Coding : Jurnal Komputer dan Aplikasi
Volume 07, No. 1 (2019), hal 97-107 ISSN: 2338-493X

IMPLEMENTASI INTRUSION PREVENTION SYSTEM (IPS) MENGGUNAKAN *SNORT* DAN *IPTABLE* PADA MONITORING JARINGAN LOKAL BERBASIS *WEBSITE*

[1]Rudy Suwanto, [2]Ikhwan Ruslianto, [3]Muhammad Diponegoro
[1][2] [3]Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963 e-mail:
[1]rudysuwanto@student.untan.ac.id, [2] ikhwanruslianto@siskom.untan.ac.id,
[3]muhammad.diponegoro@siskom.untan.ac.id

Abstrak

Server adalah pusat penyedia layanan dan pengolahan data dalam suatu jaringan, permintaan yang dikirim oleh *client* akan diolah *server*. Kinerja *server* bergantung terhadap paket yang dikirim oleh *client* pada jaringan. Untuk menyelesaikan permasalahan tersebut, dilakukan penelitian dengan mengimplementasikan *intrusion prevention system* (IPS) menggunakan *snort* dan *iptables* pada jaringan lokal. Antarmuka *website* sebagai media pemantauan kinerja *server* dan penanganan serangan yang terjadi. Deteksi serangan yang dilakukan oleh *snort* terfokus pada serangan pada *port icmp*, serangan pada *port tcp* dan serangan pada *port udp*. Sistem penanganan *iptables* menggunakan alamat *ip* penyerang untuk diberikan aksi, aksi yang dilakukan terbagi menjadi aksi *accept*, aksi *reject* dan aksi *drop*. Pengujian dilakukan dengan kondisi serangan *Ping of Death* dan *Port Scanning*. Dari pengujian yang telah dilakukan didapat persentase keberhasilan sistem mendeteksi sebesar 90% untuk serangan *ping of death* dan 85% pada serangan *port scanning*. Waktu aksi *iptables* rata-rata pada 2,27/detik untuk aksi *accept*, 1,42/detik untuk aksi *reject* dan 5,0/detik untuk aksi *drop*. Kinerja cpu *server* awal adalah 4% naik menjadi 15,3% akibat serangan *ping of death* dan mendapat kenaikan sebesar 58,1% untuk serangan *port scanning* (*intense scan*). Dari pengujian yang telah dilakukan kenaikan berlebihan terdapat pada *port udp* dapat mengakibatkan kenaikan kinerja *server* hingga 95%.

Kata Kunci: *Intrusion Prevention Sistem (IPS)*, Kinerja *Server*, Snort, *IPTables*, *Ping of Death, Port Scanning*

1. PENDAHULUAN

Server merupakan pusat penyedia layanan dan pengolahan data dalam suatu jaringan, permintaan yang dikirim oleh *client* akan diolah *server*. Kinerja *server* bergantung terhadap paket pertukaran data yang dikirim oleh *client* pada jaringan. Dalam suatu jaringan administrator bertindak untuk memberikan kenyamanan sehingga dibuat monitoring *server* agar dapat melihat tingkat kinerja *server*. Dari monitoring *server* yang telah dilakukan oleh administrator pada terdapat sebuah kekurangan yaitu keamanan yang terbagi menjadi, cara mendeteksi serangan dan melakukan tindakan apabila terjadi serangan pada jaringan. Keamanan sangat dibutuhkan agar *server* dalam kondisi yang baik. Jika dalam kondisi yang kurang baik serangan dapat dengan gampang masuk dan menyerang jaringan komputer, jika pada jaringan komputer tersebut tidak terdapat sistem keamanan untuk mengatasi serangan yang masuk. Ada beberapa serangan yang sering terjadi pada jaringan yang dilakukan *client* diantaranya adalah DoS (*Denial of Service*) dan *Port Scanning* merupakan

serangan yang dilakukan oleh pelaku melalui paket-paket jaringan dalam jumlah tertentu untuk mengacaukan jaringan dan mendapatkan informasi dari jaringan. Serangan tersebut dilakukan pada *port* yang terbuka di *server*, umumnya serangan mengarah pada *port icmp, tcp* dan *udp*. Serangan pada *port* tersebut dapat berdampak pada baik dan buruknya kinerja dari suatu *server* untuk memberikan pelayanan pada *client* lain.

Teknologi yang dapat digunakan dalam menyelesaikan masalah ini salah satunya adalah dengan membuat sistem *Intrusion Prevention System* (IPS) untuk mendeteksi dan menangani serangan yang menuju ke *server*. IPS merupakan suatu sistem kembangan yang menyatukan *Intrusion Detection System* (IDS) dan IPTables. IDS mampu mendeteksi serangan sesuai dengan rule yang dibuat dan IPTables digunakan untuk memberikan penanganan terhadap serangan sesuai dengan aksi yang dilakukan oleh administrator. Dwi Kuswanto tahun 2014 melakukan penelitian yang berjudul "Unjuk Kerja *Intrusion Prevention Sistem* (IPS) Berbasis Suricata Pada Jaringan Lokal Area *Network*"[1]. Penelitian menggunakan Suricata dan tidak menggunakan Aksi dari IPTable sebagai pemilihan penanganan serangan. Salah satu penelitian yang menggunakan IPTable sebagai aksi untuk serangan adalah Tamsir Ariyadi tahun 2012 yang berjudul "Implementasi *Intrusion Prevention System* (IPS) Pada Jaringan Komputer Kampus B Universitas Bina Darma[2].

Namun pada penelitian yang telah dilakukan tersebut masih belum terdapat kontrol yang dapat dilakukan di *website* untuk memberikan pilihan aksi yang dapat dilakukan terhadap serangan.

Berdasarkan permasalahan dari latar belakang maka dibuatlah penelitian tentang Implementasi *Intrusion Prevention System* (IPS) menggunakan Snort dan IPTables pada Jaringan Lokal Berbasis *Website*. Sebagai solusi dalam mendeteksi serangan dan memberikan penanganan terhadap serangan. Penanganan serangan yang dapat dikontrol melalui *interface website*. Sehingga jika ada serangan terdeteksi oleh sistem administrator dapat memberikan penangan ke serangan untuk menjaga kinerja *server* agar tetap baik.

2. LANDASAN TEORI

2.1 *Intrusion Prevention System* (IPS)

Secara konsep, IPS adalah sistem yang mampu atau memiliki fungsi mendeteksi dan memberikan penanganan serangan. Dengan kata lain, IPS merupakan pengembangan dari IDS dengan menambahkan beberapa komponen seperti *firewall* dan beberapa komponen lain untuk bekerja sama dalam mencegah dan menghentikan terjadinya penyusupan dari *client*[3].

2.2 SNORT

Snort merupakan sebuah perangkat lunak yang bersifat sumber sistem bebas (*open source*) yang dikembangkan oleh www.sourcefire.com dan menjadi salah satu *open source file* terbaik untuk mendeteksi serangan pada jaringan[4]. Dijelaskan bahwa ada beberapa keuntungan dari penggunaan snort antara lain kemudahannya dalam melakukan konfigurasi dan penggunaan *rules* yang fleksibel.

Dimana apabila terdapat sebuah serangan yang baru dapat dengan mudah menambahkannya dalam rules *database*, serta snort juga memiliki kemampuan dalam menganalisis paket data mentah yang membuatnya menjadi salah satu IDS terbaik.

Gambar 2. Snort
Sumber:www.snort.org

Pada pengoperasiannya SNORT dapat bekerja ke dalam tiga *Mode*, yang dapat disesuaikan dengan kebutuhan pengguna.

1. *Sniffer Mode*

Pada *Mode* operasi ini snort bertindak sebagai sniffer, yaitu snort dapat menangkap atau melihat semua paket yang lewat dalam jaringan dimana Snort diletakkan. Snort menampilkan hasil dari sniffing ini secara real time. Tampilan dari data real time dalam bentuk console.

2. *Packet* Logger *Mode*

Pada *Mode* ini selain dapat melihat semua paket yang lewat, Snort juga dapat mencatat atau logging menyimpannya pada *storage*. Snort menyimpan paket tersebut dengan keluaran *file* u2.

3. *Network Intrusion Detection Mode*

Mode IDS secara luas dikenal dengan *mode* deteksi terhadap serangan. Ciri khas dari *mode* ini adalah dengan menjalankan snort beserta *file* konfigurasi yang pengguna telah ditentukan. *File* (snort.conf) ini sudah banyak mencakup konfigurasi-konfigurasi yang dibutuhkan dalam *mode* ini.

pada penelitian ini *mode* snort yang dipakai adalah IDS atau deteksi serangan. Tujuannya untuk mendeteksi semua paket acak yang masuk ke jaringan dan paket serangan ditentukan berdasarkan *rule* yang dibuat.

2.3 IPTables

Adalah suatu kelompok arsitektur pemrosesan paket jaringan aturan ke dalam tabel berdasarkan fungsi (filter paket, jaringan terjemahan alamat, dan paket lainnya) yang masing-masing memiliki rantai (urutan) aturan pemrosesan. Aturan terdiri dari kecocokan (digunakan untuk menentukan apa yang akan dilakukan dengan pencocokan paket).

Dapat disimpulkan bahwa *iptables* merupakan suatu *firewall* yang ada pada linux yang berfungsi untuk menganalisis dan menyaring paket data yang masuk kedalam *firewall*, dan dibagikan ke 3 kategori aksi yaitu:

1. *Drop*
   (membiarkan paket tersebut seolah-olah tidak pernah diterima).
2. *Accept*
   (menerima paket tersebut untuk diproses lebih lanjut).
3. *Reject*
   (menolak dan memberitahukan pengirim bahwa paket data tidak bisa diterima).

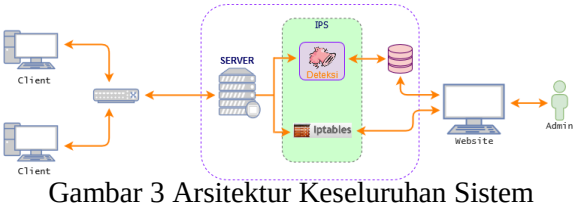
Purdy menjelaskan bahwa IPTabless juga sebagai alat untuk menyaring paket-paket yang masuk, keluar dan sedang berlalu lintas di dalam *Firewall* melalui *server*. IPTabless mendedikasikan lima "hook point" di dalam jalur pemrosesan paket kernel: *Prerouting*, *Input*, *Forward*, *Postrouting* dan *Output*. Setiap aturan merupakan peluang untuk mempengaruhi atau memantau aliran paket[5].

3. METODOLOGI PENELITIAN

Proses penelitian dimulai dengan melakukan studi pustaka terkait *server*, system kerja IPS, konfigurasi Snort, dan Konsep Aksi IPTables yang dilakukan memalui *website*. Selanjutnya dilakukan perancangan yang terdiri dari perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras meliputi keseluruhan topologi untuk membentuk jaringan lokal. Perancangan perangkat lunak adalah dari keseluruhan sistem IPS yang akan diimplementasikan menggunakan snort sebagai sistem deteksi dan IPTables sebagai aksi serangan pada *server* dengan antar muka *website*. Selanjutnya melakukan pengujian untuk mengetahui kinerja *server* dan pengaruh sistem IPS. Pengujian dilakukan dengan beberapa skenario uji berupa serangan dengan cara *ping of death* dan *port* scanning. Setelah dilakukan pengujian, maka dilakukan analisa dari pengujian untuk mendapatkan kesimpulan akhir dari proses penelitian.

4. PERANCANGAN SISTEM

Rancangan sistem secara keseluruhan dapat dilihat pada Gambar 3.



Gambar 3 Arsitektur Keseluruhan Sistem

Sistem IPS melakukan proses monitoring dan tindakan berdasarkan aturan yang dibuat penggunanya. Sistem dapat melihat kegiatan yang ada di jaringan dengan mendeteksi aktivitas yang dilakukan pada pengguna jaringan. *Client* menyerang dengan memberikan sebuah permintaan yang besar ke *server*, paket akan melewati *firewall* bawaan *server* sebelum sistem mendeteksi paket. Pada penelitian ini paket yang menuju *port icmp*, *tcp* dan *udp*, menjadi fokus utama pengamanan serangan pada *server*. Paket yang telah dideteksi melewati jaringan akan disaring (*filter*) oleh aturan (*rule*)

yang telah dibuat pengguna, data aktivitas jaringan akan di simpan dalam bentuk *log file* dan akan di kirim ke *database* untuk di tampilkan pada *website*.

Dengan memonitoring data serangan tersebut admin dapat memberi tindakan serangan berdasarkan IP Address penyerang, tindakan tersebut menggunakan fitur *iptables* melalui *website*. Pada penelitian ini *Intrusion Prevention System* (IPS) menggunakan snort sebagai sistem untuk memonitoring dan penyaringan data dan IPTables untuk melakukan tindakan dari serangan.

Pada perancangan sistem penelitian terbagi menjadi perancangan perangkat keras dan perancangan perangkat lunak.

4.1. Perancangan Arsitektur Perangkat Keras

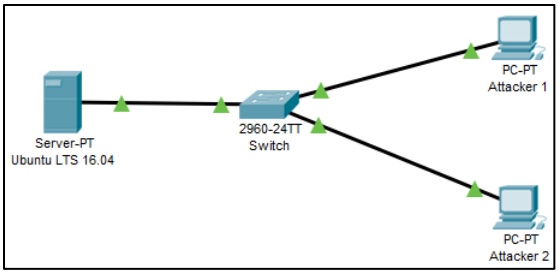
Perancangan arsitektur perangkat keras merupakan perancangan untuk melakukan pengujian pada penelitian pada proses perancangan perangkat keras terdiri dari *server*, *client* yang disatukan dalam satu jaringan yang sama (lokal) atau biasa disebut topologi. Berikut merupakan kebutuhan perangkat keras yang digunakan topologi. Berikut merupakan kebutuhan perangkat keras yang digunakan dalam penelitian yang terdiri dari:

1. *Server* Ubuntu

   Server digunakan sebagai tempat penampung data yang terjadi selama penelitian berlangsung. Komputer yang digunakan sebagai *server* dengan spesifikasi: Intel® Core i3, Ram 8Gb DDR3, Hard Drive 320GB, Interface Realtek 8111E PCI-E Gigabit, LAN Card D-Link DGE-560T dan sistem operasi yang digunakan adalah Ubuntu 16.04 LTS.
2. Topologi jaringan

   Pada perancangan topologi yang digunakan adalah local *client server* sehingga perangkat keras yang dibutuhkan adalah sebagai berikut Switch / Hub, Kabel UTP Standar, Konektor RJ-45 CAT5 dan Laptop *client* sebagai attacker serangan.

Topologi yang digunakan dalam Perancangan penelitian secara garis besar dapat dilihat pada gambar 4.

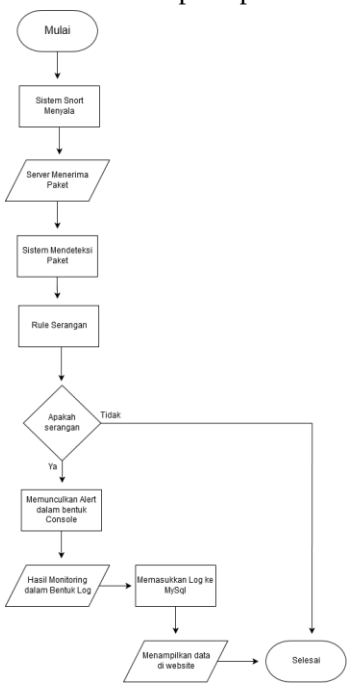


Gambar 4. Topologi Jaringan

Topologi jaringan merupakan bagian dari proses perancangan perangkat keras. Perancangan perangkat keras dilakukan untuk menghubungkan perangkat ke dalam satu jaringan. Proses perancangan perangkat keras terdiri dari *server*, *client* yang disatukan dalam satu jaringan yang sama (lokal) atau biasa disebut topologi. Cara kerja pada penelitian ini adalah *client* bisa menjadi attacker dan dibuat skenario *client* menyerang *server* dengan dua cara ping of death dan *port scanning*. Konfigurasi yang dilakukan adalah *client server* dimana perangkat jaringan berada pada satu jaringan yang sama.

4.2 Perancangan Sistem Snort

Perancangan sistem snort pada penelitian berguna sebagai sistem pengumpulan data dari setiap serangan. Agar sistem dapat berfungsi dan berjalan sesuai dengan yang dirancang, maka diperlukan alur sistem seperti pada Gambar 5.



Gambar 5. Diagram Alir sistem snort

Server yang sebelumnya dirancang berada pada satu topologi jaringan, akan di monitoring menggunakan sistem snort untuk melihat terjadinya serangan. Sistem snort akan berada pada *server* untuk memonitoring aktivitas *client* yang terhubung ke jaringan.

Aktivitas yang dilakukan *client* akan dicocokkan atau disamakan menggunakan *rule* untuk menentukan apakah data termasuk serangan. Data yang dianggap serangan akan tersimpan ke dalam *log* snort (u2), proses pengubahan data tersebut dilakukan oleh banyard2 agar dapat tersimpan di *database*s.

4.3 Perancangan Pola Serangan

Perancangan serangan bertujuan untuk menguji sistem saat menerima serangan. Pada penelitian ini serangan yang digunakan terbagi menjadi dua ping of death dan *port* scanning.

1. Serangan ping of death

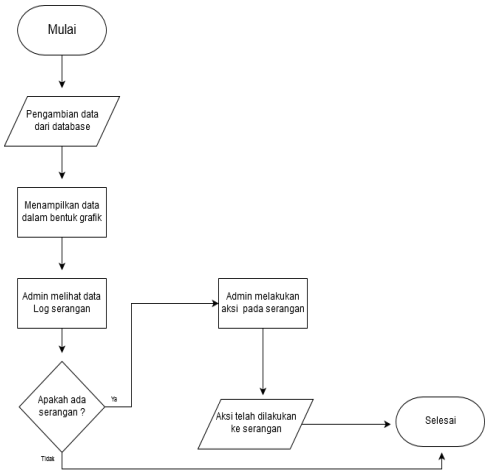
   Ping of death adalah suatu metode untuk meminta *request* yang berlebihan ke *server*. *Ping of death* menyerang *port icmp* dan *udp* pada *server*, *client* yang melakukan serangan tersebut berniat membuat *server* menjadi down.
2. Serangan *port* scanning

   port scanning adalah serangan untuk memperoleh informasi target. Tujuan serangan adalah membanjiri *request* ke *port icmp*, *tcp* dan *udp*, hingga mendapat informasi melalui *port* tersebut.

4.4 Perancangan *Website* IPS

Website pada penelitian ini berfungsi sebagai penyedia data hasil monitoring yang dilakukan sistem snort dan sebagai sistem prevention atau sebagai sistem pencegahan serangan. Antarmuka *website* menampilkan seluruh data *server*, status, jumlah *request* dan lain-lain. Data yang masuk diolah sesuai logika untuk menjalankan beberapa fitur yang disediakan oleh *website* IPS seperti sistem notifikasi, grafik real-time, status, sensor dan aksi (*iptabel*).

Melalui *website* IPS administrator dapat mengetahui kondisi terbaru dari *server* dan keseluruhan arsitektur tanpa harus masuk kedalam sistem *server* tersebut. Alur dari perancangan *website* IPS dapat dilihat pada Gambar 6.

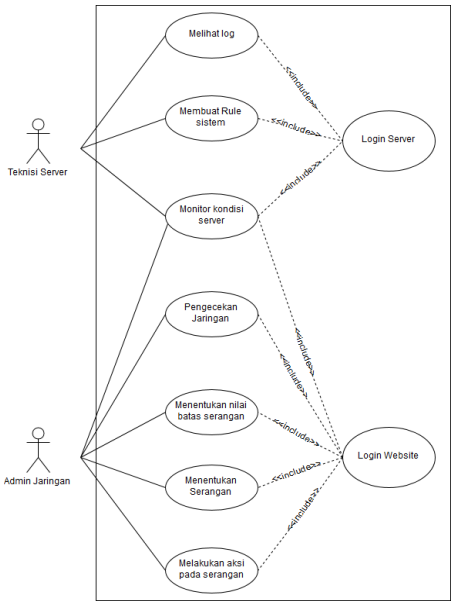


Gambar 6. Diagram Alir *Website* IPS

Pada gambar 6 menjelaskan sistem IPS yang berjalan pada *website*. Sistem dimulai ketika mendapat data serangan yang baru, *website* mengambil data serangan terbaru berdasarkan *timestamp* dari data. Data tersebut akan ditampilkan *website* dalam bentuk grafik agar admin jaringan dapat membaca data tersebut.

4.3.1 *Use case* Diagram Sistem

Use case diagram digunakan sebagai media untuk menjelaskan alur kerja sistem yang digunakan. Penjelasan terkait *use case* ditunjukkan pada Gambar 7.



Gambar 7. *Use case* diagram sistem

Dalam tahap perancangan penelitian sistem digunakan *use case* diagram untuk menggambarkan *fungsionalitas* sistem yang dilakukan aktor. Terdapat dua aktor pada sistem ini yaitu admin jaringan dan teknisi *server*. Sistem dapat digunakan oleh admin melalui antarmuka *website*. Admin dapat melakukan monitoring kondisi *server* melaui *website*, menentukan batas nilai serangan menentukan serangan yang terjadi dan memerikan aksi terhadap serangan yang dianggap dapat mengganggu kinerja *server*.

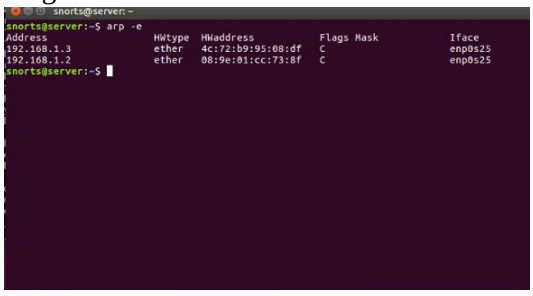
Penggunaan sistem oleh teknisi *server* mencakup teknisi dapat melihat *log* serangan yang terjadi pada jaringan, membuat rule serangan pada *port* yang ingin dilindungi di *server* dan monitoring kondisi *server* yang termasuk dalam keadaan *server* dan kinerja *server* saat terjadi serangan melalui diagnosa fisik *server*.

5. IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1 Implementasi Arsitektur perangkat keras

Implementasi perangkat keras meliputi proses perakitan komponen- komponen yang dibutuhkan yang meliputi topologi jaringan.

Hasil dari penerapan skema pada gambar 4 merupakan perancangan arsitektur perangkat keras pada jaringan. Implementasi perancangan dapat dilihat berdasarkan topologi *client server* pada gambar 8.

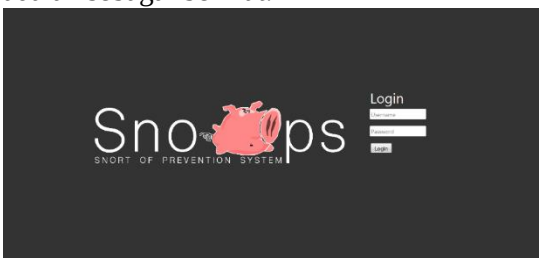


Gambar 8 Topologi jaringan

Pada gambar 8 diperlihatkan hubungan antara *client* dan *server*. *Server* memiliki dua perangkat yang terhubung ke interface enp0s25, alamat IP pada *server* adalah “192.168.1.1”. Perangkat yang terhubung memiliki alamat IP “192.168.1.3” dan “192.168.1.2”. Alamat IP perangkat *client* disesuaikan dengan *server* agar dapat terhubung dengan baik. Dengan demikian *server* telah terhubung dengan kedua *client* yang di rancang.

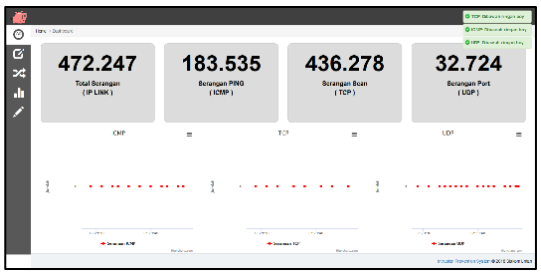
5.2 Implementasi *Website* IPS

Implementasi *website* dilakukan untuk merealisasikan hasil perancangan *website* pada penelitian. Hasil dari perancangan *website* IPS adalah sebagai berikut.



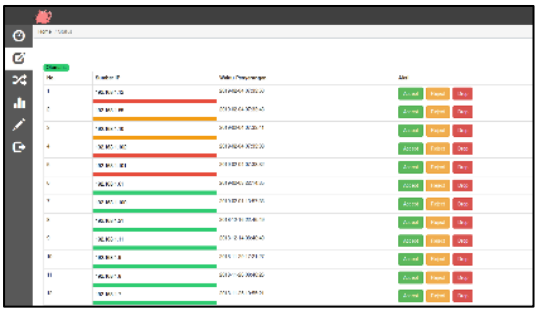
Gambar 9. Halaman login *website*

Halaman login dibuat untuk melalukan pengecekan data admin yang tersedia pada *database*.



Gambar 10. Tampilan deteksi

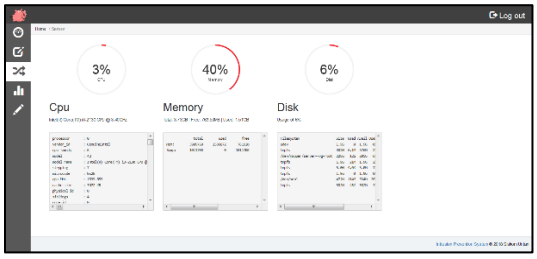
Pada gambar 10 Admin dapat melakukan kontrol penuh penggunaan fitur pada *website* seperti, monitoring serangan pada jaringan, melihat data yang didapat sistem.



Gambar 11. Tampilan Menu Aksi

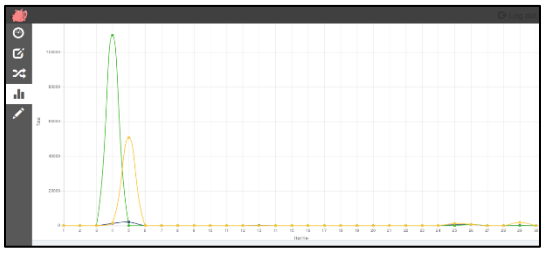
Menu aksi digunakan untuk memberi aksi terhadap alamat *ip* yang dianggap sebagai

serangan. Aksi yang dilakukan menggunakan fitur *iptables* yang terhubung dengan *website.*



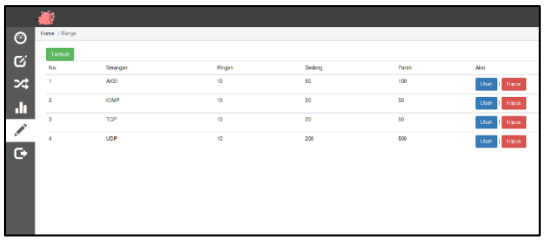
Gambar 12. Tampilan Menu Sensor

Pada menu ini berguna untuk mengetahui kinerja server. Administrator dapat melihat kinerja terbaru server, yakni dengan melihat grafik *cpu, memory* dan *disk*.



Gambar 13. Tampilan Menu Grafik

Data serangan yang telah tertangkap dapat di lihat menggunakan waktu serangan dan di tampilkan secara grafik dengan indikator warna sesuai tingkatan serangan. Grafik terbagi menjadi 3 yakni *port icmp, tcp* dan *udp* dengan nilai yang tertera pada saat grafik dipilih.



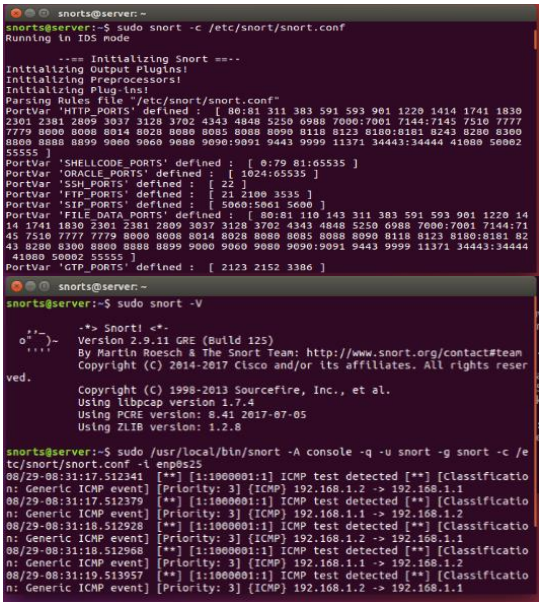
Gambar 14. Tampilan Menu Range

Pada gambar diatas menunjukkan administrator dapat memberi nilai pada tipe serangan yang meliputi level ringan, sedang dan parah. Nilai dapat diubah administrator sesuai dengan kebutuhan. Pada menu ini terdapat fitur tambah, ubah dan hapus.

5.3 Pengujian Snort

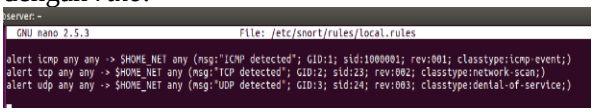
Pada tahap pengujian sistem snort adalah tahapan untuk melakukan pengujian mulai dari rule snort, snort IDS dan barnyard2. Dimana dari hasil pengujian untuk melihat kinerja sistem snort yang dibuat sehingga peneliti mendapatkan hasil dari penelitian yang telah dilakukan.

Pengujian snort dilakukan untuk melihat apakah snort dapat bekerja dengan baik saat terjadi serangan pada jaringan. Sehingga dengan dilakukannya pengujian sistem snort diperoleh hasil atau keluaran data yang baik.



Gambar 15. Snort IDS *Mode*

Dari gambar diatas snort telah dapat berjalan sesuai untuk mendeteksi adanya serangan. Snort *Mode logging packet* IDS berjalan dengan baik untuk mendeteksi serangan dari *client*. *Alert* yang dikeluarkan dalam bentuk *console* menyesuaikan dengan *rule*.



Gambar 16. Rule

Rule yang telah diimplementasikan berfungsi untuk mendeteksi tipe *port* yang menjadi tujuan dari serangan. *Rule* menjadi aturan yang berada pada snort, sehingga saat snort berjalan *rule* akan selalu dipanggil.

5.4 Pengujian Serangan

Pada tahap ini dilakukan pengujian serangan untuk melihat apakah sistem dapat mendeteksi adanya serangan dan dapat mengklasifikasikannya ke *port icmp*, *tcp* dan *udp*. Pengujian serangan terbagi menjadi dua

pengujian yaitu serangan *ping of death* dan *port scanning*. Serangan tersebut dilakukan dengan melakukan pengujian secara berulang selama sepuluh kali pengujian, untuk mendapat perbandingan data serangan yang baik.

5.4.1 *Ping of death*

Pengujian pada *port icmp* yang terakhir adalah *Ping of Death*, pengujian ini *client* melakukan permintaan berkala dan beban paket secara besar ke *server*.

```
192.168.1.1 -l 65500 -n 1000
```

Serangan *icmp* yang dilakukan dengan jumlah beban (-l 65500) dan jumlah paket yang dikirim oleh penyerang sebanyak 1000 kali permintaan ke *server*. Dari perintah diatas hasil dari pengujian dapat dilihat dibawah.

Tabel 1. Pengujian *Ping of Death*

No.	Sumber	Tujuan	Kondisi	Paket yang diterima *server*		Kinerja Cpu *server* naik hingga (%)
				ICMP	UDP	
1	192.168.1.2	192.168.1.1	IDS	2000	60	11
2	192.168.1.2	192.168.1.1	IDS	2000	60	10
3	192.168.1.2	192.168.1.1	IDS	2000	60	15
4	192.168.1.2	192.168.1.1	IDS	2000	60	13
5	192.168.1.2	192.168.1.1	IDS	2000	60	12
6	192.168.1.2	192.168.1.1	IDS	2000	60	14
7	192.168.1.2	192.168.1.1	IDS	2000	60	22
8	192.168.1.2	192.168.1.1	IDS	2000	60	18
9	192.168.1.2	192.168.1.1	IDS	2000	60	13
10	192.168.1.2	192.168.1.1	IDS	2000	60	17
11	192.168.1.2	192.168.1.1	IDS	2000	60	17
12	192.168.1.2	192.168.1.1	IDS	2000	60	13
13	192.168.1.2	192.168.1.1	IDS	2000	60	10
14	192.168.1.2	192.168.1.1	IDS	2000	60	11
15	192.168.1.2	192.168.1.1	IDS	2000	60	12
16	192.168.1.2	192.168.1.1	IDS	2000	60	15
17	192.168.1.2	192.168.1.1	IDS	2000	60	20
18	192.168.1.2	192.168.1.1	IDS	2000	60	17
19	192.168.1.2	192.168.1.1	IDS	2000	60	18
20	192.168.1.2	192.168.1.1	IDS	2000	60	22

Setelah melakukan pengujian di dapatlah angka kenaikan tertinggi dari *server* adalah 22 % dengan nilai rata-rata sebesar 15 %. Dari pengujian selama 20 kali yang dilakukan tingkat keberhasilan sistem dalam mendeteksi serangan sebesar 90% untuk serangan *ping of death.*

5.4.2 *Port Scanning*

Port Scanning merupakan serangan yang dilakukan untuk mendapatkan informasi *port* mana yang terbuka. Serangan ini dapat memungkinkan tindakkan penyerang untuk memonitoring target tujuan dan penyerang dapat menganalisa setiap data yang terhubung dengan penyerang. Secara umum serangan *port scanning* biasa dilakukan menggunakan aplikasi zenmap untuk mendapatkan hasil yang maksimal.

Tabel 2. *port scanning intense scan*

No	Sumber	Tujuan	Kondisi	Paket yang diterima *server*			Kinerja Cpu *server* naik hingga (%)
				ICMP	UDP	TCP	
1	192.168.1.3	192.168.1.1	IDS	20	4	2461	62
2	192.168.1.3	192.168.1.1	IDS	20	4	2460	61
3	192.168.1.3	192.168.1.1	IDS	20	4	2459	46
4	192.168.1.3	192.168.1.1	IDS	20	4	2463	61
5	192.168.1.3	192.168.1.1	IDS	20	4	2455	65
6	192.168.1.3	192.168.1.1	IDS	20	4	2460	61
7	192.168.1.3	192.168.1.1	IDS	20	4	2464	68
8	192.168.1.3	192.168.1.1	IDS	20	4	2459	64
9	192.168.1.3	192.168.1.1	IDS	20	4	2462	66
10	192.168.1.3	192.168.1.1	IDS	20	4	2458	73
11	192.168.1.3	192.168.1.1	IDS	20	4	2461	68
12	192.168.1.3	192.168.1.1	IDS	20	4	2460	66
13	192.168.1.3	192.168.1.1	IDS	20	4	2459	73
14	192.168.1.3	192.168.1.1	IDS	20	4	2463	61
15	192.168.1.3	192.168.1.1	IDS	20	4	2455	61
16	192.168.1.3	192.168.1.1	IDS	20	4	2460	61
17	192.168.1.3	192.168.1.1	IDS	20	4	2464	73
18	192.168.1.3	192.168.1.1	IDS	20	4	2459	64
19	192.168.1.3	192.168.1.1	IDS	20	4	2462	66
20	192.168.1.3	192.168.1.1	IDS	20	4	2458	68

Dari tabel diatas dapat dilihat penyerang melakukan *port*scan dengan fitur “intense scan” yang terdapat pada zenmap untuk mengetahui celah dari *server*. kenaikan kinerja cpu dalam tingkat tertinggi berapa pada 73% dengan nilai rata-rata 58,1%. Nilai kenaikan kinerja tersebut dengan keadaan *server* menyediakan layanan service pada *client* dan mendapat serangan.

Tabel 4. Pengujian *port scanning all port tcp*

No	Sumber	Tujuan	Kondisi	Paket yang diterima *server*			Kinerja Cpu *server* naik hingga (%)
				ICMP	UDP	TCP	
1	192.168.1.3	192.168.1.1	IDS	20	5	259519	42
2	192.168.1.3	192.168.1.1	IDS	20	5	248746	55
3	192.168.1.3	192.168.1.1	IDS	20	5	258544	48
4	192.168.1.3	192.168.1.1	IDS	20	4	257009	51
5	192.168.1.3	192.168.1.1	IDS	20	11	250825	51
6	192.168.1.3	192.168.1.1	IDS	20	4	258500	48
7	192.168.1.3	192.168.1.1	IDS	20	5	254768	54
8	192.168.1.3	192.168.1.1	IDS	20	4	256854	59
9	192.168.1.3	192.168.1.1	IDS	20	5	259612	55
10	192.168.1.3	192.168.1.1	IDS	20	4	256519	51
11	192.168.1.3	192.168.1.1	IDS	20	4	259519	48
12	192.168.1.3	192.168.1.1	IDS	20	5	257009	48
13	192.168.1.3	192.168.1.1	IDS	20	5	258544	53
14	192.168.1.3	192.168.1.1	IDS	20	4	257009	51
15	192.168.1.3	192.168.1.1	IDS	20	11	250825	54
16	192.168.1.3	192.168.1.1	IDS	20	11	250825	51
17	192.168.1.3	192.168.1.1	IDS	20	5	254768	53
18	192.168.1.3	192.168.1.1	IDS	20	4	256854	51
19	192.168.1.3	192.168.1.1	IDS	20	5	259612	53
20	192.168.1.3	192.168.1.1	IDS	20	11	256519	59

Dari tabel diatas dapat dilihat penyerang melakukan *port scanning* dengan fitur "*intense scan, all TCP*" yang terdapat pada zenmap, *scan* berfokus pada *port tcp* dari *server*. Dari pengujian yang telah dilakukan maka di dapatlah nilai kenaikan kinerja cpu pada *server* sebesar 59% dengan nilai rata-rata 51,75%. Pengujian yang telah dilakukan lebih memfokuskan serangan pada *port tcp*. Pada tabel diatas dapat dilihat bahwa nilai *tcp* lebih banyak dari pada nilai pada *port icmp* dan *udp*.

Tabel 5. *Port* scanning plus *udp*

No	Sumber	Tujuan	Kondisi	Paket yang diterima server			Kinerja Cpu server naik hingga (%)
				ICMP	UDP	TCP	
1	192.168.1.3	192.168.1.1	IDS	20	5	259519	42
2	192.168.1.3	192.168.1.1	IDS	20	5	248746	55
3	192.168.1.3	192.168.1.1	IDS	20	5	259544	48
4	192.168.1.3	192.168.1.1	IDS	20	4	257009	51
5	192.168.1.3	192.168.1.1	IDS	20	11	250825	51
6	192.168.1.3	192.168.1.1	IDS	20	4	258500	48
7	192.168.1.3	192.168.1.1	IDS	20	5	254768	54
8	192.168.1.3	192.168.1.1	IDS	20	4	256854	59
9	192.168.1.3	192.168.1.1	IDS	20	5	259612	55
10	192.168.1.3	192.168.1.1	IDS	20	4	256819	51
11	192.168.1.3	192.168.1.1	IDS	20	4	259519	48
12	192.168.1.3	192.168.1.1	IDS	20	5	257009	48
13	192.168.1.3	192.168.1.1	IDS	20	5	258544	53
14	192.168.1.3	192.168.1.1	IDS	20	4	257009	51
15	192.168.1.3	192.168.1.1	IDS	20	11	250825	54
16	192.168.1.3	192.168.1.1	IDS	20	11	250825	51
17	192.168.1.3	192.168.1.1	IDS	20	5	254768	53
18	192.168.1.3	192.168.1.1	IDS	20	4	256854	51
19	192.168.1.3	192.168.1.1	IDS	20	5	259612	53
20	192.168.1.3	192.168.1.1	IDS	20	11	256819	59

Dari tabel diatas dapat dilihat penyerang melakukan *port scanning* dengan fitur "intense scan plus *UDP*" yang terdapat pada zenmap, serangan ini dilakukan untuk meyakinkan seberapa *valid* data *port udp* yang telah dilakukan sebelumnya pada *server*. Pada pengujian ini didapat nilai kenaikan kinerja cpu sebesar 90% dengan nilai rata-rata 62,25%. Sehingga penulis harus menunggu sistem dapat kembali bekerja. Dari pengujian yang telah dilakukan menggunakan metode penyerangan *port scanning* nilai kenaikan kinerja yang paling tinggi sebesar 62,25%. Kenaikan tersebut berlaku jika *server* menerima beban yang lebih pada *port udp*, dikarenakan *port* tersebut memproses setiap unit data pada protokol.

Kemudian dari proses pengujian yang dilakukan bertujuan untuk mendapatkan nilai tingkat keberhasilan sistem dalam mendeteksi serangan. Tingkat keberhasilan sistem yang didapat sebesar 85%, dikarenakan terjadinya proses sistem tidak dapat mendeteksi paket yang lewat jika paket tersebut lebih dari 9000 paket untuk *port udp*.

5.5 Karakteristik Serangan

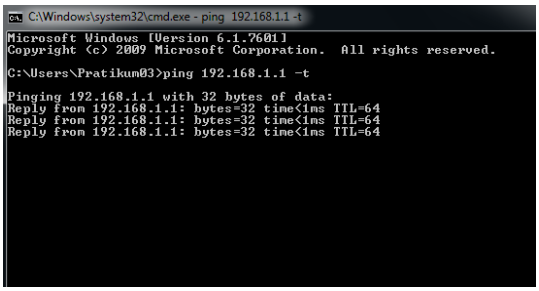
Dari pengujian yang telah dilakukan terdapat sebuah kebiasaan atau karakteristik serangan, dimana hal ini bisa menjadi penentu dari admin jaringan. Serangan ping of death dan port scanning mempunyai kebiasaan dalam sifat serangannya, dari pengujian yang telah dilakukan serangan tersebut selalu menuju port yang sama. Aktivitas yang dilakukan oleh client dapat dilihat sebagai serangan dengan melihat tabel berikut.

Table 5.19 Karakteristik Serangan

Tipe Serangan	Penyerangan Port		
	ICMP	TCP	UDP
Ping of Death	✓	-	✓
Port Scanning	✓	✓	✓

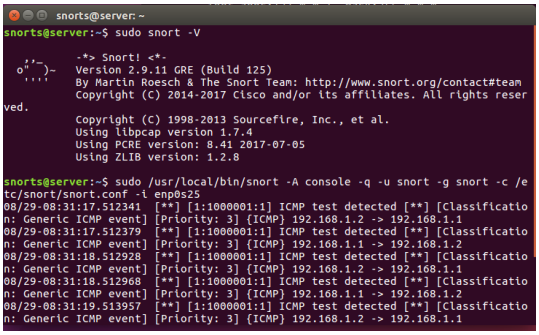
5.6 Pengujian Keseluruhan Sistem

Pada pengujian keseluruhan sistem di lakukan untuk melihat apakah sistem yang diimplementasikan dan *Website* monitoring yang dibuat dapat berjalan dengan baik. Saat adanya serangan dari client, snort menangkap paket yang dicurigai sebagai serangan dan menyimpan dalam bentuk log snort dengan format u2.



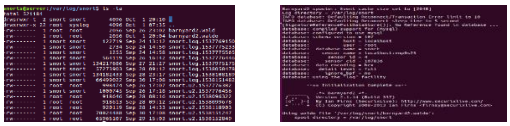
Gambar 5.64 Client melakukan ping

Penyerang melakukan request ke 192.168.1.1 secara terus. Permintaan ke server terus menerus dapat meningkatkan kinerja RAM Secara berlebihan, maka dari itu penulis membuat rule untuk mengklasifikasikan ping kedalam serangan pada port ICMP.



Gambar 5.65 Snort *Mode* IDS

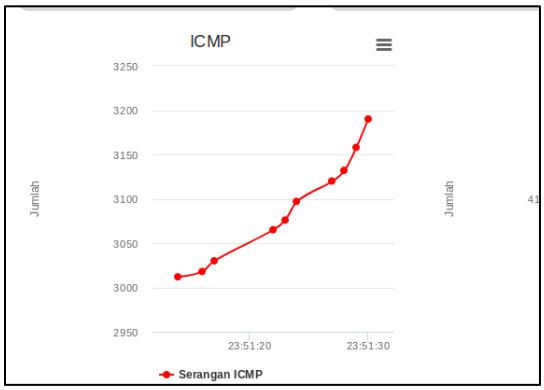
Snort menangkap adanya serangan pada protokol ICMP dan memberikan alert dalam bentuk console ke administrator. Kemudian hasil dari alert tersebut tersimpan dalam log snort dengan format u2.



(a) (b)

Gambar 5.66 (a) hasil output snort (b) proses banyard2

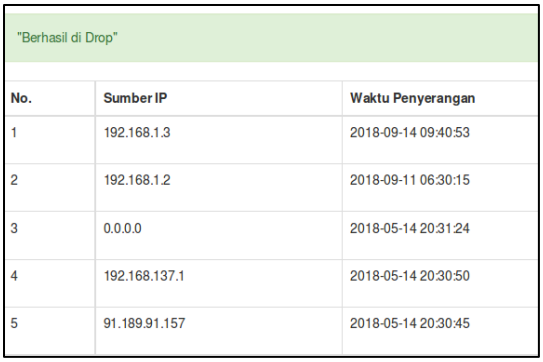
Barnyard2 mengubah (u2) snort menjadi log dan memasukkan ke database snort.



Gambar 5.67 *Website* monitoring

Data yang telah disimpan snort akan ditampilkan ke *website* dalam bentuk grafik. Dari gambar diatas dapat dilihat grafik *icmp* bergerak ke atas menandakan adanya data yang masuk *ke port icmp*.

Jumlah data yang dalam waktu penyerangan dapat dilihat pada menu grafik yang perlihatkan pada gambar 5.68.

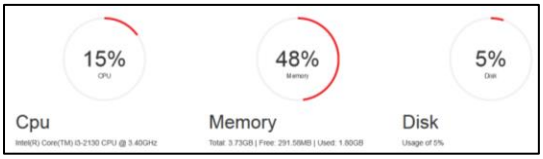


"Berhasil di Drop"

No.	Sumber IP	Waktu Penyerangan
1	192.168.1.3	2018-09-14 09:40:53
2	192.168.1.2	2018-09-11 06:30:15
3	0.0.0.0	2018-05-14 20:31:24
4	192.168.137.1	2018-05-14 20:30:50
5	91.189.91.157	2018-05-14 20:30:45

Gambar 5.68 Status

Untuk mengetahui alamat ip penyerang dapat melihat menu status, untuk membedakan data penyerang dapat dilihat pada waktu penyerangan dan admin dapat memberikan aksi berupa *accetp*, *reject* dan *drop*. Pada gambar diatas contoh aksi adalah *drop*.



Gambar 5.71 Kinerja server

Pengujian penelitian ini dilakukan dengan menggunakan perangkat yang ada pada lab komputasi, Prodi Rekayasa Sistem Komputer Fakultas MIPA Universitas Tanjungpura.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan dari "Implementasi Sistem IPS (*Intrusion Prevention System*) menggunakan Snort dan IPTables pada jaringan lokal berbasis *website*, didapat kesimpulan sebagai berikut:

1. Dalam pengujian sistem snort yang telah dilakukan selama 20 kali percobaan untuk mendeteksi paket serangan *ping of death* dan *port scanning*. Keberhasilan sistem dalam mendeteksi serangan sebesar 90% untuk serangan *ping of death* dan 85% pada serangan *port scanning*. Sistem lebih baik saat mendeteksi pada serangan *ping of death* dikarenakan serangan tersebut hanya memfokuskan diri pada satu titik *port* yaitu *icmp*.

2. Pencegahan serangan yang menuju *port* (*icmp, tcp* dan *udp*) pada penelitian dapat diatasi menggunakan iptables dengan aksi (*accept, reject* dan *drop*). Aksi *iptables* dapat berjalan dengan kecepatan nilai rata-rata (*accept*= 2,27 detik, *reject*= 1,42 detik, *drop*= 5,0 detik) untuk mencegah aktivitas yang dianggap serangan oleh administrator.
3. Kinerja cpu *server* awal adalah 4% pada saat *server* menerima serangan *ping of death* terjadi kenaikan kinerja cpu *server* sebesar 15,3% dan mendapat kenaikan sebesar 58,1% untuk serangan *port* scanning (intense scan). Dari pengujian *port scanning* "*intense scan plus udp*" dapat dilihat pada table 5.9 pengujian no.5 yang telah dilakukan sebanyak 20 kali terjadi kenaikan aktivitas atau serangan di *port udp* pada pengujian ke 5 (dari 58% hingga 95%) sehingga membuat *server* tidak mampu memproses dan berhenti sementara melakukan aktivitas atau *down*.

6.2 Saran

Berdasarkan penelitian yang telah dilakukan pada "Implementasi Sistem IPS (*Intrusion Prevention System*) menggunakan Snort dan IPTables pada jaringan lokal berbasis *Website*", maka diperoleh saran untuk penelitian lebih lanjut yaitu:

1. Pada penelitian selanjutnya diharapkan adanya penambahan fitur yang dapat membaca atau management tipe serangan menggunakan riwayat aktivitas serangan.
2. Penelitian selanjutnya disarankan untuk menggunakan jaringan *wireless* yang terhubung dengan penyerang melalui *wireless*.
3. Membuat sebuah kontrol monitoring yang dapat terhubung ke internet, sehingga data sistem monitoring dapat diakses pada administrator saat tidak berada ditempat.
4. Mencoba penelitian dengan menggunakan metode dan arsitektur yang berbeda agar dapat dibandingkan hasil dari penelitian selanjutnya.

DAFTAR PUSTAKA

[1] Kuswanto, D. (2014). Unjuk Kerja Intrusion Prevention Sistem Berbasis Suricata Pada Jaringan Lokal Area Network.

[2]. Ariyadi, T. (2012). Implementasi Intrusion Prevention System (IPS) Pada Jaringan Komputer Kampus B Universitas Bina Darma.

[3]Karen Scarfone, P. M. (2007). *Guide to Intrusion Detection and Prevention System (IDPS).* Gaithersburg, Maryland: National Institute of Standards and Technologi.

[4]Sourcefire. (2003). *Snort*. Retrieved from Sourcefire: www.sourcefire.com

[5]N.Purdy, G. (2004). *Firewalls, NAT & Accounting (LINUX IPTABLE).* United State of America: O'Reilly Media, Inc.