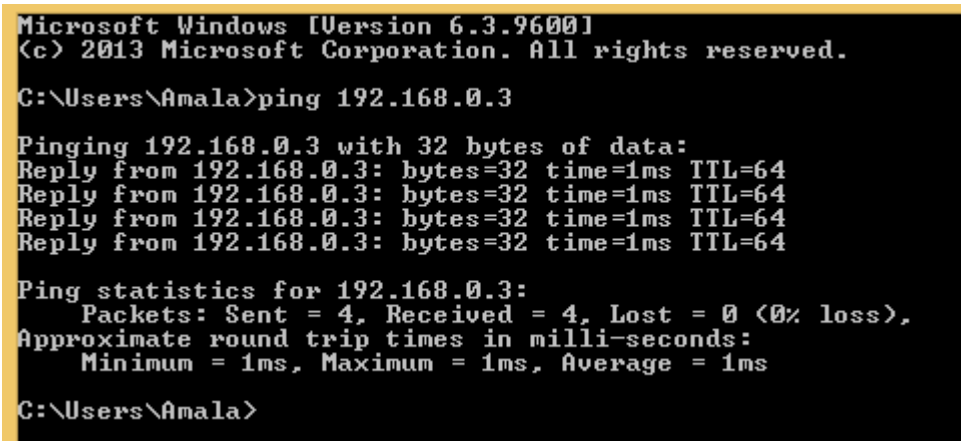


Gambar IV-1 Tampilan *Network Interface* Pada Server IDS

2. Tes koneksi pada jaringan

Setelah melakukan pemeriksaan *interface* dan IP Address pada server IDS, selanjutnya adalah memeriksa koneksi antar PC server dengan PC penggaggu, apakah terhubung atau tidaknya. Karena jika belum terhubung maka proses

pengujian gangguan terhadap server IDS tidak bisa dilakukan. Untuk melakukan pemeriksaan koneksi, bisa dilakukang dengan cara PING melalui terminal *console* ke perangkat yang akan diuji.

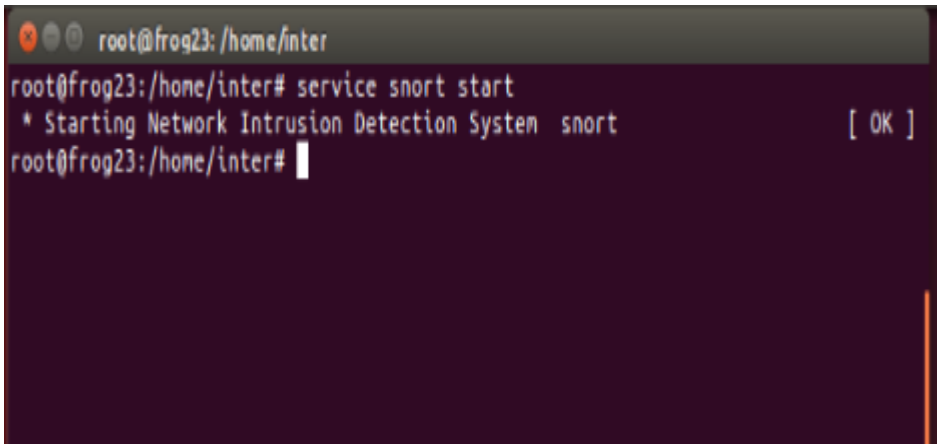


Gambar IV-2 Tampilan Pengujian Koneksi

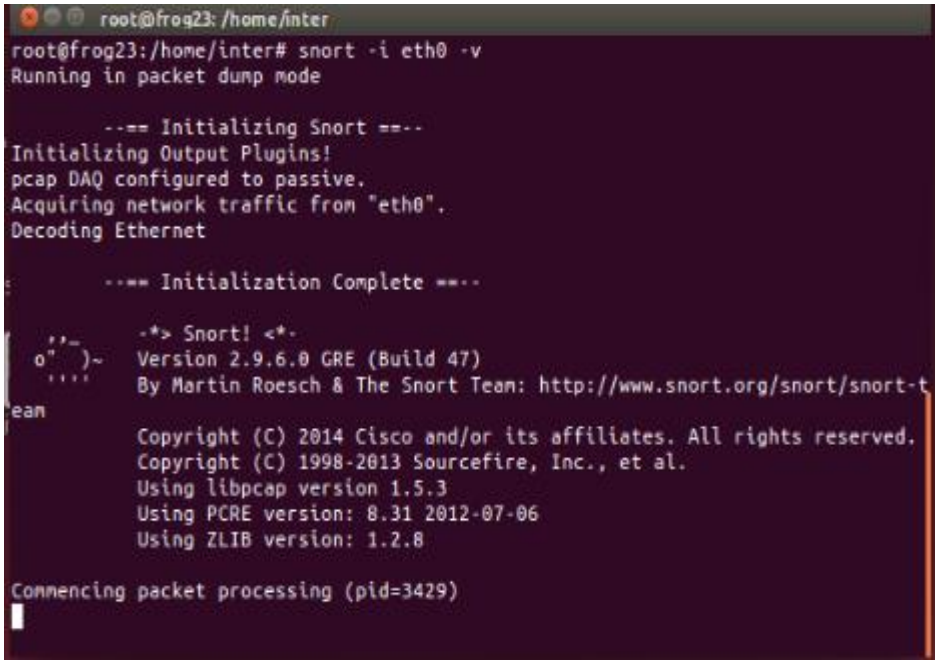
3. Menjalankan Snort IDS

Setelah Server IDS terhubung dengan PC penyerang pada jaringan selanjutnya mengecek apakah Snort IDS sudah berjalan dengan baik atau tidak pada Server IDS. Maka untuk melihat Snort IDS telah bekerja gunakan perintah di bawah ini pada terminal *console* :

```
# sudo service snort start
# sudo snort –i eth0 –v
```



Gambar IV-3 Tampilan Saat Menjalankan Snort IDS



Gambar IV-4 Tampilan Running Snort IDS

4.2 Tahap-Tahap Pengujian dan Pengambilan Data

Berikut pengujian dari sistem yang telah dirancang yaitu:

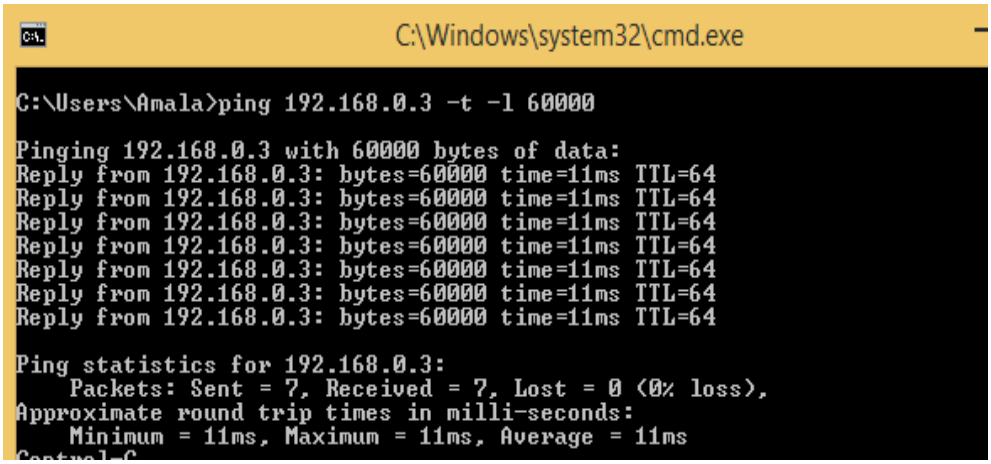
1. Dilakukan pengujian gangguan terhadap server dengan teknik DoS.
2. Dilakukan pengujian gangguan terhadap server dengan teknik *port scanner* menggunakan aplikasi Nmap untuk mengetahui *port* mana saja yang terbuka pada server.

4.3 Pengujian dan Pengambilan Data

Setelah sistem berjalan dengan baik selanjutnya adalah pengujian gangguan dan pengambilan data dari gangguan tersebut, berikut adalah hasil dari pengujian yang dilakukan.

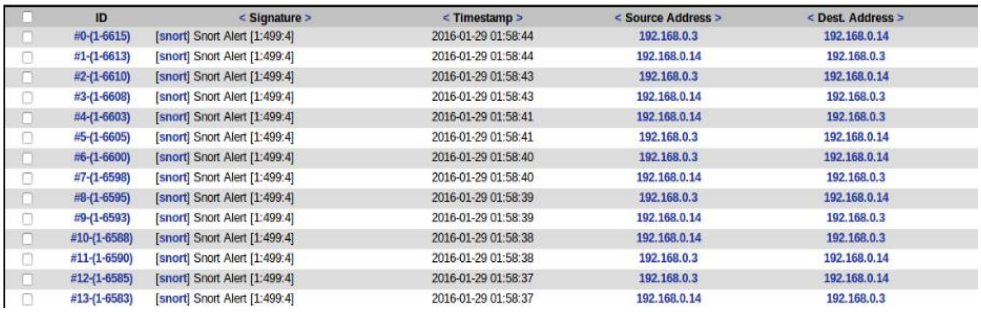
4.3.1 Pengujian Dengan Teknik Denial of Service (DoS)

Pada pengujian ini dilakukan dengan cara membanjiri atau melakukan pengiriman paket PING yang berukuran besar kepada server IDS.



Gambar IV-5 Tampilan Proses DoS

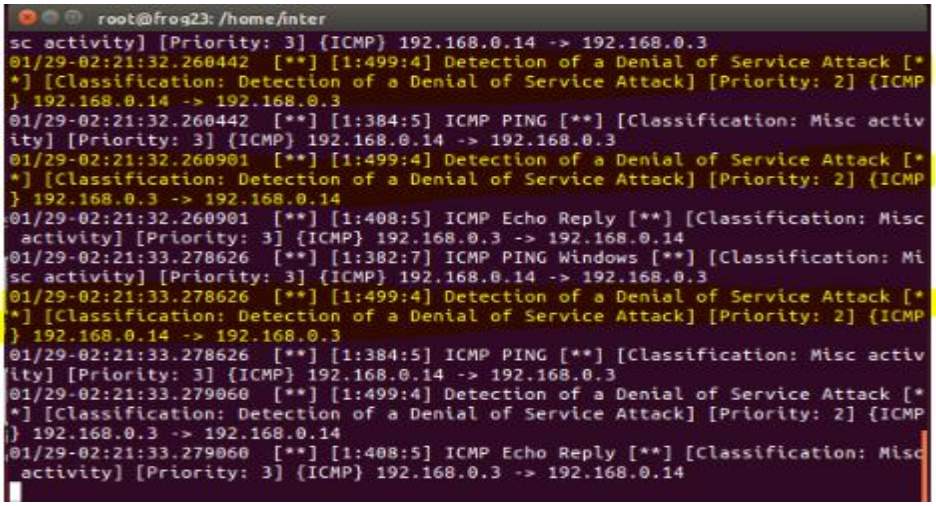
Dan berikut hasil tampilan log gangguan DoS pada *interface Basic Analysis and Security Engine* (BASE) dan log gangguan pada terminal *console*.



ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >
#0-(1-6615)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:44	192.168.0.3	192.168.0.14
#1-(1-6613)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:44	192.168.0.14	192.168.0.3
#2-(1-6610)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:43	192.168.0.3	192.168.0.14
#3-(1-6608)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:43	192.168.0.14	192.168.0.3
#4-(1-6603)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:41	192.168.0.14	192.168.0.3
#5-(1-6605)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:41	192.168.0.3	192.168.0.14
#6-(1-6600)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:40	192.168.0.3	192.168.0.14
#7-(1-6598)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:40	192.168.0.14	192.168.0.3
#8-(1-6595)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:39	192.168.0.3	192.168.0.14
#9-(1-6593)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:39	192.168.0.14	192.168.0.3
#10-(1-6588)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:38	192.168.0.14	192.168.0.3
#11-(1-6590)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:38	192.168.0.3	192.168.0.14
#12-(1-6585)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:37	192.168.0.3	192.168.0.14
#13-(1-6583)	[snort] Snort Alert [1:499:4]	2016-01-29 01:58:37	192.168.0.14	192.168.0.3

Gambar IV-6 Hasil *Log* DoS Pada BASE

Pada gambar IV.6 dapat dilihat log gangguang yang disebabkan oleh DoS ditampilkan ke *interface* BASE dengan signature Snort Alert dan TimeStamp atau waktu kejadian yang lengkap dengan tanggal dan waktu kejadian, BASE juga dapat menampilkan Source IP Address penggangu dan IP Address mana yang dituju.



Gambar IV-7 Hasil *Log* Pada Terminal *Console*

Sedangkan pada Gambar IV-7 menunjukan Log gangguan pada Terminal *Console* juga mendeteksi dan mencatat ada nya gangguan DoS, dengan menunjukan waktu dan tanggal kejadian, jenis gangguan, klasifikasi dari gangguan dan alamat IP dari komputer penggangu ke komputer tujuan.

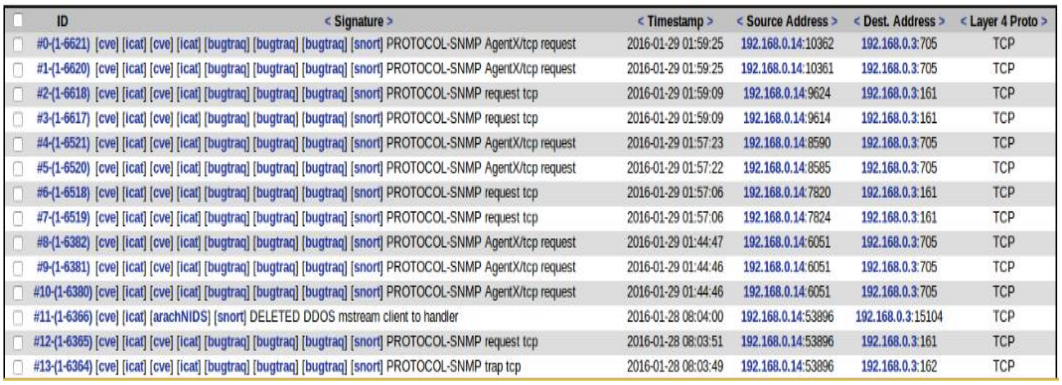
4.3.2 Pengujian Dengan Teknik *Port Scanner* Nmap

Pengujian berikutnya adalah dengan melakukan gangguan *port scanner* dengan menggunakan aplikasi Nmap, untuk mencari *port* mana saja yang sedang terbuka pada Server IDS.



Gambar IV-8 Tampilan Proses *Port Scanner* Menggunakan Nmap

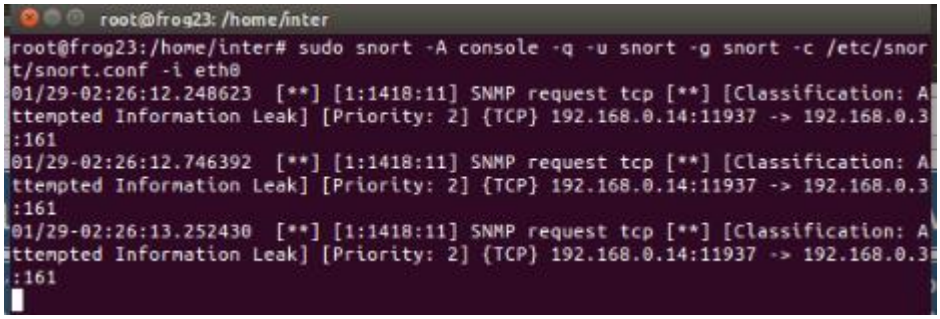
Dan berikut tampilan log gangguan *port scanner* Nmap pada BASE dan terminal *console* Server IDS.



ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-6621)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP AgentX/tcp request	2016-01-29 01:59:25	192.168.0.14:10362	192.168.0.3:705	TCP
#1-(1-6620)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP AgentX/tcp request	2016-01-29 01:59:25	192.168.0.14:10361	192.168.0.3:705	TCP
#2-(1-6618)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP request tcp	2016-01-29 01:59:09	192.168.0.14:9624	192.168.0.3:161	TCP
#3-(1-6617)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP request tcp	2016-01-29 01:59:09	192.168.0.14:9614	192.168.0.3:161	TCP
#4-(1-6521)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP AgentX/tcp request	2016-01-29 01:57:23	192.168.0.14:8590	192.168.0.3:705	TCP
#5-(1-6520)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP AgentX/tcp request	2016-01-29 01:57:22	192.168.0.14:8585	192.168.0.3:705	TCP
#6-(1-6518)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP request tcp	2016-01-29 01:57:06	192.168.0.14:7820	192.168.0.3:161	TCP
#7-(1-6519)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP request tcp	2016-01-29 01:57:06	192.168.0.14:7824	192.168.0.3:161	TCP
#8-(1-6382)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP AgentX/tcp request	2016-01-29 01:44:47	192.168.0.14:6051	192.168.0.3:705	TCP
#9-(1-6381)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP AgentX/tcp request	2016-01-29 01:44:46	192.168.0.14:6051	192.168.0.3:705	TCP
#10-(1-6380)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP AgentX/tcp request	2016-01-29 01:44:46	192.168.0.14:6051	192.168.0.3:705	TCP
#11-(1-6366)	[cve] [icat] [arachNIDS] [snort] DELETED DDOS mstream client to handler	2016-01-28 08:04:00	192.168.0.14:53896	192.168.0.3:15104	TCP
#12-(1-6365)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP request tcp	2016-01-28 08:03:51	192.168.0.14:53896	192.168.0.3:161	TCP
#13-(1-6364)	[cve] [icat] [cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] PROTOCOL-SNMP trap tcp	2016-01-28 08:03:49	192.168.0.14:53896	192.168.0.3:162	TCP

Gambar IV-9 Log *Port Scanner* Pada *Interface* BASE

Pada gambar IV.9 dapat dilihat tampilan log gangguan pada BASE mencatat hasil dari *Port Scanner* yang dilakukan Nmap dimana pada *Signature* BASE menujukan adanya PROTOCOL-SNMP AgentX/tcp request, PROTOCOL-SNMP request tcp dan DELETED DDOS mnstream client to handler. Dan *Timestamp* pada BASE akan menunjukan tanggal dan waktu dari kejadian yang di baca oleh Snort IDS, sedangkan untuk *Source Address* dan *Dest. Address* merupakan tampilan dari IP penggagu (Source Address) ke IP komputer tujuan (Dest Address), bahkan BASE juga menampilkan pada layer mana gangguan tersebut masuk.



Gambar IV-10 Log *Port Scanner* Pada Terminal *Console*

Sedangkan pada gambar IV.10 menunjukan tampilan log gangguan pada terminal *console* dimana juga terdapat waktu dan tanggal kejadian dan jenis gangguan yang di munculkan oleh Nmap, beserta klasifikasi dari gangguan tersebut dan alamat IP sumber ke tujuan.

4.4 Analisa

Dari dua hasil gangguan yang dibaca oleh Server IDS, dapat dilihat gangguan DoS dan *port scanner* yang dikirimkan kepada server IDS dapat dibaca oleh Snort IDS dengan sangat baik, bahkan Snort IDS bisa mencatat kapan waktu gangguan dan alamat dari IP Address tersebut dan menampilkannya pada terminal *console* dan pada *interface* BASE.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan dan pengujian yang telah dilakukan terhadap Server Snort IDS maka dapat diambil kesimpulan terhadap rancangan Server IDS adalah sebagai berikut:

1. IDS yang dirancang untuk mendeteksi adanya gangguan terhadap suatu jaringan dapat dibaca dengan baik oleh Snort IDS.
2. Snort IDS yang diinstal pada sistem operasi Linux Ubuntu Desktop 14.04 LTS dapat berjalan baik dan dapat digunakan dalam pemantauan jaringan.
3. Dari hasil pengujian gangguan terhadap Server Snort IDS. Snort IDS dapat mengenali jenis gangguan yang ditimbulkan dan dapat menampilkan secara cepat dan tetap kapan terjadinya gangguan dan dari mana asal gangguan tersebut.

5.2 Saran

Saran yang diajukan untuk pengembangan selanjutnya dan melengkapi kekurangan-kekurangan adalah sebagai berikut :

1. Adanya penanggulangan secara langsung terhadap gangguan yang masuk atau yang terbaca oleh Snort IDS.
2. Dilakukannya pengujian gangguan terhadap jaringan yang berbeda, tidak hanya pada dalam satu jaringan saja

DAFTAR PUSTAKA

[1] Junior, A. N. S., Harianto, Agus dan Alexander., 2009, Perancangan dan Implementasi Intrusion Detection System pada Jaringan Nirkabel BINUS University, 23506011.

[2] Forouzan, Behrouz A., Data Communication and Networking, Mc-Graw Hill, Fourth Edition, 2007.

[3] Veizi, Rando., 2013. IDS/IPS: Intrusion Detection/Prevention System

[4] Rafeeq Ur , Rehman., Intrusion Detection Systems with Snort, (Online), diakses pada tanggal 26 Maret 2015 dari word wide web : http://ptgmedia.pearsoncmg.com/images/0131407333/downloads/0131407333.pdf

[5] www.manual.snort.org.