# Analisis Kolaborasi IDS SNORT dan HoneyPot

**[1)] Pentrani Lanjarasih Restanti, [2)] Dr St. Dwiarso Utomo, SE, M.Kom, Akt**

Program Studi Sistem Informasi
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro
Jl. Nakula I No. 5-11, Semarang, Indonesia
Email : [1)] 112201003941@mhs.dinus.ac.id [2)] dwi777utomo@gmail.com

***Abstract***

*Snort is one tool in the IDS with Snort open source community to be the tool of choice in securing computer networks. Ease of understanding the rules in Snort and the ease in making the signature is also an advantage possessed by Snort. While Honeyweb is one honeypot that simulates a web server also have some signature attacks that have been identified and have the ability to detect new attacks that are recorded in a log file that provides information to network administrators to secure web server. With this research could examine the existing collaboration in Honeypot implementation techniques that exist today. Data were collected using field studies and literature. Library studies include observations, interviews and literature. The results of this study are Honeyweb and Snort is able to increase the security of the IDS, in this case the web server.*

***Keywords :*** *Nework security, IDS, Honeypot, Honeyweb, Snort.*

[1)]Mahasiswa Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
[2)] Wakil Rektor 2, Universitas Dian Nuswantoro

## Abstrak

*Snort* merupakan salah satu *tool* pada *IDS* dengan komunitas *opensource* sehingga *Snort* menjadi *tool* pilihan dalam mengamankan jaringan komputer. Kemudahan memahami *rules* pada Snort dan kemudahan dalam membuat *signature* juga merupakan keunggulan yang dimiliki oleh *Snort*. Sedangkan *Honeyweb* merupakan salah satu *Honeypot* yang mensimulasikan *webserver* juga mempunyai beberapa *signature* serangan yang sudah dikenali dan mempunyai kemampuan mendeteksi beberapa serangan baru yang dicatat dalam *filelog* yang memberikan informasi kepada administrator jaringan untuk mengamankan *webserver*nya. Dengan penelitian ini dapat mengkaji kolaborasi yang ada dalam teknik implementasi Honeypot yg ada saat ini. Data dikumpulkan dengan menggunakan metode studi lapangan dan studi pustaka. Studi Pustaka mencakup Observasi, wawancara dan Literatur. Hasil penelitian ini yaitu *Honeyweb* dan *Snort* mampu meningkatkan pengamanan pada IDS, dalam hal ini *webserver*.


***Kata kunci*** : *Keamanan jaringan, IDS, Honeypot, Honeyweb, Snort.*

## 1. Pendahuluan

Keinginan untuk cepat mengakses informasi mengenai sistem telah memaksa teknologi berkembang dengan cepat. Namun terkadang sistem tersebut tidak digunakan oleh orangyang tepat, misalnya pemilik sistem atau orang yang berhak untuk mengelola sistem. Banyak hal untuk melakukan pengamanan, yaitu diperlukannya perangkat yang mendukung serta sumber daya manusia yang mampu dalam menyelesaikan setiap masalah keamanan dari system informasi itu. Banyak aspek yang bisa mengancam keamanan sistem tersebut, yaitu ancaman yang bersifat *interruption* dimana informasi dan data dalam sistem dirusak dan dihapus sehingga jika dibutuhkkan, data atau informasi tersebut tidak ada lagi. Kemudian ancaman yang bersifat *interception* yaitu informasi yang ada disadap oleh orang yang tidak berhak mengakses informasi yang terdapat pada system ini. Selanjutnya modifikasi yaitu ancaman terhadap integritas dari *system* informasi tersebut dan yang terakhir adalah *fabrication* yaitu orang yang tidak berhak berhasil memalsukan suatu informasi yang ada sehingga orang yang menerima informasi tersebut menyangka bahwa informasi tersebut berasal dari orang yang dikehendaki oleh penerima informasi tersebut

Dengan membuat berbagai *rules* untuk mendeteksi ciri-ciri khas (*signature*) dari berbagai macam serangan, maka *Snort* dapat mendeteksi dan melakukan *logging* terhadap serangan-serangan tersebut. *Software* ini bersifat *opensource* berdasarkan *GNU General Public License* [*GNU89*], sehingga boleh digunakan dengan bebas secara gratis, dan kode sumber (*source code*) untuk *Snort* juga bisa didapatkan dan dimodifikasi sendiri bila perlu. *Snort* pada awalnya dibuat untuk sistem operasi (*operating system*) berdasarkan *Unix,* tetapi versi *Windows* juga sudah dibuat sehingga sekarang ini *Snort* bersifat *cross-platform*. *Snort* sendiri merupakan *software* yang masih berbasis *command-line*, sehingga cukup merepotkan bagi pengguna yang sudah terbiasa dalam lingkungan *Graphical UserInterface* (*GUI*). Oleh karena itu, ada beberapa *software* pihak ketiga yang memberikan *GUI* untuk *Snort*,

misalnya *IDS Center* untuk *Microsoft Windows*, dan *Acid* yang berbasis *PHP* sehingga bisa diakses melalui *web browser*.

*Snort* memiliki bahasa pembuatan *rules* yang relatif mudah dipelajari danfleksibel. Ini berarti bahwa pengguna dapat dengan mudah dan cepat membuat berbagai *rules* baru untuk mendeteksi tipe-tipe serangan yang baru. Selain itu, berbagai *rules* khusus dapat dibuat untuk segala macam situasi. *Snort* sudah memiliki sebuah *database* untuk berbagai macam *rules*, dan *database* ini secara aktif terus dikembangkan oleh komunitas *Snort* sehingga tipe-tipe serangan yang baru dapat dideteksi dan dicatat.

## 2. Tinjauan Pustaka

*Honeypot* merupakan sebuah sistem atau komputer yang sengaja "dikorbankan" untuk menjadi target serangan dari *hacker*. Konsep serupa *honeypot* (sebelumnya belum diberi istilah seperti itu) dipercaya sudah cukup lama ada, walaupun tidak ada literatur yang membahasnya sebelum tahun 1990. Tahun 1990 Clifford Stoll menerbitkan buku The Cuckoo's Egg, yang lebih mirip cerita detektif. Penerbitnya Pocket Books, yang lebih dikenal dengan novel. Inilah penerbitan pertama yang menguraikan konsep *honeypot*. Buku ini menceritakan kejadian sesungguhnya selama periode sepuluh bulan di tahun 1986-1987. Stoll adalah astronom pada Lawrence Berkeley Lab yang menjadi admin berbagai komputer untuk komunitas astronom. Selisih akuntansi senilai 75 sen membuatnya menyadari akan adanya hacker bernama '*Hunter*,' yang telah menyusup ke dalam sistem [1].

Bukannya menutup account penyusup ini, Stoll malah membiarkannya berada dalam sistem, agar dapat mempelajarinya lebih jauh dan memburunya. Tanpa disadari penyerang, Stoll menyiapkan direktori *SDINET* (*Strategic Defence Initiave Network*) dan mengisinya dengan file-file yang pura-pura berisi berbagai file keuangan dan rahasia negara. *Hacker* ini ternyata tidak tertarik pada file-file keuangan. Makalah teknis pertama mengenai honeypot terbit pada tahun 1990 itu juga, tulisan Bill Cheswick berjudul '*An Evening with Berfeld in Which a Cracker Is Lured, Endured and Studied*' .Berbeda dengan yang pertama, Cheswick memang menyiapkan suatu sistem yang memang untuk diserang, menjadikannya kasus pertama dari *honeypot* yang sesungguhnya. Pada makalah ini Cheswick bukan saja membahas cara membangun dan menggunakan *honeypot*, melainkan juga menceritakan bagaimana seorang *hacker* Belanda dipelajari sewaktu dia menyerang dan menguasai sistem. Cheswick pertamatama membangun suatu sistem dengan beberapa kelemahan (termasuk *Sendmail*) untuk mendapatkan ancaman apa saja yang ada dan bagaimana cara kerjanya [2]. Tujuannya bukanlah untuk

menangkap orang tertentu, melainkan untuk mempelajari kegiatan membahayakan apa saja yang bisa terjadi terhadap network dan sistemnya. Cheswick menciptakan suatu lingkungan terkontrol yang disebutnya sebagai 'jail' (ia tidak menyebutnya sebagai honeypot), yang mengurung kegiatan sang penyerang. *Hacker* Belanda dengan nickname Berfeld ini memasuki sistem dengan memanfaatkan kelemahan pada *Sendmail* sampai mendapatkan kendali terhadap sistem.Secara umum, *honeypot* dapat didefinisikan sebagai sebua sumber daya sistem informasi dimana nilai guna dari sumber daya tersebut justru berdasar kepada terdeteksinya kasus penggunaan yang tidak terotorisasi atau tidak diperbolehkan secara hukum dari sumber daya tersebut. Atau dengan kata lain, *honeypot* adalah sebuah sumber daya yang bersifat seakan-akan target yang sebenarnya, yang dengan sengaja disediakan untuk diserang atau diambil alih. Oleh karena itu, *honeypot*akan diamati, diserang bahkan dieksploitasi oleh penyerang atau penyusup. Tujuan utama dari *honeypot* ini adalah untuk mengumpulkan informasi dari suatu serangan dan penyerang yang melakukannya. *Intruder* atau penyerang merupakan istilah umum yang diberikan untuk menggambarkan seseorang yang berusaha untuk masuk ke dalam sistem dalam arti berusaha menggunakan sistem dimana mereka tidak memiliki autorisasi atau menggunakan sistem untuk maksud yang menyimpang di luar hak-hak yang mereka miliki.

Menurut Lance Spitzner, seorang arsitek sistem keamanan *Sun Microsystems*, "*A honeypot is security resource whose value lies in being probed, attacked, or compromised*." Definisi ini menjadi acuan beberapa makalah lainnya. Dari definisi itu dapat diambil kesimpulan bahwa *honeypot* baru dikatakan suatu sistem keamanan jika *honeypot* tersebut disusupi, diserang, atau dikendalikan oleh penyerang.Ada juga seorang insinyur berkebangsaan *Swiss* bernama Reto Baumann menyikapi interpretasi yang diberikan oleh Lance Spitzner. Menurut Baumann melalui tugas akhir diplomanya, "*A honeypot is a resource which pretends to be a real target. A honeypot is expected to be attacked or compromised. The main goals are the distraction of an attacker and the gain of information about an attack and the*

*attacker*." Jadi, menurut Baumann, *honeypot* adalah sebuah sumberdaya sistem keamanan yang dibuat sebagai tujuan utama penyerang yang sebenarnya merupakan sistem yang palsu untuk menjebak penyerang.Sistem *honeypot* biasanya hanya sebuah sistem yang dihubungkan dengan jaringan produktif, atau sistem yang asli, yang ada dengan tujuan untuk menjebak penyerang. Sistem tersebut kemudian dapat mengemulasikan berbagai variasi sistem atau lubang-lubang dari sistem yang mudah untuk diserang.[5]

Komputer tersebut melayani setiap serangan yang dilakukan oleh *hacker* dalam melakukan penetrasi terhadap *server* tersebut. Metode ini ditujukan agar *administrator* dari server yang akan diserang dapat mengetahui trik penetrasi yang dilakukan *hacker* serta bisa melakukan antisipasi dalam melindungi server yang sesungguhnya. Setiap tindakan yang dilakukan oleh penyusup yang mencoba melakukan koneksi ke *honeypot* tersebut, maka *honeypot*akan mendeteksi dan mencatatnya.

Perlu diingat bahwa peran dari *honeypot* bukanlah menyelesaikan suatu masalah yang akan dihadapi server, akan tetapi memiliki kontribusi dalam hal keseluruhan keamanan. Dan besarnya kontribusi tersebut tergantung dari bagaimana kita menggunakannya. Intinya, walaupun tidak secara langsung melakukan pencegahan terhadap serangan *(firewall)* tetapi dapat mengurangi dari intensitas serangan yang akan dilakukan oleh penyusup ke *server* yang sesungguhnya.

Tidak satupun perangkat yang memiliki kesempurnaan dalam penggunaannya. Demikian juga dalam hal melakukan defensif yang dilakukan oleh perangkat untuk melindungi *server* dari serangan yang akan terjadi. Perangakat tersebut tidak bisa menghilangkan serangan, akan tetapi memperkecil risiko serangan. Begitu pula peran dari *honeypot*, *honeypot* sangat berguna dalam hal pendeteksian terhadap serangan bukan menghilangkan serangan itu.

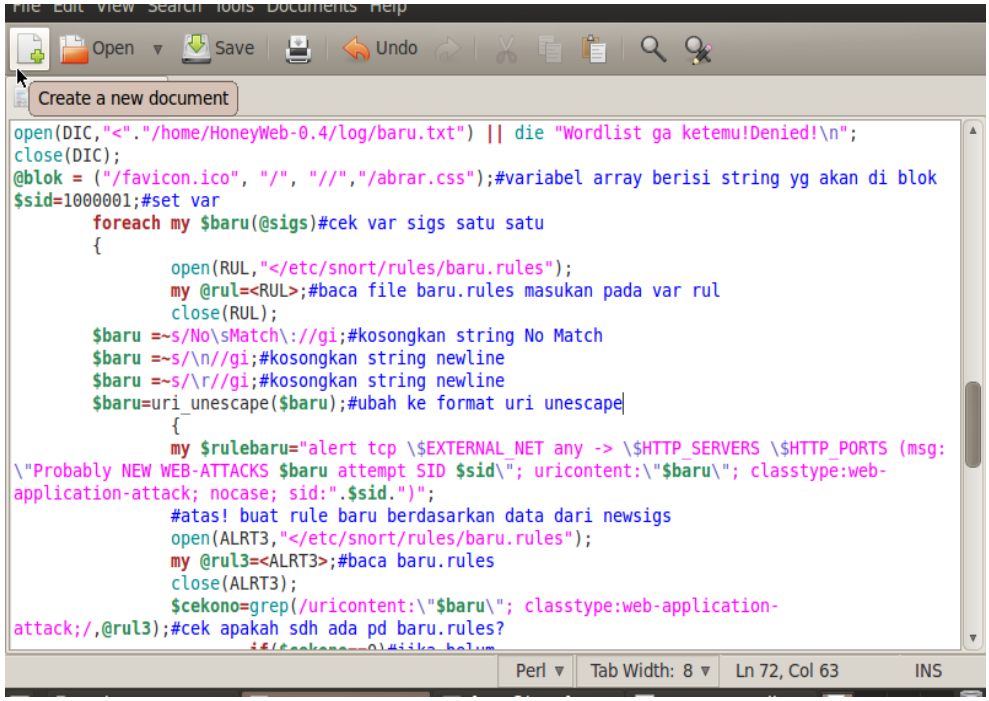
Honeypot bisa mengetahui aktivitas dari *script kiddies* sampai *hacker* sekalipun, dari duni luar maupun yang berasal dari janringan internal alias orang dalam. Sekaligus mengetahui motivasi hacking yang akan dilakukan,

sehingga administrator dapat membuat suatu kebijakan keamanan *(firewall)* dalam hal mngurangi resiko yang akan terjadi terhadap server yang sesungguhnya.

## 3. Metode dan Perancangan Sistem

Penelitian yang dilakukan, diselesaikan melalui tahapan penelitian yang terbagi dalam lima tahapan, yaitu: (1) Obyek Penelitian, (2) Jenis Data, (3) Ruang Lingkup Penelitian, (4) Pengumpulan Data, (5) Sumber Data, (6) Teknik Analisis Data (7) Penulisan Laporan.

## 4. Hasil dan Pembahasan




**Gambar 4.1 *Generate Rule* Baru dari *Honeyweb* untuk *Snort* oleh *Script Autorule***

Hasil pada *script* proses generate *rule snort* ini menggunakan metode *multithreads*, dengan metode ini pencipta yang mengharapkan dapat menambah kemampuan menambah kecepatan dan pemanfaatan *resource* dari *prosessor server* yang sudah mendukung *multicore*. Sehingga memungkinkan beberapa instruksi yang banyak dari *script* dapat dilakukan

bersamaan, mengingat pada suatu pengamaman jaringan dibutuhkan sistem yang cepat. Jadi pemilihan penggunaan *multithreads* saya anggap tepat.

Setelah dilakukannya wawancara dengan Pencipta *script*, dihasilkan ide awal penggabungan *snort* dan *honeyweb* ini adalah Pencipta meneliti, menguji dan melakukan serangan web dan menggunakan *IDS snort*, masih banyak yang tidak terdeteksi oleh *IDS* ini. Untuk itu, pencipta mencoba mengeksplor lebih lanjut bagaimana agar kelemahan ini dapat tertutup akan tetapi secara otomatis tanpa mengidentifikasi secara manual.

Kemudian dilakukan ujicoba identifikasi serangan web, dan ditemukannya masi terdapat lubang pada penggunaan *rule default snort*, pencipta mencoba mencari aplikasi yang juga dapat mengidentifikasi serangan *web*, tapi jika level yang digunakan sama-sama hanya bersifat detektor, dianggap kurang. Untuk itu pencipta gunakan salah satu varian dari *honeypot*. Selain dalam *log*nya dapat diambil untuk refrensi untuk *rule*, fungsi dari *honeypot* sendiri sebagai media jebakan pun dapat menjadi nilai tambah.

Dalam penggabungan ini, yang masih dikedepankan sementara untuk menambah identifikasi sektor web, diharapkan nantinya dilakukannya penggabungan berikutnya untuk mengamankan port selain port 80 milik *webserver*, kemudian ketika adanya serangan dari *port scan* dan *DDos* masih dirasa lambat, perlu digunakan *server* yang lebih bisa menghendel, tapi ini secara teknis, buka efek dari *script*. Penulis berangan agar nantinya dikembangkan untuk penggunaan interface dan penggabungan selain *honeyweb*.

Setelah proses tanya jawab singkat yang dilakukan, kemudian diadakannya percobaan langsung dengan dilakukannya serangan dengan metode offline manual (melalui aplikasi) dan secara online, dimaksudkan dengan pengujian, diharapkan penggunaan pengamanan secara nyata dapat optimal. Berikut data hasil pengamanan serangan :

**Tabel 4.1 Hasil Pendeteksian *Snort* dalam 10 Detik**

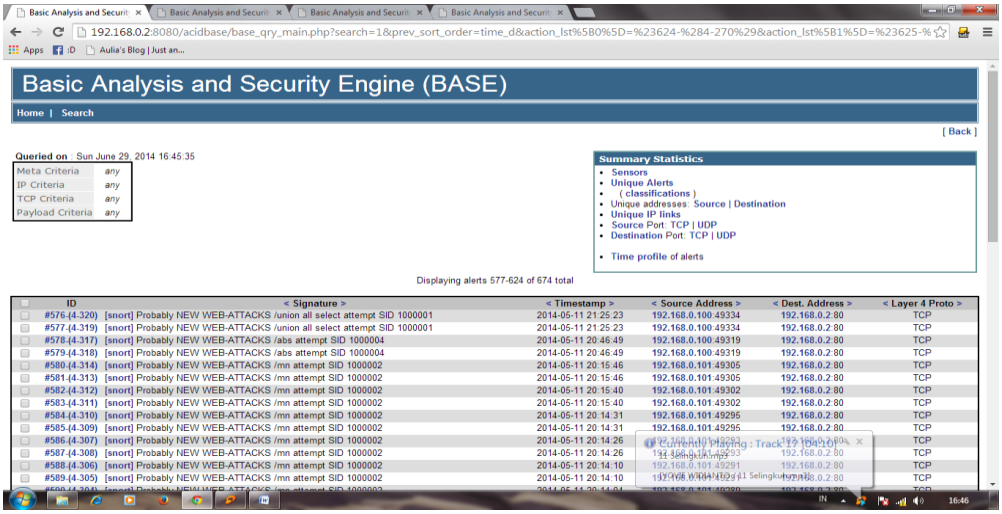
| Percobaan | Pendeteksian *Snort* Sebelum terbentuk *rule* (10s) | | |
|---|---|---|---|
| Ke- | *havij 1.15 pro* | *Loic* | 12x Manual |
| 1 | 23 Serangan | 341 Serangan | 2 Serangan |
| 2 | 18 Serangan | 235 Serangan | 0 Serangan |
| 3 | 18 Serangan | 2685 Serangan | 0 Serangan |

Hasil yang ditulis pada Tabel 4.1 adalah hasil pendeteksian *Snort* dalam suatu serangan yang masih menggunakan *rule default*.

**Tabel 4.2 Pendeteksian Oleh *Honeyweb***

| Percobaan | Pendeteksian Serangan Oleh *Honeyweb* | | |
|---|---|---|---|
| Ke- | *havij 1.15 pro* | *Loic* | 12x Manual |
| 1 | 39 Serangan | 0 Serangan | 8 Serangan |
| 2 | 21 Serangan | 0 Serangan | 2 Serangan |
| 3 | 21 Serangan | 0 Serangan | 15 Serangan |

Pada penelitian ini dilakukan pengaktifan oleh dua aplikasi yaitu *Snort* dan *Honeyweb* sekaligus agar dapat dibandingkan hasil yang didapat.




**Pada Gambar 4.2 menandakan penggunaan *rule* baru hasil *generate* dapat sepenuhnya digunakan untuk identifikasi sebuah serangan.**

Pada Gambar 4.2 penggunaan *rule* yang dihasilkan terbukti telah berhasil digunakan untuk proses identifikasi.

**Tabel 4.3 Pendeteksian *Snort* Setelah Penambahan *Rule* Baru**

| Pendeteksian *Snort* Sesudah terbentuk *rule* baru (30s) | | | | | |
|---|---|---|---|---|---|
| *havij 1.15 pro* | *Rule* terbentuk | *Loic* | *Rule* terbentuk | 12x Manual | *Rule* terbentuk |
| 32 Serangan | 2 *Rules* | 342 Serangan | 0 *Rule* | 10 Serangan | 4 *Rules* |
| 21 Serangan | 3 *Rules* | 235 Serangan | 0 *Rule* | 7 Serangan | 2 *Rules* |
| 24 Serangan | 3 *Rules* | 2693 Serangan | 0 *Rule* | 15 Serangan | 5 *Rules* |

Terbukti dari Tabel 4.3 Setelah terciptanya *rule* baru, *snort* mampu mengidentifikasi lebih banyak dengan memanfaatkan *rule* yang tercipta.

**Tabel 4.4 *IP Public* yang Tertangkap dari Hasil Percobaan *Online***

| Percobaan Ke - | *IP Public* dari Penyerang yang Terdeteksi | | | |
|---|---|---|---|---|
| 1 | 172.45.34.xxx | 107.182.54.xxx | 112.215.43.xxx | 112.215.34.xxx |
| 2 | 85.40.135.xxx | 112.215.34.xxx | | |
| 3 | 172.45.48.xxx | 112.215.34.xxx | 112.215.36.xxx | |
| 4 | 132.32.111.xxx | 112.215.34.xxx | | |

Pada Tabel 4.4 adalah *IP Public* yang tertangkap melakukan percobaaan serangan.

**Tabel 4.5 Hasil Percobaan Online**

| Waktu Pengujian | Klasifikasi Serangan | | | Total *Rule* |
|---|---|---|---|---|
| | *Port Scan* | *IP Flood* | *Web & SQL Attack* | Terbentuk |
| 3 Hari | 134 Serangan | 0 Serangan | 3 Serangan | 1 *Rule* |
| 5 Hari | 453 Serangan | 54 Serangan | 1 Serangan | 0 *Rule* |
| 3 Hari | 23 Serangan | 2102 Serangan | 8 Serangan | 3 *Rules* |
| 1 Hari | 83 Serangan | 13 Serangan | 3 Serangan | 2 *Rules* |

Dari hasil percobaan yang dilakukan pada tabel 4.5 bahwa tercipta 19 *rules* berbeda.

Dari hasil sementara penelitian awal didapatkan bahwa :

1. Aplikasi pengamanan data menggunakan dua buah teknik yaitu *SNORT* dan *HoneyWeb*, fungsi kedua teknik ini yaitu berbagi informasi mengenai serangan pada data, sehingga data yang telah dipantau dapat terjamin keamanannya.

2. Pada *HoneyWeb* digunakan bahasa pemrograman *python*, sedangkan untuk mengintegrasikan kemampuan yang dimiliki oleh *SNORT* dan *HoneyWeb* maka digunakan bahasa pemrograman *perl*.
3. *Sharing* pengetahuan antara *SNORT* dan *HoneyWeb* dapat dilihat pada *autorule.pl*.
4. Implementasi *HoneyWeb* dilakukan menggunakan *Browser*.

## 4.1 Analisis dan Pembahasan

*Sharing* pengetahuan yang dilakukan oleh *snort* dan *honeyweb* memang dianggap perlu penggunaan aplikasi ke-3, namun aplikasi ini bukan menjadi penggalang dalam optimalitas dari kecepatan dan keakuratan di masing-masing.

Pada Gambar 4.1 terlebih dahulu *script* akan mengambil *newsign* dari *honeyweb* yang akan dijabarkan sebagai pengklasifikasian menurut kebutuhan yang nantinya dapat di tambahkan pada *rule snort*. Kemudian penggunaaan identifikasi serangan, pencipta menggunakan *uricontent*, menurut penulis penggunaan metode ini dalam melakukan identifikasi diharapkan detail. Untuk panduan format penulisan dari *rule snort* dapat didapatkan di panduan manual yang disediakan oleh *snort*. Setelah pengklasifikasian lengkap, selanjutnya *script* akan mengisi di format penulisan *rule snort* dan secara langsung akan disimpan pada *baru.rule* dan untuk penggunaan *rule* ini, harus diadakannya seting dalam *snort.conf* dengan ditambahkannya nama *rule* yang ingin digunakan

Tabel 4.1 adalah hasil pendeteksian *Snort* dalam suatu serangan yang masih menggunakan *rule default*. Pada percobaan ini kami gunakan aplikasi bernama *havij 1.15 pro*, manfaat aplikasi ini digunakan untuk *SQL injection*, *Loic* aplikasi ini digunakan untuk mencoba serangan berupa *DDos*, dan Penulis mencoba menggunakan serangan secara manual dengan

menggunakan browser. Hasil ini yang nantinya akan dibandingkan dengan hasil pendeteksian dari *Honeyweb*. Pada pendeteksian serangan ini yang

masih menggunakan *rule default,* menunjukkan bahwa ada beberapa *rule* yang sudah dapat digunakan sebagai parameter dalam penentuan serangan.

Pada Tabel 4.2 adalah hasil pendeksian oleh *Honeyweb* yang waktu proses serang diambil dari serangan yang sama dengan hasil pada Tabel 4.1. Tabel 4.2 menunjukkan perbedaan hasil antara dua proses *monitoring* ini. Dari hasil suatu penelitian yang sama, *honeyweb* dapat mengidentifikasi serangan lebih banyak. Dengan ini data dari *honeyweb* bisa digunakan pembanding jika *Snort* belum biasa mengidentifikasi jenis serangan tertentu pada serangan khususnya *web* dan *SQL*.

Pada Gambar 4.2 penggunaan *rule* yang dihasilkan terbukti telah berhasil digunakan untuk proses identifikasi. Penentuan *SID* sebagai pengenal serangan pun telah tepat dengan menggunakan standar yang telah diatur dalam manual *snort.* Dari hasil pengujian serangan, pengklasifikasian bekerja tanpa mengurangi kinerja *honeyweb* dan *snort*.

Jika terbukti *snort* belum bisa mengidentifikasi serangan tertentu namun *honeyweb* bisa melakukan identifikasi, nantinya *script autorule.pl* akan mengambil *signature log newsign.txt* milik *honeyweb* dan dijadikan refrensi *rule* baru untuk *snort.* Setelah *rule* baru di *snort* tercipta, pengetahuan tentang jenis serangan bertambah. Pada serangan berikutnya *snort* sudah dapat mengidentifikasi lebih banyak serangan. Terbukti dari Tabel 4.3 Setelah terciptanya *rule* baru, *snort* mampu mengidentifikasi lebih banyak dengan memanfaatkan *rule* yang tercipta.

Setelah dilakukannya serangan serara *offline* dan sudah diketahui bahawa *rule* baru dapat berfungsi dan didapatkan hasil, Penulis mengambil inisiatif untuk mencoba melakukan pengetesan secara *online* untuk mengukur sejauh mana dengan metode ini layak untuk dimanfaatkan untuk pengamanan. Setelah dilakukan pengujian dengan tenggang waktu yang sudah tertera pada Tabel 4.5, Pada Tabel 4.4 adalah *IP Public* yang tertangkap melakukan percobaaan serangan.

Dari hasil percobaan yang dilakukan pada tabel 4.5 bahwa tercipta 19 *rules* berbeda. Hal ini membuktikan adanya kemungkinan efektifitas pengembangan *rule* dari penggabungan.

## 5. Simpulan

Pada aplikasi Pengamanan data menggunakan SNORT dan HoneyWeb ini dapat ditarik kesimpulan sebagai berikut :

1. Berdasarkan implementasi aplikasi yang telah dibuat, membuktikan bahwa SNORT dan HoneyWeb merupakan 2 (dua) buah metode yang cocok untuk diimplementasikan pada aplikasi pengamanan data.
2. Berdasarkan hasil implementasi, aplikasi dapat dikategorikan bahwa aplikasi ini dapat menciptakan keadaan aman pada server.
3. Proses deteksi serangan berjalan lancar, dimana IP penyerangan dapat dilihat melalui log pada file IP.txt dan pesan.txt.

Pada penelitian ini, algoritma yang dimodifikasi baik secara umum dalam melakukan proses pengamanan data. Dapat dilihat pada beberapa percobaan yang telah dilakukan diatas, dimana serangan akan kembali dilihat setiap jam guna menjamin kemaman.

## 6. Daftar Pustaka

[1] Rafiudin, Rahmat. 2010. Mengganyang Hacker Dengan Snort. Yogyakarta: Penerbit Andi.

[2] Firrar. U. 2006. Trik Menjebak Hacker Dengan Honeypot. Yogyakarta: Penerbit Andi.

[3] Anjik, Sukamaaji, S.Kom. Rianto, S.Kom.2008. Konsep Dasar Pengembangan Jaringan Dan Keamanan Jaringan. Yogyakarta: Penerbit Andi.

[4] Ariyus, Dony, M.Kom. 2007. Intrusion Detection Sytem : Sistem Pendeteksi Penyusupan Pada Jaringan Komputer. Yogyakarta: Penerbit Andi.

*[5]* *Honeypot*, http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=515:honeypot&catid=10:jaringan&Itemid=14/ (Diakses pada 20 Maret 2014)

[6] Paradikma Internet security, Joshi, R., & Sardana, A. (2011). Honeypots: A New Paradigm to Information Security. Science Publisher.

*[7]* *Dionaea*, http://dionaea.carnivore.it/ (Diakses pada 21 Maret 2014)

[8] Serangan Distributed Denial of Service (DDos), http://www.infokomputer.com/fitur/41-sekuriti/3557-kamus-keamanan-komputer?showall=1 (Diakses pada 20 Maret 2014)

[9] Jurnal PA Implementasi *Honeypo*t Dengan Menggunakan Dionaea, Di Jaringan Hotspot Fizz, http://repository.politekniktelkom.ac.id/Proyek%20Akhir/TK/JURNAL%20PA%20IMPLEMENTASI%20HONEYPOT%20DENGAN%20MENGGUNAKAN%20DIONAEA%20DI%20JARINGAN%20HOTSPOT%20FIZZ.pdf (Diakses pada 22 Maret 2014)

[10] Jurnal Tren Kemanan Internet 2011, http://library.binus.ac.id/eColls/eThesis/Bab1/2012-1-00635-sk%201.pdf (Diakses pada 22 Maret 2014)