Security Information and Event Management (SIEM) Untuk Mengidentifikasi Serangan Siber Pada *Web* Berstandar OWASP

1st Tia Rahmawati
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
tiarahmawati1812@gmail.com

2nd Nyoman Bogi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
aditya@telkomuniversity.ac.id

3rd Sofia Naning
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sofiananing@telkomuniversity.ac.id

Abstrak* — Pelaku kejahatan siber biasanya menyerang suatu sistem keamanan informasi yang memuat data-data penting dan rahasia. Biasanya penyerangan yang sering terjadi dilakukan terhadap aplikasi *web*. Banyak pengembang aplikasi *web* yang kurang memperhatikan sisi keamanan aplikasi *web* sehingga banyak dieksploitasi oleh para *hacker*. *Security Information and Event Management* (SIEM) yang digunakan akan memantau dan mengumpulkan semua *log report.* SIEM akan memproses *log* untuk menganalisis keamanan. Semua data yang didapatkan telah dipresentasikan dalam bentuk visual seperti grafik. Kemudian data akan dikirimkan ke *plugin* agar pengguna lebih mudah mengakses *log report.

***Kata kunci*— SIEM**

I. PENDAHULUAN

Seiring dengan perkembangan teknologi informasi yang semakin pesat dan dinamis, tingkat kejahatan siber juga semakin meningkat dewasa ini. Kejahatan siber (*cyber crime*) yang dimaksud ialah percobaan serangan terhadap suatu keamanan sistem informasi. Menurut Ditjen Aptika Kominfo, aktivitas yang dilakukan oleh sekelompok orang yang ingin menembus suatu sistem keamanan bertujuan untuk mendapatkan, mengubah, mencari, dan bahkan menghapus informasi yang ada pada sistem tersebut jika memang benar-benar dibutuhkan [1].

Pelaku kejahatan siber biasanya menyerang suatu sistem keamanan informasi yang memuat data-data penting dan rahasia. Biasanya penyerangan yang sering terjadi dilakukan terhadap aplikasi *web*. Banyak pengembang aplikasi *web* yang kurang memperhatikan sisi keamanan aplikasi *web* sehingga banyak dieksploitasi oleh para hacker. Menurut hasil *Web Security Report SiteLock* pada tahun 2022 , didapatkan 172 serangan dalam sehari untuk satu website dan *website* diakses oleh bot sebanyak 2306 kali per minggu. Bot digunakan oleh *hacker* untuk mencari kelemahan situs *web*. Jumlah serangan di tahun 2022 meningkat sebanyak 210% dibandingkan tahun 2020. Terdapat 2 jenis kerentanan tertinggi yang tercatat yaitu *Cross Site Scripting* (XSS) sebanyak 1 juta halaman *website*, dan SQL *Injection* sebanyak 332 ribu halaman website. Untuk mengatasi berbagai permasalahan terkait keamanan aplikasi web diperlukan suatu sistem yang dapat mencegah serangan berbahaya [6]-[8].

SIEM (*Security Information and Event Management*) adalah sistem yang digunakan untuk mengumpulkan, menganalisis, dan memantau aktivitas keamanan dari sebuah jaringan atau sistem. SIEM dapat digunakan untuk mengidentifikasi ancaman keamanan seperti serangan jaringan, aktivitas tidak sah, atau kerentanan keamanan yang tidak terdeteksi sebelumnya. Penerapan SIEM sangat membantu dalam melakukan analisis *log* dari suatu *server* yang sedang berjalan. SIEM memiliki beberapa *tools* yang dapat membantu dalam menganalisis *log* dengan mengumpulkan semua informasi yang mengakses *server* yang sedang berjalan [3]-[5].

II. KAJIAN TEORI

A. IDS (*Intrusion Detection System*)

IDS (*Intrusion Detection System*) adalah sistem yang dirancang untuk mendeteksi adanya serangan atau aktivitas yang mencurigakan dalam jaringan komputer atau sistem komputer. Tujuan utama IDS adalah mengidentifikasi potensi ancaman keamanan, termasuk serangan dari luar atau dalam jaringan, dan memberikan peringatan kepada administrator jaringan atau tim keamanan terkait [1].

B. Snort

Snort adalah salah satu contoh perangkat lunak IDS (*Intrusion Detection System*) yang populer dan terbuka secara gratis. Snort dirancang untuk mendeteksi dan mencegah serangan terhadap jaringan komputer.

Snort menggunakan pendekatan *signature-based* untuk mendeteksi serangan dengan membandingkan lalu lintas jaringan dengan *database* yang berisi pola serangan yang diketahui. Pola serangan ini disebut "*rules*" dan berisi aturan-aturan yang menentukan karakteristik khusus dari serangan

yang ingin dideteksi. Snort dapat mengenali serangan yang melibatkan protokol jaringan umum seperti TCP/IP, UDP, ICMP, HTTP, dan lainnya.

III. METODE

A. Studi Literatur

Proses mencari dan menganalisis literatur yang telah ditulis tentang topik atau bidang penelitian tertentu. Ini termasuk mencari, membaca, dan menyusun literatur seperti jurnal ilmiah, buku, makalah konferensi, dan sumber lain yang berkaitan dengan topik yang sedang diteliti.

B. Tahap Perancangan Sistem

Proses di mana sistem dirancang secara menyeluruh sebelum digunakan. Tahap ini mencakup analisis kebutuhan, merancang struktur sistem, mengidentifikasi komponen utama, dan membuat rencana kerja yang jelas untuk implementasi sistem.

C. Tahap Pengujian Sistem dan Analisa

Proses di mana sistem yang telah dikembangkan atau diimplementasikan diuji secara menyeluruh untuk memastikan bahwa mereka beroperasi sesuai dengan persyaratan yang ditetapkan sebelumnya. Tahap ini melibatkan pengujian fungsionalitas, kinerja, keamanan, dan kesesuaian sistem dengan kebutuhan pengguna.

D. Troubleshooting

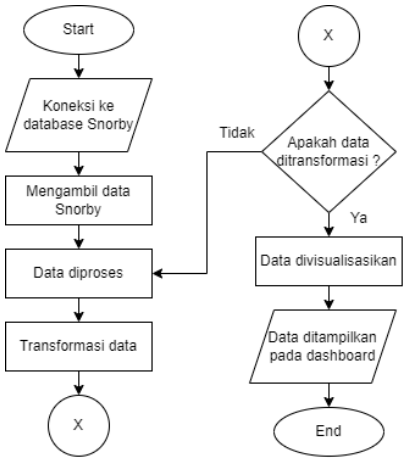
Masalah atau gangguan yang terjadi dalam sistem, perangkat, atau proses diidentifikasi, didiagnosis, dan diselesaikan melalui proses troubleshooting. Tujuan utama adalah mengatasi masalah yang terjadi agar sistem dapat berfungsi dengan baik.

E. Tahap Kesimpulan

Tahap di mana hasil dan hasil proyek atau penelitian dijelaskan dan disimpulkan. Tahap ini juga melibatkan pengambilan kesimpulan berdasarkan data dan informasi yang dikumpulkan selama penelitian.

IV. HASIL DAN PEMBAHASAN

Siem Information and Event Management yang disingkat dengan SIEM bertujuan untuk mengelola, mengumpulkan, menganalisis dan memberikan laporan terkait dengan data keamanan dari bermacam-macam sumber. Pada perancangan ini menggunakan Snorby sebagai aplikasi *web* yang menyediakan *dashboard* untuk sistem *management Intrusion Detection System* (IDS). IDS adalah sebuah sistem dirancang untuk mendeteksi aktivitas yang mencurigakan, ancaman atau serangan terhadap jaringan atau sistem komputer. Pada Gambar 1 adalah *flowchart* dari cara kerja dari snorby.



GAMBAR 1
(*Flowchart* SIEM)

Pada Gambar 1 dapat dijelaskan bahwa langkah awal adalah membuat koneksi antara Snorby dan *database* yang menyimpan data hal ini bertujuan untuk bisa mengakses data yang diperlukan. Data-data tersebut akan diambil dari *database* sesuai dengan keperluan Snorby. Kemudian data tersebut diproses yang melibatkan penggalian informasi atau pengolahan data secara umum, data diproses oleh sistem dengan mengambil beberapa informasi yang relevan dan diperlukan untuk *dashboard*. Data yang telah diproses akan diubah formatnya sesuai dengan kebutuhan visualisasi pada *dashboard.* Data yang telah diubah formatnya akan digunakan untuk membuat visualisasi seperti grafik yang akan ditampilkan pada *dashboard.*

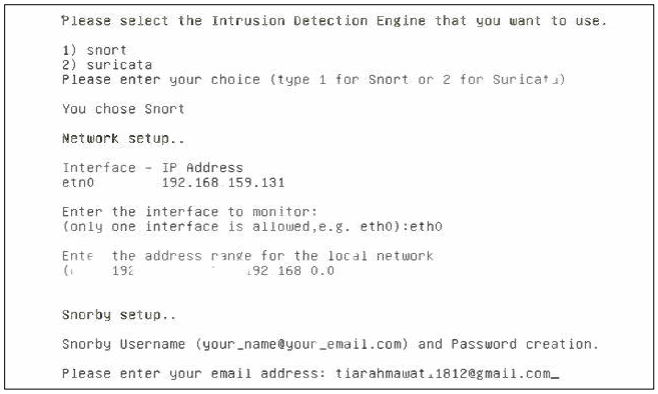
Perancangan ini memilih Snorby yang digunakan sebagai antarmuka pengguna untuk mengelola data *log* dan hasil deteksi dari sistem IDS yang terpasang di dalam *SmoothSec*.

```
*** Welcome to SmoothSec 3.4 ***

Available deployments:

One Network Interface is required.
standard     (IDS mode   All in one mode [Snorby + Sensor])

console      (IDS mode - Distributed [Only Snorby web console])

sensor       (IDS mode - Distributed [Only sensor])


Three Network Interface are required.
ips-standard (IPS mode - All in one mode [Snorby + Sensor])

ips-console  (IPS mode - Distributed [Only Snorby web console])

ips-sensor   (IPS mode - Distributed [Only sensor])


exit         (If unsure.)

Type here your favourite deployment : standard
```

GAMBAR 2
(Mode operasi)

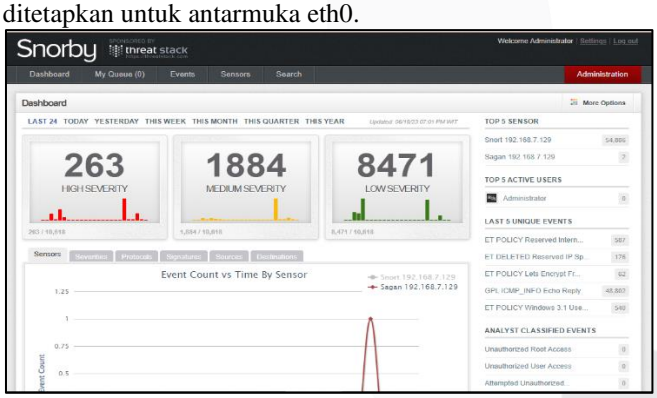
Pada Gambar 2 menjelaskan tentang mode operasi yang ditawarkan oleh *SmoothSec* dalam konteks IDS dan IPS. Pada perancangan ini memilih *one network interface standard.* Dalam mode *all in one SmoothSec* menyediakan solusi IDS dengan menggabungkan Snorby dan sensor IDS dalam satu entitas yang berjalan pada satu antarmuka jaringan.

Dalam konfigurasi ini, satu antarmuka jaringan pada *SmoothSec* digunakan untuk memonitoring lalu lintas keamanan yang masuk dan keluar. Antarmuka ini juga sebagai titik pemasangan sensor IDS untuk mendeteksi serangan yang mencurigakan.



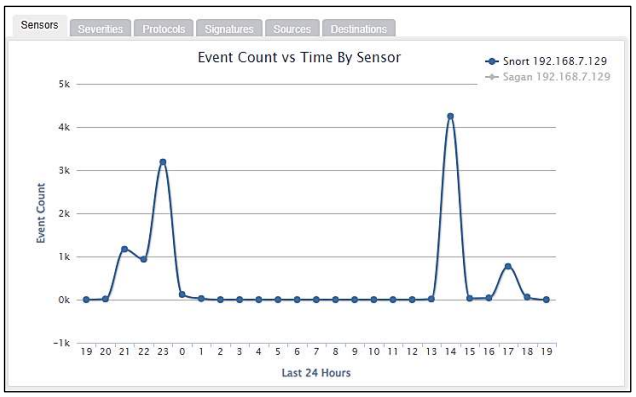
GAMBAR 3
(*Network Setup*)

Pada Gambar 3 memasukkan nomor 1 untuk pilihan sebagai *Intrusion Detection* yang digunakan pada perancangan ini. Kemudian mengatur *network setup. Interface* eth0 sebagai antarmuka jaringan yang dipilih untuk menghubungkan *computer* ke jaringan lokal atau *internet*. Diminta untuk memasukkan *address range local network* dengan alamat 192.168.0.0 yang merujuk pada alamat IP yang digunakan dalam suatu jaringan lokal. Dalam hal ini 192.168.0.0 adalah alamat jaringan yang ditetapkan sebagai alamat jaringan basis atau alamat awal. *Address range* ini bagian dari alamat IP yang dialokasikan secara khusus untuk jaringan lokal atau IP *private*. Alamat IP *private* digunakan secara internal dalam jaringan lokal untuk mengatur komunikasi antar perangkat dalam jaringan tersebut. Eth0 192.168.159.131 merujuk pada pengaturan alamat IP yang ditetapkan untuk antarmuka eth0.



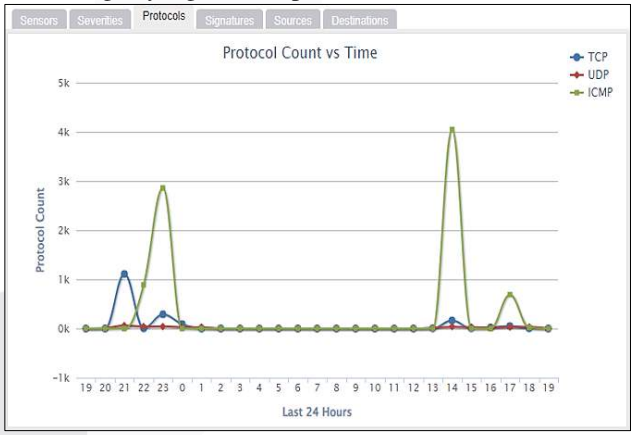
GAMBAR 4
(Snorby GUI)

Pada Gambar 4 terdapat angka 263 pada *high severity* yang berarti terdapat 263 serangan yang sangat serius dapat berpotensi merusak sistem, terdapat angka 1884 pada *medium severity* menunjukkan bahwa saat itu tercatat 1884 serangan dengan tingkat keparahan sedang. Serangan dengan tingkat keparahan sedang mencakup serangan yang memiliki dampak besar, tetapi tidak seberbahaya *high severity*. Angka 8471 pada *low severity* menunjukkan sejumlah serangan dengan tingkat keparahan terdapat 8471 insiden.



GAMBAR 5
(*Event Count vs Time by Sensor*)

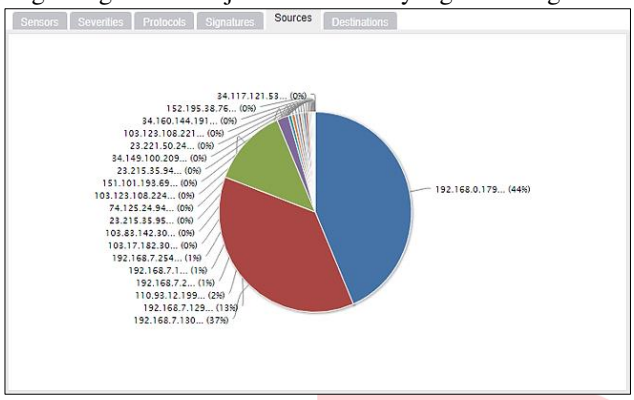
Pada Gambar 5 menggambarkan grafik *even count vs time by sensor* pada Snorby untuk representasi visual dari data yang dihasilkan oleh sensor-sensor yang terhubung dengan sistem deteksi intrusi Snorby. Snorby adalah antarmuka pengguna *web* yang digunakan untuk memantau dan menganalisis aktivitas jaringan serta mendeteksi ancaman keamanan. Grafik tersebut menggambarkan jumlah kejadian (*even count*) yang terjadi pada setiap waktu yang tercatat, berdasarkan sensor-sensor yang ada. Grafik tersebut memperlihatkan bagaimana distribusi kejadian (*even count*) berubah seiring waktu pada masing-masing sensor. Hal ini memberikan wawasan tentang pola aktivitas dan tren serangan yang mungkin terjadi dalam jaringan. Dengan menganalisis grafik ini, pengguna Snorby dapat mengidentifikasi sensor-sensor yang mungkin mengalami peningkatan aktivitas yang mencurigakan atau menemukan pola serangan yang muncul pada waktu tertentu.



GAMBAR 6
(Grafik *Protocols*)

Pada Gambar 6 memberikan gambaran tentang distribusi protokol yang digunakan dalam lalu lintas jaringan selama periode waktu tertentu. Setiap bar atau titik pada grafik mewakili jumlah kejadian yang tercatat untuk setiap protokol yang dideteksi oleh sensor di dalam snorby. Contohnya, grafik tersebut dapat menampilkan jumlah kejadian yang terkait dengan protokol HTTP, FTP, DNS, atau ICMP pada interval waktu yang ditentukan, seperti per jam, per hari, atau per minggu. Dengan melihat grafik ini, pengguna Snorby dapat memahami pola penggunaan protokol yang berbeda-beda dari waktu ke waktu. Grafik "*protocol count vs time*" membantu dalam menganalisis tren protokol yang digunakan dalam lalu lintas jaringan. Pengguna dapat mengidentifikasi protokol yang paling banyak digunakan, memantau

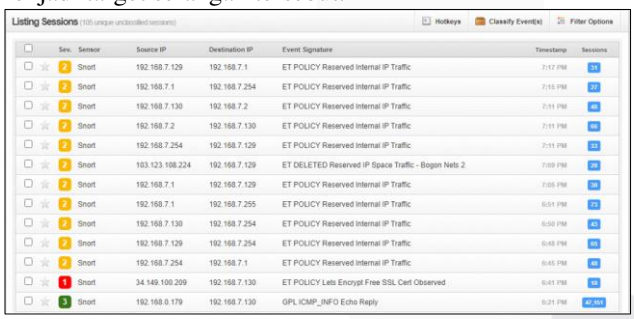
perubahan pola penggunaan protokol dari waktu ke waktu, serta mendeteksi protokol yang tidak biasa atau tidak terduga yang mungkin menunjukkan aktivitas yang mencurigakan.



GAMBAR 7
(*Signatures*)

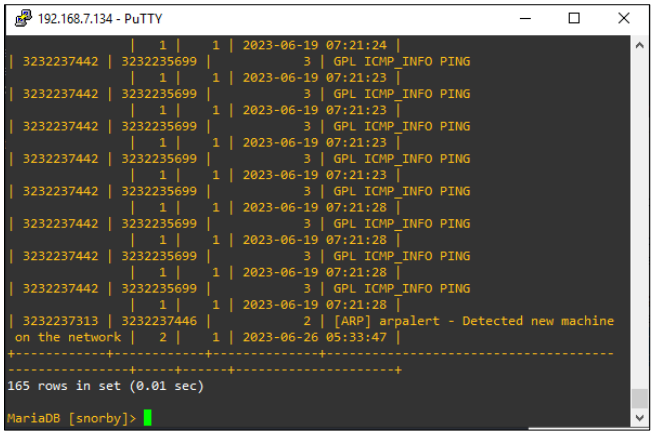
Pada Gambar 7 menunjukkan *signatures* yang mana dengan fitur ini memudahkan pengguna untuk melihat apa saja yang paling banyak terdeteksi oleh sistem. Dapat dilihat bahwa ICMP yang paling banyak terdeteksi oleh sistem dengan presentasi 56%.

Terdapat *Severity* yang berarti mengacu pada tingkat keparahan atau tingkat seriusnya suatu *event* atau kejadian yang terdeteksi oleh sistem deteksi intrusi Snorby. jika ada serangan atau aktivitas mencurigakan, *source* IP akan menunjukkan alamat IP dari mana serangan tersebut berasal dan jika ada serangan yang ditujukan ke sistem atau alamat IP tertentu, *destination* IP akan menunjukkan alamat IP yang menjadi target serangan tersebut.



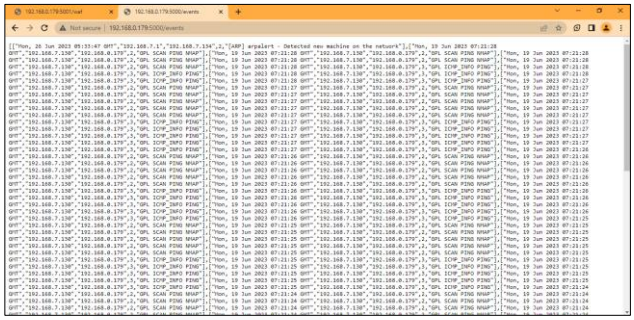
GAMBAR 8
(*Event* yang terekam oleh snorby)

Pada Gambar 8 menjelaskan hasil *event*, semua serangan akan di petakan sesuai *rules* yang ada, hanya saja snorby tidak dapat melakukan tindakan seperti memblock atau *drop* serangan.



GAMBAR 9
(*Database*)

Pada Gambar 9 menunjukan data-data pada snorby yakni *database* pada *events_with_join.* Pada data diatas dapat dilihat bahwa *source* IP dan *destination* IP menunjukan bahwa data IP masih dalam bentuk enkripsi. Kemudian data-data tersebut akan diubah dalam bentuk format JSON agar *source* IP dan *destination* IP menjadi data deskripsi sehingga *client* dapat mengakses *database* tersebut dengan perantara API.



GAMBAR 10
(JSON)

Pada Gambar 10 dapat dilihat bahwa *database* dari snorby yang masih terenkripsi dapat terhubung dengan JSON, pada *source* IP dan *destionation* IP yang sebelumnya masih dalam bentuk enkripsi pada JSON data-data tersebut sudah terdeskripsi. Untuk dapat mengakses JSON tersebut maka *client* hanya memerlukan API sebagai jembatan penghubung antara *database events_with_join* dan JSON.

Pada pengujian ini melakukan beberapa serangan seperti pengujian *ping, Distributed Denial of Service* (DDoS) dan *brute force* yang mengimplementasikan skenario serangan yang mencakup langkah-langkah penyerang Dari beberapa pengujian serangan yang telah dilakukan, Adapun datanya dapat dilihat pada Tabel 1.

TABEL 1
(Hasil Pengujian)

No.	*Testing system scenario*	*Test Tools*	*Expected results*	*Results of testing system*	*Conclusion*
1	*Ping Server*	Kali Linux	*Detected*	*Detected*	*Successful*
2	DDoS	Kali Linux	*Detected*	*Detected*	*Successful*
3	*Brute force*	Kali Linux	*Detected*	*Detected*	*Successful*

Pada Tabel 1 dapat dilihat bahwa semua pengujian sistem berjalan sesuai dengan yang diharapkan seperti hasil data

yang ditunjukkan dalam Tabel 1. Penyerang dapat diidentifikasi oleh sistem. Sistem ini telah diuji coba dan dapat mendeteksi instruksi serangan berdasarkan *rule ping*, DDoS dan *brute force*. Penambahan *rule* akan meningkatkan keamanan server dan jaringan yang dilindungi karena perkembangan teknologi dan berbagai metode penyerangan.

TABEL 2
(Hasil Pengujian Performansi)

No.	Komponen	Sebelum diserang (%)	Setelah diserang (%)
Pengujian 1			
1	*Processor* CPU	10	40
2	*Memory*	3,6	56
Pengujian 2			
1	*Processor* CPU	18	41
2	*Memory*	3,7	57
Pengujian 3			
1	*Processor* CPU	15	43
2	*Memory*	3,7	57
Pengujian 4			
1	*Processor* CPU	14	50
2	*Memory*	3,7	59
Rata-rata		8,96	50,37

Pada Tabel 2 adalah hasil pengujian performansi yang dilakukan 4 kali pengujian. Berdasarkan data tersebut, terlihat bahwa semua pengujian menghasilkan penurunan kinerja pada kedua komponen setelah diserang. Berdasarkan data rata-rata, dapat disimpulkan bahwa serangan yang dilakukan pada komponen tersebut mengakibatkan penurunan kinerja yang signifikan. Hal ini menunjukkan adanya dampak yang merugikan dari serangan tersebut terhadap sistem komputer yang diuji.

V. KESIMPULAN

Dengan adanya implementasi SIEM dengan Snorby sebagai antarmuka pengguna dan integrasi dengan SmoothSec sebagai IDS, sistem ini terbukti berhasil mendeteksi serangan yang mencurigakan dan memberikan data dan visualisasi yang berguna bagi pengguna untuk menganalisis dan meningkatkan keamanan jaringan. Meskipun performansi sistem mengalami penurunan setelah diserang, tetapi kemampuan untuk mendeteksi dan melacak serangan adalah aspek penting dalam memitigasi potensi ancaman keamanan yang lebih besar.

REFERENSI

[1] S. Ali and T. Nadeem Malik, “Intrusion Detection and Prevention against Cyber Attacks for an Energy Management System,” Mehran University Research Journal of Engineering and Technology, vol. 41, no. 1, p. 202, doi: 10.22581/muet1982.2201.20.

[2] H. Alnabulsi, M. R. Islam, and Q. Mamun, “Detecting SQL injection attacks using SNORT IDS,” in Asia-Pacific World Congress on Computer Science and Engineering, APWC on CSE 2014, Institute of Electrical and Electronics Engineers Inc., 2014. doi: 10.1109/APWCCSE.2014.7053873.

[3] Md. A. Islam and Md. M. Islam, “A Novel Signature-Based Traffic Classification Engine To Reduce False Alarms In Intrusion Detection Systems,” International journal of Computer Networks & Communications, vol. 7, no. 1, pp. 63–80, Jan. 2015, doi: 10.5121/ijcnc.2015.7105.

[4] I. Gede, W. Bangga, and S. M. Ladjamuddin, “SIMULASI SNORT SEBAGAI ALAT PENDETEKSI INTRUSI PADA WEB DAMN VULNERABLE WEB APPLICATION,” Jurnal Rekayasa Informasi, vol. 11, no. 2, 2022.

[5] D. T. Yuwono, “Analysis Performance Intrusion Detection System in Detecting Cyber-Attack on Apache Web Server,” *IT Journal Research and Development*, pp. 169–178, Feb. 2022, doi: 10.25299/itjrd.2022.7853.

[6] R. Muwardi, H. Gao, H. U. Ghifarsyam, M. Yunita, A. Arrizki, and J. Andika, “Network Security Monitoring System Via Notification Alert,” *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 1, no. 2, pp. 113–122, Nov. 2021, doi: 10.51662/jiae.v1i2.22.

[7] E. Gunadhi and A. Sudrajat, “PENGAMANAN DATA REKAM MEDIS PASIEN MENGGUNAKAN KRIPTOGRAFI VIGENERE CIPHER,” 2016. [Online]. Available: http://jurnal.sttgarut.ac.id

[8] C.-H. Yang and C.-H. Shen, “IMPLEMENT WEB ATTACK DETECTION ENGINE WITH SNORT BY USING MODSECURITY CORE RULES,” 2009.

[9] R. M. Muhammad, I. Dyah Irawati, and M. Iqbal, “Integrated Security System Implementation for Network Intrusion,” 2021.

[10] N. Novi and Z. Zaini, “Secure Socket Layer untuk Keamanan Data Rekam Medis Tumor Otak pada Health Information System,” *JURNAL NASIONAL TEKNIK ELEKTRO*, vol. 6, no. 3, p. 137, Jul. 2017, doi: 10.25077/jnte.v6n3.405.2017.