JETri, Volume 11, Nomor 1, Agustus 2013, Halaman 1 - 16, ISSN 1412-0372

PERANCANGAN DAN ANALISIS SISTEM PENDETEKSI INTRUSI BERBASIS *NETWORK INTRUSION DETECTION SYSTEM* (NIDS) PADA SISTEM KEAMANAN JARINGAN KOMPUTER

Ferrianto Gozali & Achmad Lucky Setiaji***

(*) Dosen Jurusan Teknik Elektro, FTI Universitas Trisakti

(**) Alumni Jurusan Teknik Elektro, FTI Universitas Trisakti

Abstract

Network intrusion detection system is a system that can detect illegal accesses or intrusions happened in a computer network. Actually, there are many types of intrusion detection systems and the differences are based on how network administrators implement the system to secure the network. In this study, the system called Network Intrusion Detection System or NIDS in brief, is used to design and implement the intrusion detection system in the network model design. Intrusion detection system will utilize the snort application that serves as a sensor and detection server and be implemented in a network model that has been designed previously. The performance of the system is investigated through the monitoring of the use of disk space, memory usage and cpu usage of the system during intrusion detection identification process. Three different intrusion scenarios such as Portscanning, ICMP flooding and SYN flooding is performed to see the effect in the system's performance. During the test, the use of disk space still has not shown any significant use, due to the detection time limit used in this study was too short. However, the difference in memory usage and cpu is clearly visible, when detecting intrusion object.

Keywords: *network, intrusion, detection, snort, flooding.*

1. PENDAHULUAN

Berdasarkan data yang diperoleh dari *Internet World Stats* [www.internetworldstats.com], di Indonesia terjadi peningkatan pengguna internet yang sangat signifikan, dari sekitar 2 juta pengguna pada tahun 2000 menjadi hampir 55 juta di akhir tahun 2011. Hal ini menunjukkan makin meningkatnya kesadaran masyarakat akan pentingnya dan manfaat yang diperoleh.. Peningkatan pengguna internet berkaitan langsung dengan peningkatan akses pada sistem jaringan menyediakan pelayanan pada pengguna jaringan. Hal ini berarti dibutuhkannya suatu

sistem yang dapat mengendalikan penyediaan layanan yang disediakan seiring dengan peningkatan aksesibilitas penggunaan jaringan tersebut.

Dalam mengontrol kebutuhan jaringan, diperlukan suatu sistem yang dapat mengatur dan mengawasi aksesibilitas yang terjadi pada jaringan. Salah satu cara yang dilakukan adalah dengan menggunakan suatu sistem keamanan yang mampu berperan aktif untuk mengawasi akses pengguna pada jaringan yang disebut *Firewall*. *Firewall* merupakan salah satu sistem keamanan jaringan yang berfungsi dalam mengawasi lalu lintas data suatu jaringan, baik komunikasi yang terjadi secara global maupun lokal di dalam suatu jaringan.

Namun *firewall* sangat rentan akan usaha penyusupan (intrusi) yang dilakukan oleh pengguna yang tidak punya hak akses atau *intruder*, yang menyebabkan sistem jaringan tidak dapat menjalankan tugas pelayanan terhadap pengguna dengan optimal [Stalling, 2007: 703-732]. Oleh sebab itu dibutuhkan sistem lain yang dapat mendukung kinerja pada sistem keamanan tersebut.

Pada penelitian merancang dan menganalisis suatu perangkat pendeteksi yang digunakan pada sistem kemanan jaringan dalam membantu mendeteksi penyusupan atau intrusi di dalam jaringan tersebut. Pengukuran dilakukan dengan melakukan berbagai skenario intrusi untuk melihat kemampuan sistem pendeteksi serta pengaruhnya terhadap unjuk kerja dari sistem jaringan.

2. NETWORK INTRUSION DETECTION SYSTEM

Salah satu tipe intrusion detection system adalah *network-based intrusion detection system*. *Network-based intrusion detection system* bekerja berdasarkan analisis yang dilakukan terhadap lalu lintas paket data pada suatu jaringan [Proctor, 2001: 203-261]. Paket data yang diperoleh berasal dari satu lingkup jaringan dan dapat berasal dari keluaran router ataupun switch. Umumnya proses yang dilakukan adalah menangkap dan menganalisis protokol TCP/IP pada paket data tersebut, akan tetapi protokol jaringan lainnya juga dapat digunakan sebagai proses analisis.

Seringkali serangan yang ditujukan pada suatu jaringan, secara langsung menuju kelemahan pada sistem operasi yang terdapat pada server jaringan dan dapat

dimanfaatkan untuk tujuan kegiatan ilegal seperti akses ilegal, pencurian sumber informasi/data atau bahkan *denial of service*. Sebuah *network intrusion detection system* (NIDS) hanya melakukan seperti fungsi utamanya, yaitu sebagai sistem pendeteksi penyusupan (intrusi) pada jaringan. Rumitnya proses analisis pada *network intrusion detection system*, disebabkan oleh parameter nilai yang digunakan oleh sistem dan selalu bergantung pada kemampuan dari administrator jaringan yang menjalankanya NIDS tersebut. Beberapa sistem deteksi yang baik membutuhkan operator jaringan yang memahami betul jaringan yang dikelolanya.

Mode operasional yang disediakan pada NIDS menjelaskan cara untuk mengoperasikan *network intrusion detection system* dan menjelaskan tujuan akhir dari pendeteksian. Beberapa hal penting dalam mode operasional yang digunakan pada *network intrusion detection system* [Rehman, 2003: 140-155] antara lain;

1. Peringatan (*Tip-Off)*: Sistem digunakan untuk mendeteksi penyusupan (intrusi) yang telah terjadi. Dengan mengamati pola dari perilaku dan aktivitas yang mencurigakan mungkin terdeteksi sebagai peringatan kepada operator jika penyalahgunaan memungkinkan terjadi. Dengan mendefinisikan karakteristik dari peringatan kepada sistem, yang memungkinkan terdeteksi secara tidak terduga sebelumnya.
2. Pengawasan: Selama pengawasan, target diamati secara dekat untuk melihat pola dari penyusupan (intrusi). Karakteristik pengawasan yaitu dengan upaya peningkatan dalam mengamati perilaku dari bagian terkecil dari subjek yang di awasi. Tidak seperti peringatan, pengawasan memiliki peran ketika penyalahgunaan menunjukkan atau diduga terjadi. Pengawasan selalu mengikuti peringatan baik dari intrusion detection system atau beberapa petunjuk lainnya.
3. Peralatan Forensik (*digital forensic*): Sebuah *network intrusion detection system* juga dapat digunakan sebagai peralatan forensik dalam menganalisis lalu lintas paket data di dalam jaringan. Peralatan yang sama dalam mengawasi dapat juga digunakan sebagai analisis lalu lintas untuk mencari kecenderungan dan petunjuk lainnya. Fungsi lain adalah sebagai berikut:

- Memonitor catatan transaksi secara *online.*
- Melacak pertumbuhan jaringan.
- Menghasilkan laporan detail kerusakan dari pelayanan yang digunakan pada jaringan.
- Mengidentifikasikan perubahan yang tidak terduga pada jaringan.

Pada kenyataannya, tantangan teknis yang dihadapi sistem deteksi tersebut dalam waktu dekat akan melumpuhkan kemampuannya dalam memberikan hasil analisis dan deteksi yang signifikan terhadap parameter nilai. Kuncinya adalah keseimbangan dalam melakukan perancangan dan kebijakan terhadap sistem deteksi yang digunakan [Gullet, 2012].

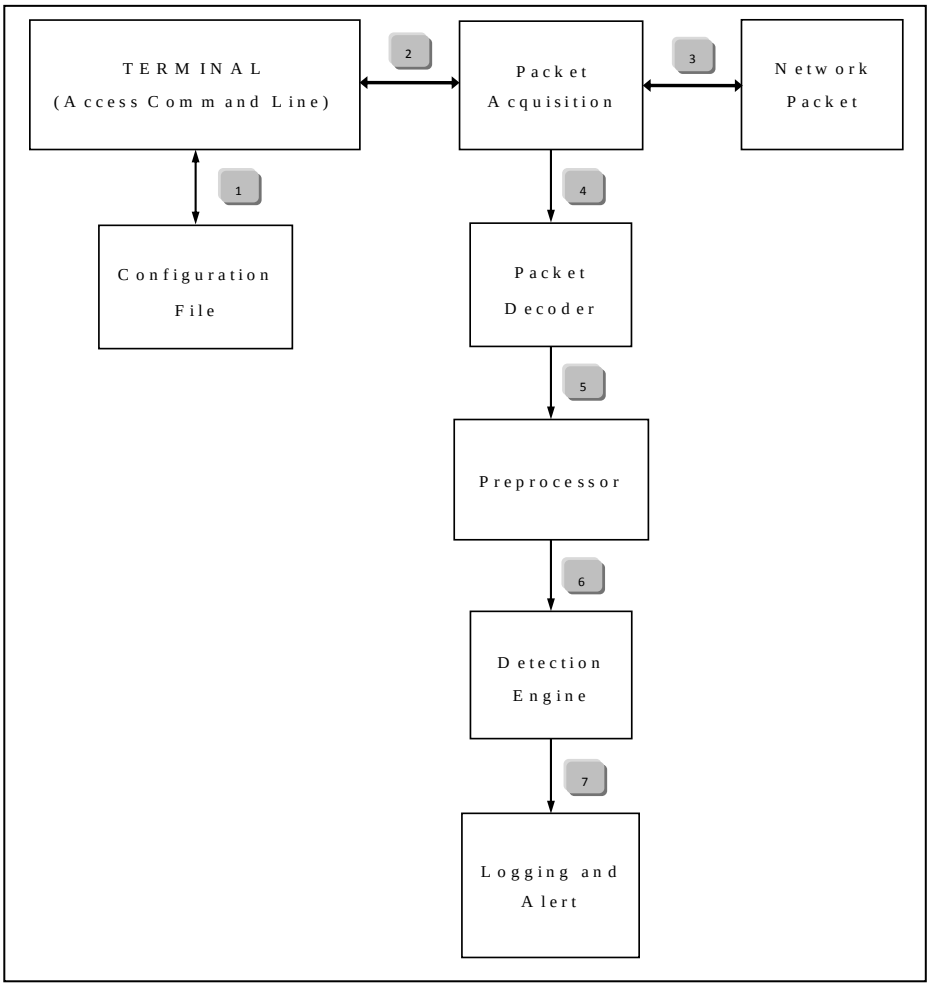
3. METODOLOGI

3.1. Perancangan Network Intrusion Detection System

Network intrusion detection system merupakan suatu sistem yang berfungsi dalam melakukan memonitor *segment* atau *subnet* di dalam suatu jaringan [Gullet, 2012]. Salah satu aplikasi sumber terbuka sebagai *network intrusion detection system* yang digunakan dalam penelitian ini adalah *snort*. *Snort* merupakan sebuah aplikasi keamanan yang berfungsi untuk mengawasi dan memonitor bentuk-bentuk penyalahgunaan (*intrusion*) pada jaringan dan juga dapat dikonfigurasikan untuk melakukan pencegahannya [Snort Team, 2012: 11-13]. Memiliki kemampuan dalam analisis *packet* jaringan dan menyimpan bentuk dari *log* secara *real-time* berbasiskan TCP/IP. Secara umum, *snort* mempunyai tiga fungsi utama, yaitu sebagai *packet sniffer*, *packet logger* dan *network intrusion detection system* (NIDS). Arsitektur dari *Snort* dapat dilihat pada Gambar 1 pada halaman berikut.

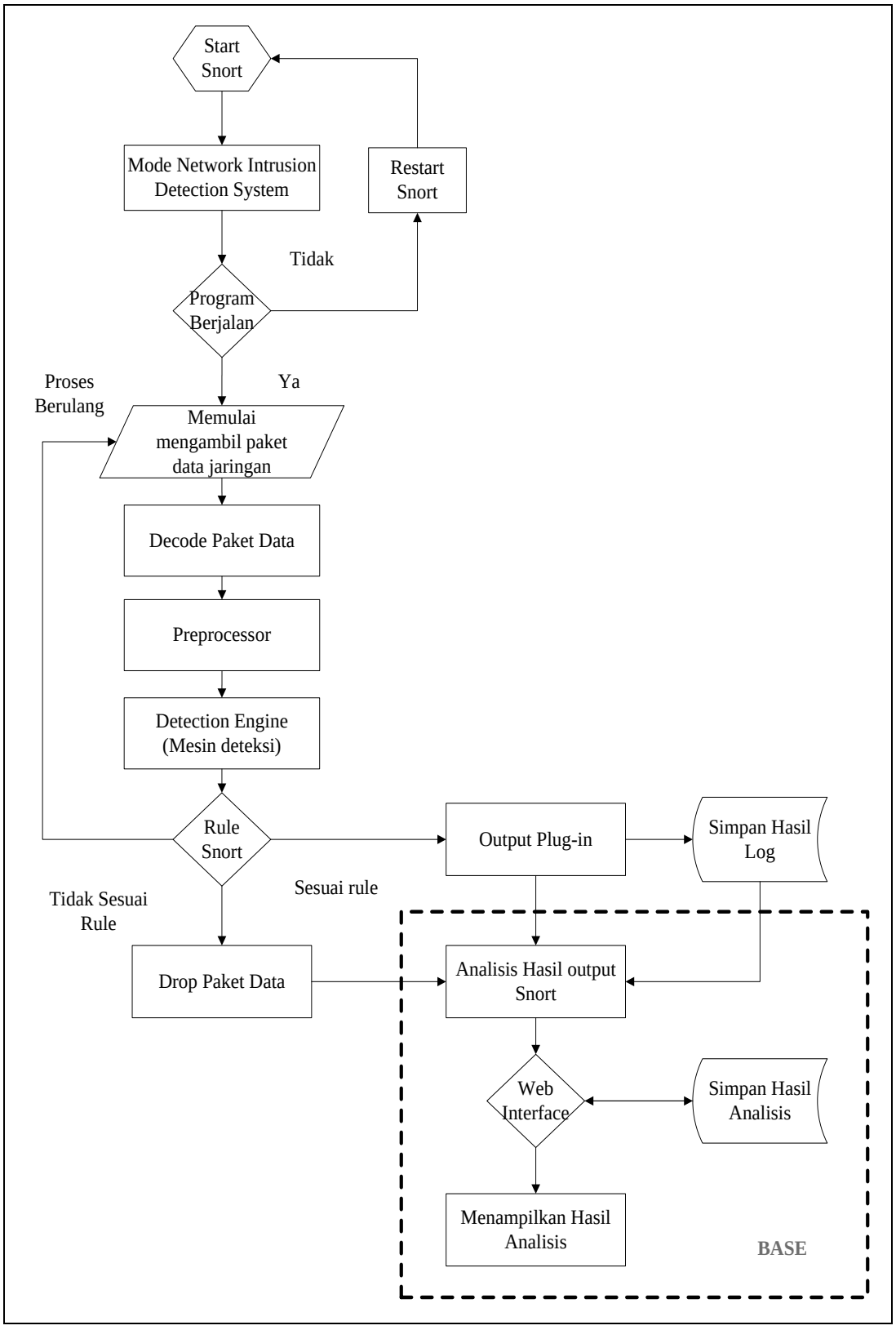
Pada proses 1, yaitu proses akses user pada *snort* melalui *console,* antara lain adalah mengkonfigurasikan mode dengan file konfigurasi (*rule*, *preprocessor* dan *output* pada *plug-in*). Pada proses 2, menjalankan beberapa proses awal pada *snort,* yaitu memulai proses kerja mesin deteksi dan *library* pada *Libpcap*. Pada proses 3, menerima paket yang berasal dari jaringan dan menggunakan *cross platform library* pada *Libpcap* yang memungkikan API untuk dapat menerima paket

yang langsung berasal dari jaringan yang diawasi. Pada proses 4, yaitu proses dalam memisahkan (*decode*) bagian-bagian yang penting pada paket jaringan, agar proses *preprocessor* dapat dilakukan. Pada proses 5, yaitu melakukan normalisasi terhadap paket jaringan yang sudah di *decode* dan dapat dilakukan konfigurasi terhadap *preprocessor* agar dapat melakukan deteksi tanpa digunakannya *rule*. Pada proses 6, yaitu melakukan evaluasi *rule-rule* terhadap paket jaringan. Pada proses 7, setelah *rule-rule* tersebut *match* (terjadi *hit*) maka sistem akan memberi peringatan (*alert*) dan *log*.



Gambar 1. Arsitektur s*nort.*

Sedangkan tahapan proses yang berlangsung pada *snort* digambarkan dalam bentuk diagram alir seperti pada Gambar 2.



Gambar 2. Flowchart n*etwork intrusion detection system*

Untuk melakukan proses deteksi, administrator akan mengaktifkan mode *network intrusion detection system* pada *snort* dengan menggunakan konsol pada terminal. *Administrator* menggunakan perintah yang disesuaikan dengan file *rule* dan target yang dideteksi. *Snort* akan menjalankan mode *network intrusion detection system*, melakukan pengambilan paket yang melalui *snort*, melakukan proses dekoding, melakukan penyamaan *rule* dengan paket jaringan yang tertangkap dan menyimpan hasil pendeteksi pada basis data MySQL. Hasil tersebut nantinya akan dianalisis oleh BASE (Gambar 2) yang memungkinkan memudahkan dalam analisis yang dilakukan dan memberikan laporan (hasil) deteksi pada *snort*.

3.2. Spesifikasi Perangkat

Perangkat keras yang digunakan sebagai sistem deteksi intrusi dalam perancangan NIDS (*network intrusion detection system*) memiliki spesifikasi:

- *Processor* : Intel® Core 2 Duo Processor T5870, 2,0 GHz , 2 MB L2 Cache 800 MHz FSB
- *Memory* : 2048 MB DDR2 800 MHz
- *Hard disk* : 250 GB, 5400 rpm

Perangkat keras yang digunakan sebagai *server* jaringan (*network management*) yang berfungsi sebagai *router*, *gateway* dan *firewall* memiliki spesifikasi:

- *CPU* : *AMD Geode LX800-500 MHz processor*
- *Memory* : *1 x DDR DIMM 400MHz w/o ECC registered Up to 1G with 1 slot*
- *Ethernet Adapter*: 410/100 *Mbps (Realtek® 8139CL+)*
- *Storage* : *Compact Flash 1 x CompactFlash™ Type II Socket*

Sistem operasi yang digunakan oleh sistem deteksi intrusi yang akan di *install* pada perangkat keras (*hardware*) adalah Linux Ubuntu. Ubuntu memiliki dukungan baik yang berasal dari komunitas Ubuntu dan profesional. Sistem operasi linux Ubuntu yang digunakan yaitu Ubuntu 12.04 (*Precise Pangolin*) LTS (*Long Term Services*). Ubuntu versi ini, akan memberikan dukungan terhadap sistem

keamanan dan perbaikan lainnya hingga April 2014. Perangkat lunak untuk membentuk NIDS sebagai sistem deteksi intrusi adalah sebagai berikut:

1. *Snort.* Telah dijelaskan sebelumnya, *Snort* merupakan sebuah aplikasi *open source* yang digunakan sebagai *network intrusion detection system* yang dikembangkan oleh *Sourcefire*. Memiliki kelebihan dalam kombinasi dari *signature*, protokol dan pemeriksaan berbasiskan anomali.
2. *Libpcap*. Pada bidang komputer administrasi jaringan, *pcap* (*packet capture*) terdiri dari *application programming interface* (API) untuk menangkap paket jaringan. tidak seperti sistem Unix dalam mengimplementasikan pcap pada *Libpcap library.*
3. *PCRE*. *Perl Compatible Regular Expression* (PCRE) sebuah pengungkapan dasar dalam *library* C yang terinspirasi oleh *Perl external interface,* yang dikembangkan oleh Philip Hazel.
4. *Libdnet*. *Libdnet* merupakan API dasar pada jaringan yang memiliki akses ke beberapa protokol.
5. *Barnyard2*. *Barnyard* merupakan sistem *output* yang digunakan pada *snort. Snort* menciptakan keluaran *binary* yang unik dinamakan "*Unified*". *Barnyard2* membaca file ini dan kemudian mengirimkan kembali data menuju *back-end* pada basis data.
6. *DAQ*. *DAQ* merupakan API *data aquissition* yang dibutuhkan oleh *snort* pada versi 2.9.0 dan selanjutnya.
7. *Base*. Merupakan aplikasi analisis dan perangkat keamanan. Aplikasi ini dikembangkan berdasarkan pada projek *analysis console for intrusion* (ACID).

Aplikasi ini menyediakan *front-end* berupa *web-base* yang memudahkan dalam pengaturannya dan pemeliharaaannya dalam melakukan analisis terhadap intrusi yang terdeteksi oleh *snort.*

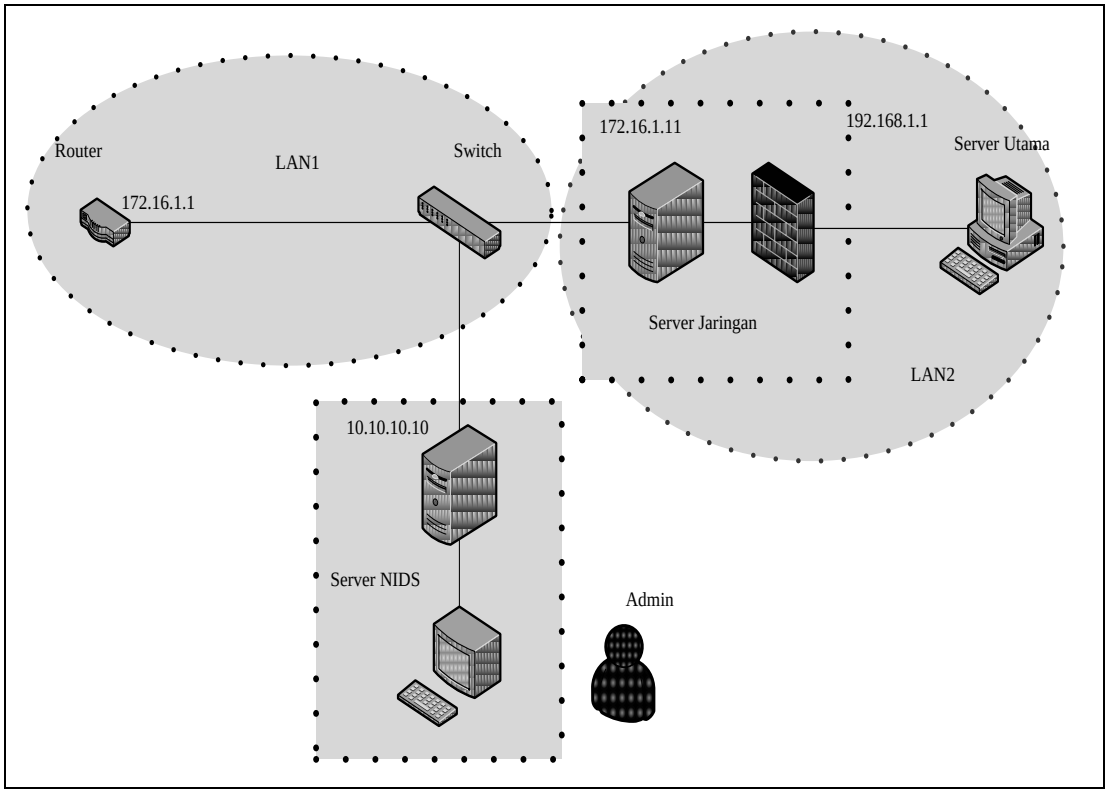
Untuk sistem operasi yang digunakan sebagai *router* adalah sistem operasi *ZeroShell*. *Zeroshell* adalah salah satu dari distro atau distribusi sistem operasi yang berbasis *linux kernel* yang dirancang untuk *server* dan perangkat *embedded* untuk layanan utama yang dibutuhkan oleh Jaringan Komputer.

Berikut kelebihan yang dimiliki oleh sistem operasi *ZeroShell*:

1. *Load Balancing* dan *Failover* dari beberapa koneksi internet.
2. *Captive Portal* sebagai tempat pengguna jaringan *login* ke jaringan komputer melalui web, *zeroshell* difungsikan sebagai *gateway* dimana *Captive Portal* diaktifkan dan dimana alamat IP diberikan secara dinamis melalui DHCP, pengguna yang ingin masuk ke jaringan ini diwajibkan melakukan autentikasi *username* dan *password* melalui *web browser.*
3. QoS (*Quality of Service*) untuk mengontrol lalu lintas jaringan.
4. HTTP *transparent proxy server.*
5. *Router* dengan *route* statis dan dinamis.
6. *Firewall Packet Filter* dan *Stateful Packet Inspection* (SPI) dengan filter yang berlaku pada *routing* atau *bridging* pada semua jenis antarmuka jaringan termasuk VPN dan VLAN.
7. *Network Address Translation.*
8. DNS *Server.*
9. *Multi Subnet* DHCP *Server* dengan kemampuan untuk memperbaiki alamat IP dari alamat MAC klien.

3.3. Implementasi *Snort* Sebagai *Network Intrusion Detection System*

Hal pertama kali yang dapat dilakukan setelah perancangan terhadap sistem pendeteksi intrusi, yaitu dengan melakukan implementasi dengan tahap pertama yaitu instalasi sistem operasi dan perangkat lunak yang digunakan dalam membentuk sistem pendeteksi intrusi, baik perangkat keras yang digunakan oleh sistem maupun perangkat keras yang digunakan oleh perangkat jaringan. Langkah kedua setelah melakukan instalasi sistem operasi dan perangkat lunak, adalah melakukan konfigurasi yang dibutuhkan untuk menyesuaikan dengan proses pengujian yang akan dilakukan. Model jaringan yang digunakan dalam uji coba ini, merupakan jaringan dengan topologi bintang sedangkan penjelasan detail tentang konfigurasi jaringan yang digunakan pada uji coba tersebut dapat dijumpai pada penjelasan setelah gambar. Gambar 3 merupakan diagram topologi jaringan yang digunakan.

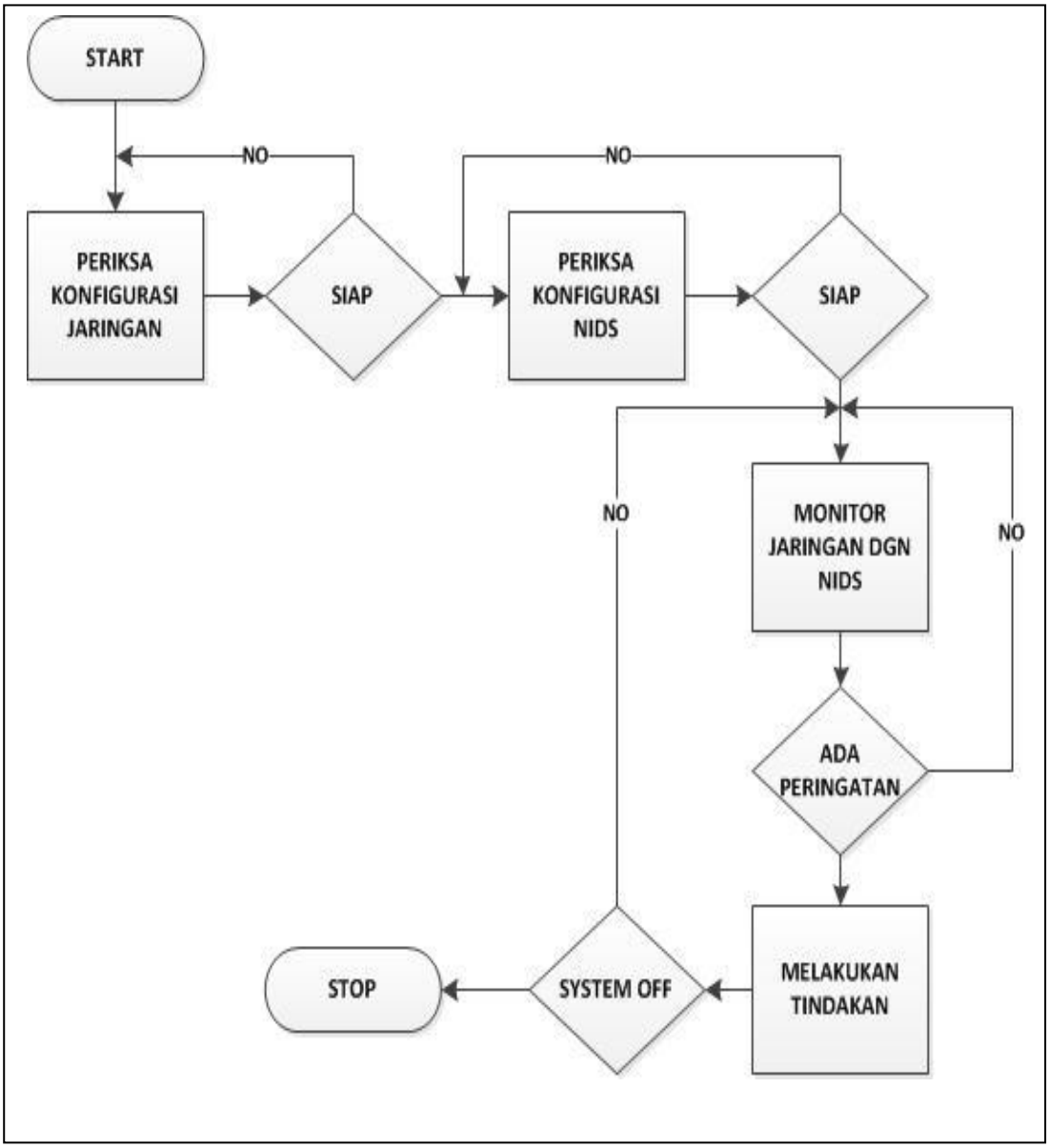


Gambar 3. Topologi jaringan sistem deteksi intrusi

Konfigurasi model jaringan pada Gambar 3 adalah sebagai berikut:

a. Menggunakan 2 jaringan *local area network*, dimana jaringan *local area network* yang pertama (LAN1) berfungsi sebagai area *external* (luar) dan jaringan *local area network* yang kedua (LAN2) berfungsi sebagai jaringan *internal* (dalam) dilengkapi dengan sistem keamanan jaringan yaitu *firewall*.
b. Menggunakan sebuah *server* yang dikonfigurasikan sebagai *network intrusion detection system* (NIDS), yang memiliki kemampuan sebagai sensor, penyimpanan *log* ketika melakukan proses deteksi.
c. Menggunakan sebuah *server* lainnya yaitu sebagai *server* jaringan (*network management*) dengan menggunakan *single board compute*r.
d. Sebuah *host* sebagai penyerang dengan IP yang tidak diketahui yang berasal dari LAN1.
e. Sebuah *server* dan *host* sebagai target pada *local area network* (LAN2) yang berbeda.

Gambar 4 merupakan diagram alir cara kerja sistem deteksi intrusi.



Gambar 4. Diagram alir cara kerja sistem deteksi intrusi.

Dari Gambar 4 diatas terlihat tahapan yang dilakukan adalah:

- Pertama kali memulai menjalankan sistem jaringan yang dilengkapi dengan NIDS adalah dengan menjalankan *server* jaringan. *Server* jaringan berfungsi sebagai *gateway*, *router* dan *firewall*. Mengakses *server* jaringan secara *local* dengan menggunakan *user interface* berbasiskan *web*.

- Melakukan konfigurasi dan melakukan pemeriksaan koneksi (hubungan) dengan *switch*.
- Setelah siap (setelah melakukan pengetesan terhadap *server* jaringan), tahap berikutnya adalah menjalankan *server* NIDS.
- Memeriksa konfigurasi yang dibutuhkan (dijelaskan pada 3.3) dan melakukan uji coba.
- Setelah siap, jalankan kembali aplikasi *snort* pada *server* NIDS melalui *command prompt* untuk melakukan *test* awal setelah konfigurasi sebelum melakukan pengujian.

4. HASIL PENGUJIAN DAN ANALISIS

Proses Pengujian dan analisis yang dilakukan adalah sebagai berikut:

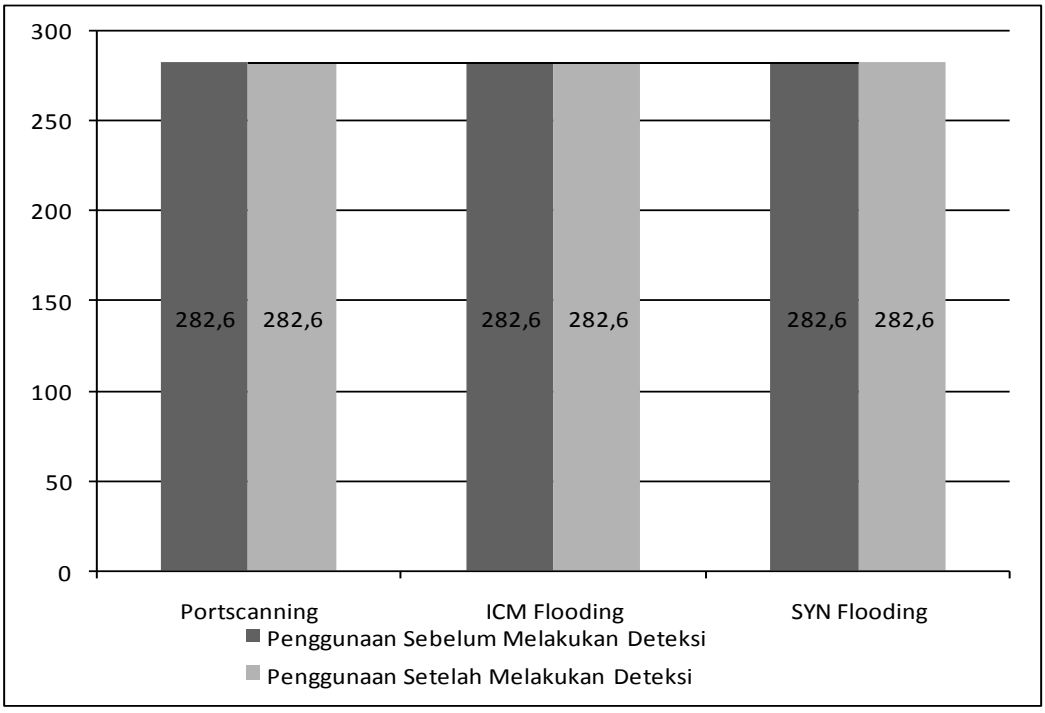
1. Pengujian dan analisis terhadap identifikasi terhadap deteksi *portscanning*, ICMP *flooding* dan SYN *flooding*. Pengujian terhadap fungsi indentifikasi intrusi bertujuan untuk mengetahui kemampuan deteksi yang dilakukan oleh sistem pendeteksi intrusi dalam mengidentifikasikan intrusi yang ditujukan kepada sistem keamanan jaringan internal. Berikut hasil dari pengujian yang dilakukan:
    a. Hasil identifikasi terhadap deteksi *portscanning* dapat disimpulkan sebagai berikut; sebelum dan sesudah deteksi yang dilakukan oleh sistem, dilakukan pengukuran terhadap intensitas penggunaan jaringan. Dimana hasil yang diperoleh yaitu terjadinya peningkatan yang signifikan terhadap lalu lintas jaringan yang terjadi pada sensor deteksi. Selain itu hasil deteksi menunjukkan bahwa paket jaringan yang tertangkap oleh sistem dan terjadi peringatan sebanyak 18 paket data.
    b. Hasil identifikasi terhadap deteksi ICMP *flooding* dapat disimpulkan sebagai berikut; sebelum dan sesudah deteksi yang dilakukan oleh sistem, dilakukan pengukuran terhadap intensitas yang terjadi pada lalu lintas jaringan. Dimana hasil yang diperoleh yaitu terjadinya peningkatan yang signifikan terhadap lalu lintas jaringan yang terjadi pada sensor

deteksi. Selain itu hasil deteksi menunjukkan bahwa paket jaringan yang tertangkap oleh sistem dan terjadi peringatan sebanyak 1251 paket data.

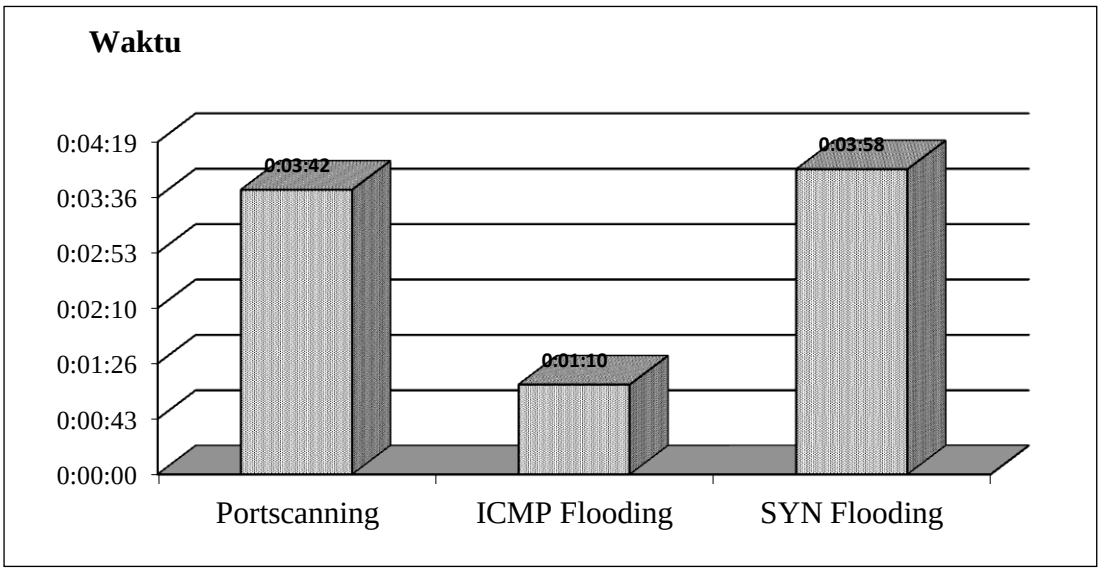
c. Hasil identifikasi terhadap deteksi SYN *flooding* dapat disimpulkan sebagai berikut; sebelum dan sesudah deteksi yang dilakukan oleh sistem, dilakukan pengukuran terhadap intensitas yang terjadi pada lalu lintas jaringan. Dimana hasil yang diperoleh yaitu terjadinya peningkatan yang signifikan terhadap lalu lintas jaringan yang terjadi pada sensor deteksi. Selain itu hasil deteksi menunjukkan bahwa paket jaringan yang tertangkap oleh sistem dan terjadi peringatan sebanyak 18 paket data

2. Pengujian dan analisis terhadap penggunaan sumber daya *(disk space,* memori dan *cpu*). Pengujian dan analisis ini bertujuan untuk melihat seberapa besar penggunaan *disk space,* memori dan *cpu* dari setiap deteksi yang dilakukan oleh sistem deteksi intrusi.

Grafik yang diperlihatkan pada Gambar 5 sampai dengan Gambar 7 memperlihatkan hasil pengujian yang dilakukan

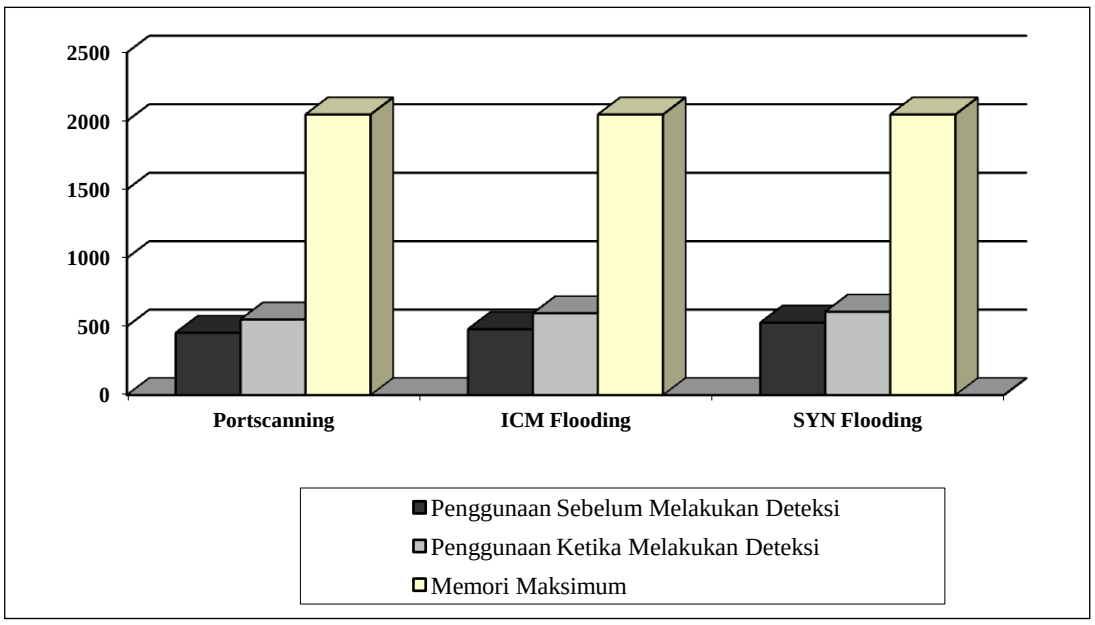


Gambar 5. Pengujian dan analisis terhadap perbandingan penggunaan disk space pada setiap deteksi.



Gambar 6. Grafik perbandingan lama waktu dalam melakukan deteksi.

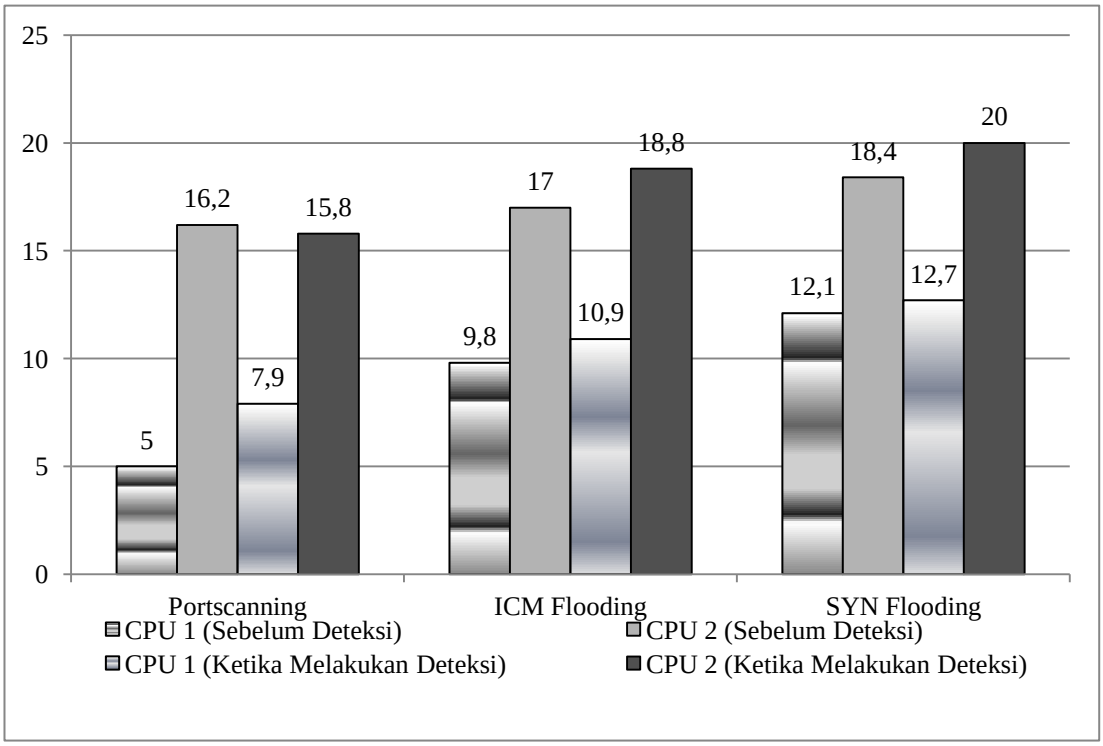
Berdasarkan Gambar 6 perbedaan deteksi pada setiap jenis deteksi yang berbeda, belum menunjukkan penggunaan signifikan *disk space* pada sistem deteksi intrusi. Hal ini kemungkinan disebabkan oleh faktor waktu yang hanya memakan waktu tidak terlampau lama dan hanya menggunakan sensor deteksi tunggal.



Gambar 7. Pengujian dan analisis terhadap penggunaan memori terhadap setiap deteksi yang berbeda.

Berdasarkan Gambar 7 setiap deteksi yang dilakukan memerlukan penggunaan memori yang berbeda-beda. Hal ini berpengaruh terhadap proses deteksi dan tempat penyimpanan sementara yang dibutuhkan oleh sistem dalam melakukan proses deteksi.

Gambar 8 merupakan perbandingan penggunaan CPU sebelum dan ketika melakukakan deteksi..



Gambar 8. Perbandingan penggunaan cpu sebelum dan ketika melakukan deteksi.

5. KESIMPULAN

Dari hasil pengukuran yang diperoleh, dapat disimpulkan sebagai berikut:

1. Sistem deteksi intrusi yang dibuat mampu untuk melakukan proses identifikasi terhadap berbagai bentuk intrusi berdasarkan *rule* yang ditetapkan.
2. Sistem pendeteksi intrusi memerlukan sumber daya (memori dan *CPU*) yang berbeda ketika mendeteksi objek intrusi yang berbeda.
3. Pada pengujian deteksi *portscanning,* sebelum deteksi dilakukan penggunaan memori oleh sistem sebesar 22,6% dan ketika deteksi dilakukan sebesar 27,5%.

Persentasi penggunaan memori pada pengujian deteksi ICMP *Flooding*, sebelum deteksi dilakukan penggunaannya sebesar 24% dan ketika deteksi dilakukan penggunaannya sebesar 29,7%. Persentasi dari penggunaan memori pada pengujian deteksi SYN *Flooding*, sebelum deteksi dilakukan penggunaannya sebesar 26,4% dan ketika deteksi dilakukan penggunaannya sebesar 30,3%.

4. Penggunaan *CPU* 1 dan 2 pada pengujian deteksi *portscanning*, sebelum deteksi dilakukan penggunaannya sebesar 5% dan 16,2%. Dan ketika deteksi dilakukan pada *CPU* 1 dan 2 persentasi penggunaannya sebesar 7,9% dan 15,8% . Ketika melakukan pengujian deteksi ICMP *flooding* dilakukan, persentasi penggunaan *CPU* 1 dan 2 sebelum deteksi penggunaannya sebesar 9,8% dan 17%. Dan ketika deteksi dilakukan penggunaannya sebesar 10,9% dan 18,8%. Pada pengujian deteksi SYN *flooding* penggunaan *CPU* 1 dan 2 sebelum deteksi dilakukan, adalah sebesar 12,1% dan 18,4% dan ketika deteksi dilakukan penggunaan *CPU* 1 dan 2 adalah sebesar 12,7% dan 20%.
5. Penggunaan *disk space* pada sistem untuk dapat menyimpan hasil dari deteksi *(log*) belum terlihat perbedaan pada penggunaan *disk space* ketika mendeteksi objek intrusi (penyusupan) yg berbeda.

DAFTAR PUSTAKA

[1]. E. Proctor, Paul, *The Practical Intrusion Detection Handbook*, Prentice Hall PTR, Prentice Hall, Inc, New Jersey, 2001.

[2]. Stallings, William, *Data and Computer Communication*, Eight Edition, Pearson Prentice Hall, New Jersey, 2007.

[3]. Ur Rehman, Rafeeq, *Intrusion Detection with Snort,* Bruce Peren's Open Source Series, 2003.

[4]. www.snort.org, Gullet, David, *Snort 2.9.3.0 On Ubuntu 10.04 LTS*, Symetric Tecnologies, 2012.

[5]. www.snort.org, Snort Team, *Snort Manual*, 2012.