CICES (*Cyberpreneurship Innovative and Creative Exact and Social Science*)
DOI: 10.33050/cices.v10i2.3243 ISSN: **2356-5209**
pp. 229~240 *Online* ISSN: **2655-3058**

Implementasi Sistem Deteksi Intrusi Dengan Metode *Signature* Dan *Port Mirroring* Pada UPA TIK Undiksha

Gede Beny Indrawan*[1], I Ketut Resika Arthana[2], Kadek Yota Ernanda Aryanto[3]
[1,2,3]Program Studi Ilmu Komputer, Fakultas Teknik dan Kejuruan, Universitas Pendidikan Ganesha
E-mail: [*1]**beny.indrawan@undiksha.ac.id**, [2]resika@undiksha.ac.id, [3]yota.ernanda@undiksha.ac.id

Abstrak

Fasilitas yang biasa digunakan untuk menyimpan, mengelola dan menyebarkan informasi merupakan bentuk fisik dari *data center*. Peran *data center* juga digunakan untuk menjaga data pengguna agar tetap aman secara fisik maupun secara digital. Universitas Pendidikan Ganesha (UNDIKSHA) memiliki *data center* yang terletak pada gedung UPA TIK, yang berfungsi untuk melayani dan menunjang proses kegiatan akademisi serta administratif. Untuk menjaga dan mengamankan data-data tersebut UNDIKSHA memerlukan sistem deteksi intrusi yang bisa mengetahui segala aktivitas yang mencurigakan dan tanpa mengganggu lalu lintas server. Penelitian ini menjelaskan tentang implementasi sistem IDS (*Intrusion Detection System*) dengan menggunakan metode deteksi *signature rules* dan memanfaatkan fitur *port mirroring*. Berdasarkan hasil yang diuji coba secara lokal pada server UNDIKSHA, sistem IDS dapat dengan baik mendeteksi serangan DOS (*Denial of Services*), SQL *injection, path traversal* dan XSS *stored*. Hasil evaluasi membuktikan bahwa IDS yang di implementasi pada server UNDIKSHA dapat mendeteksi dengan baik tanpa adanya alarm palsu, dengan kata lain IDS secara 100% bisa membedakan serangan yang benar terjadi dan aktivitas normal pengguna.
Kata Kunci*—Intrusion detection system, Data center, Port mirroring, Signature rules*

Abstract

A facility used to store, manage and disseminate information is the physical form of a data center. The role of the data center is also used to keep user data safe physically and digitally. Ganesha University of Education (UNDIKSHA) has a data center located in the UPA TIK building, which functions to serve and support the process of academic and administrative activities. To maintain and secure these data UNDIKSHA requires an intrusion detection system that can find out all suspicious activities and without disrupting server traffic. This research describes the implementation of an IDS (Intrusion Detection System) system using the signature rules detection method and utilizing the port mirroring feature. Based on the results tested locally on the UNDIKSHA server, the IDS system can properly detect DOS (Denial of Services), SQL injection, path traversal and XSS stored attacks. The evaluation results prove that the IDS implemented on the UNDIKSHA server can detect well without any false alarms, in other words, the IDS can 100% distinguish between true attacks and normal user activity.
Keywords*—Intrusion detection system, Data center, Port mirroring, Signature rules*

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**

1. PENDAHULUAN

Untuk memantau semua data yang melewati server, diperlukan usaha atau administrator yang terampil dalam memanfaatkan *software* monitoring server. Ketika admin tersebut sedang tidak mengawasi sistem, maka sistem bisa saja terjadi serangan dadakan tanpa diketehaui oleh admin [1]. Kejadian ini dapat merugikan pengguna dan menghambat pekerjaan suatu organisasi atau perusahaan, oleh karena itu diperlukan sistem yang dapat mendeteksi serta memberikan peringatan, mencatat semua aktivitas, dan dapat diakses dimana saja. IDS (*Intrusion Detection System*) merupakan sebuah sistem yang menggunakan perangkat lunak atau perangkat keras dan beroperasi secara otomatis untuk memantau kondisi dalam jaringan komputer, selain itu IDS dapat menganalisa masalah keamanan yang ada [2].

IDS memiliki kemampuan untuk mengamankan server tanpa mengganggu jalan lalu lintas tersebut. IDS harus ditempatkan pada router atau *switch* yang memiliki fitur *port mirroring*. Memanfaatkan fitur ini akan membuat IDS dapat memantau setiap paket data yang bergerak melalui server dan mendeteksi setiap aktivitas mencurigakan secara *real-time*. Keuntungan lain yang didapatkan dari memanfaatkan fitur ini yaitu memudahkan administrator memasang IDS tanpa harus mengubah topologi jaringan yang ada, karena IDS hanya perlu terhubung ke perangkat jaringan yang sama atau setiap paket data yang bergerak ke server harus dikirim juga ke IDS [3].

Proses rancangan IDS dengan teknik *port mirroring* ini sangat sesuai dengan kondisi yang diperlukan UPA TIK UNDIKSHA, karena UPA TIK memerlukan IDS untuk memantau segala aktivitas yang terjadi pada server dan harus dipasang tanpa mengganggu jalur dari server. IDS yang di instal adalah suricata dengan sistem operasi debian, untuk proses pemantauan *log* menggunakan SIEM (*Security Information and Event Management*) dari elasticsearch. SIEM adalah solusi yang menggabungkan manajemen informasi dan event keamanan untuk menyediakan analisis *real-time* terhadap informasi keamanan yang terjadi dalam sebuah organisasi. Elasticsearch dipilih karena memiliki integrasi dengan suricata dan mudah untuk proses instalasi dan konfigurasi [4].

Pengembangan sistem IDS dengan menggunakan teknik *port mirroring* bukanlah hal yang baru. Beberapa penelitian telah dilakukan berkaitan dengan implementasi sistem deteksi intrusi dengan metode *signature*. Penelitian yang dikerjakan oleh Isa et al [5] memberikan hasil pengujian dari IDS suricata dan *snort* untuk melihat bagaimana performa deteksi dari penggunaan sistem operasi yang berbeda. Hasil pengujian ini menggunakan metrik perhitungan seperti *true positive*, *false positive*, dan *accuracy*. Dari pengujian tersebut IDS *snort* lebih unggul dari suricata jika digunakan pada sistem operasi linux, sebaliknya suricata lebih unggul jika menggunakan sistem operasi *windows* untuk mendeteksi serangan.

Selanjutnya penelitian yang dilakukan oleh Tanang Anugraha et al [6] penelitian ini melakukan uji coba untuk mendeteksi serangan SQL *injection* menggunakan metode *signature* dari IDS suricata. Selain menggunakan metode deteksi, penelitian ini juga menggunakan metode *prevention* untuk mencegah serangan diteruskan server. Hasilnya suricata yang sudah dijalankan dengan model IDPS (*Intrusion Detection Prevention System*) dengan memanfaatkan nfqueue, serangan SQL *injection* berhasil terdeteksi dan diblokir oleh suricata. Pengujian ini dilakukan sebanyak 30 kali dan mendapatkan *response time* terendah yaitu 3,563 *miliseconds*.

Penelitian berikutnya yaitu dilakukan oleh Jeong dan Choi [7], studi ini memanfaatkan IDPS suricata dan *honeypot* dengan tipe *network* IDS yang bertujuan untuk mencegah serangan *zero-day*. Selain itu IDS ini juga memanfaatkan sistem antrian sementara untuk mendeteksi sekaligus memblokir serangan yang tidak diketahui secara *real-time*. Cara kerja sistem ini yaitu membuat paket serangan yang memang belum diketahui oleh IDS suricata masuk ke *honeypot*, kemudian dari *honeypot* dapat mengekstrak *signature rules* dan di implementasikan ke suricata.

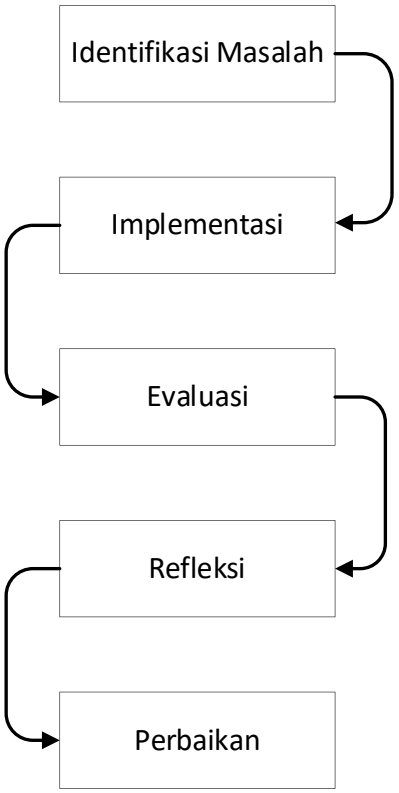
Terkait dengan kasus tersebut, peneliti mengembangkan sistem IDS suricata dengan menggunakan metode *signature rules* dan teknik *port mirroring*. Suricata sangat mudah untuk dikonfigurasi dan mendukung banyak integrasi dengan SIEM. Rancangan ini sangat sesuai dengan apa yang dibutuhkan UPA TIK UNDIKSHA, karena proses instalasinya yang tidak

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**

perlu mengganti atau mengubah topologi jaringan yang sudah terpasang. Selain itu sistem IDS ini dapat diakses dimana saja dengan memanfaatkan SIEM elasticsearch untuk tetap dapat memantau server.

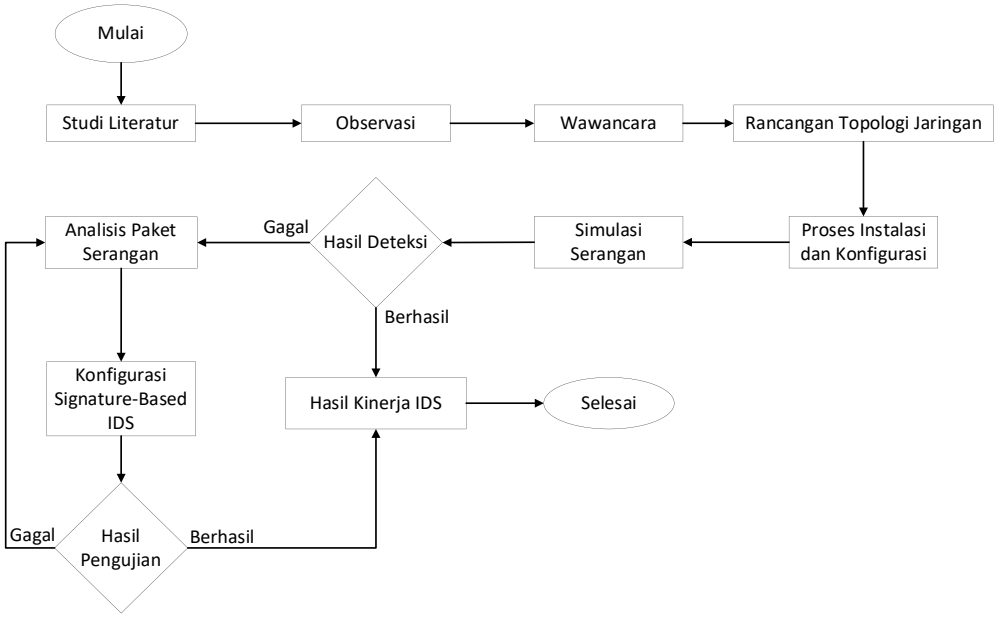
2. METODE PENELITIAN

Proses implementasi sistem IDS diterapkan dengan metode *action research*, yang dimulai dengan tahap pertama yaitu identifikasi masalah : studi literatur, observasi dan wawancara untuk mengumpulkan data yang dibutuhkan dalam proses perancangan sistem. Kemudian tahap implementasi : rancangan topologi jaringan dan proses instalasi serta konfigurasi merupakan langkah awal dalam mengimplementasikan sistem yang akan digunakan dalam server UPA TIK. Proses ini juga sudah di pertimbangkan dalam tahap pengumpulan data, agar sistem yang dijalankan sudah memenuhi persyaratan. Selanjutnya simulasi serangan, tahap ini bertujuan untuk mengetahui apakah IDS suricata bisa mendeteksi serangan-serangan yang sering terjadi pada *data center* UPA TIK, jika pengaturan *default* suricata berhasil mendeteksi serangan maka akan dilanjutkan ke tahap hasil kinerja. Lalu pada tahap evaluasi : analisis paket serangan, jika IDS suricata tidak dapat mendeteksi serangan maka akan dilanjutkan ke tahap ini untuk memonitoring aktivitas penyerang dan dibuatkan *rules* dengan metode *signature* untuk menanggulangi serangan tersebut. Selanjutnya adalah tahap refleksi : hasil kinerja IDS, hasil ini akan di uji dan dihitung kemampuan deteksinya [8]. Jika hasil perhitungan masih kurang baik maka perlu ada perbaikan, sampai hasil deteksi IDS lebih baik dan tidak menimbulkan *alarm* palsu. Terakhir adalah tahap perbaikan : ketika hasil *rules* yang diuji coba tidak sesuai dengan hasil yang ditentukan maka akan dilakukan perbaikan *rules* sampai hasilnya memenuhi kriteria (Gambar 1 dan Gambar 2).



Gambar 1. Metode Penelitian Action Research

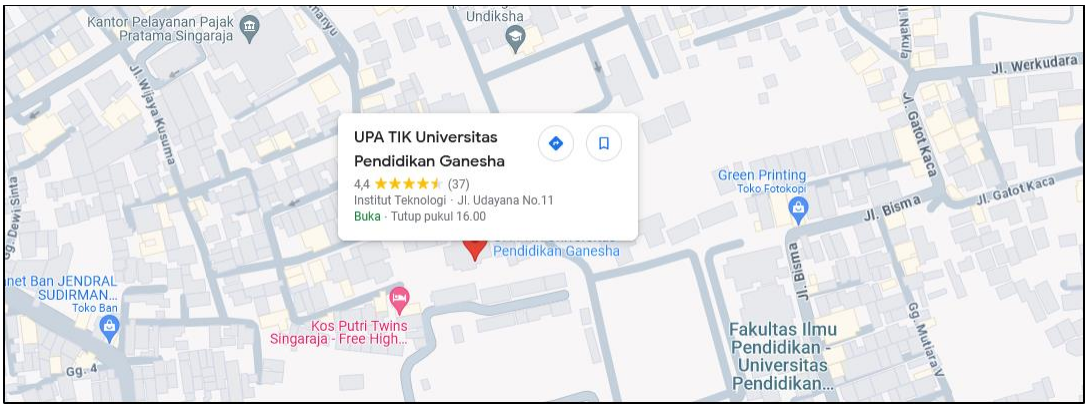
DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**



Gambar 2. Alur Penelitian Implementasi IDS

1.1. Lokasi Penelitian

Penelitian ini dilaksanakan di UPA TIK. Ruangan Divisi Jaringan, Universitas Pendidikan Ganesha, Buleleng Bali. Lokasi penelitian ditunjukkan pada gambar 3.



Gambar 3. Lokasi Penelitian Implementasi Sistem IDS

2.2 Perangkat Keras

Perangkat yang digunakan untuk uji coba dan implementasi sistem IDS dapat dilihat pada tabel 1.

Tabel 1. Daftar Perangkat Keras

Nama Perangkat Keras	Penjelasan dan Spesifikasi
Komputer IDS	Spesifikasi yang digunakan : a. Intel Core I7-3770 3.4 GHz b. RAM 6 GB c. HDD 1 TB Komputer ini digunakan untuk instalasi IDS suricata dan SIEM elasticsearch
Mikrotik CRS125-24G-1S-IN	Spesifikasi yang digunakan : a. AR9344 600 MHz

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**

Nama Perangkat Keras	Penjelasan dan Spesifikasi
	b. RAM 128 MB c. Storage 128 MB Mikrotik digunakan untuk menghubungkan server, IDS dan komputer penyerang serta memanfaatkan fitur *port mirroring*
Komputer Server	Spesifikasi yang digunakan : a. Intel Core I7-3770 3.4 GHz b. RAM 6 GB c. HDD 1 TB Komputer ini digunakan untuk instalasi server web sistem perencanaan dan penganggaran secara lokal
Komputer Penyerang	Spesifikasi yang digunakan : a. Intel Core I7-7700HQ 3.8 GHz b. RAM 32 GB c. HDD 1 TB Komputer penyerang akan menggunakan kali linux secara virtual dan berfungsi untuk menyerang server

2.3 Perangkat Lunak

Pada tabel 2 menunjukkan perangkat lunak yang digunakan untuk proses uji coba dan implementasi IDS suricata.

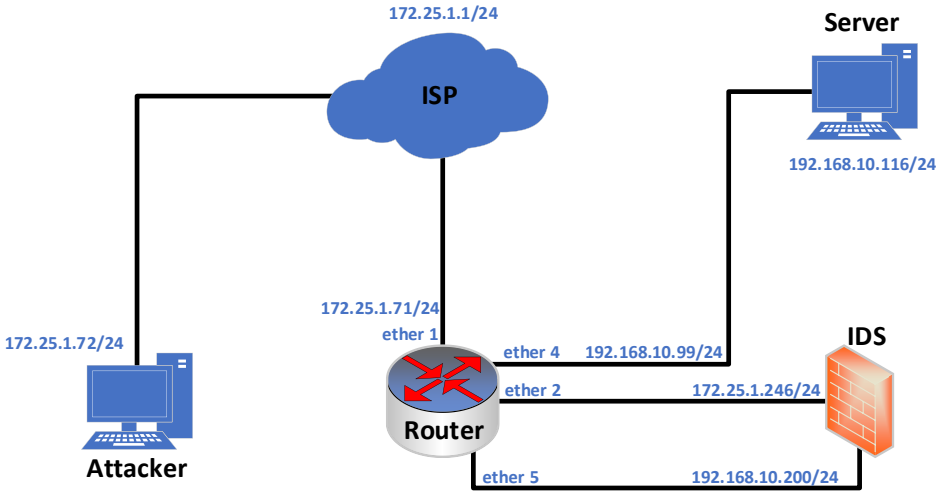
Tabel 2. Daftar Perangkat Lunak

Nama Perangkat Lunak	Penjelasan dan Spesifikasi
Wireshark	Wireshark digunakan untuk melakukan analisis serangan dan sebagai petunjuk untuk membuat *signature-rules*.
IDS Suricata	IDS suricata merupakan aplikasi yang digunakan untuk melindungi server dan mendeteksi segala aktivitas pada jaringan server.
Elasticsearch dan Kibana	Elasticsearch dan kibana adalah aplikasi untuk mengelola *log* suricata dan memiliki filter serta visualisasi yang memudahkan pengguna.
Kali Linux	Kali linux merupakan sistem operasi yang digunakan untuk penetrasi sistem seperti serangan DOS, *path traversal*, SQLi dan XSS.
DDOS-Ripper	DDOS-Ripper adalah aplikasi yang digunakan untuk menyerang server dengan mengirim paket HTTP ke target yang dituju.
ZAP Proxy	ZAP proxy merupakan aplikasi otomatis untuk mengeksploitasi kerentanan keamanan pada target yang dituju.
Burpsuite	Burpsuite digunakan untuk pengujian serangan XSS, karena burpsuite memiliki fitur *intruder* yang dapat membajiri server dengan serangan XSS.

2.4 Topologi Jaringan

Gambar 3 dan tabel 3 memberikan gambara terkait bagaimana topologi jaringan beserta alamat ip yang digunakan pada proses implementasi sistem IDS.

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**



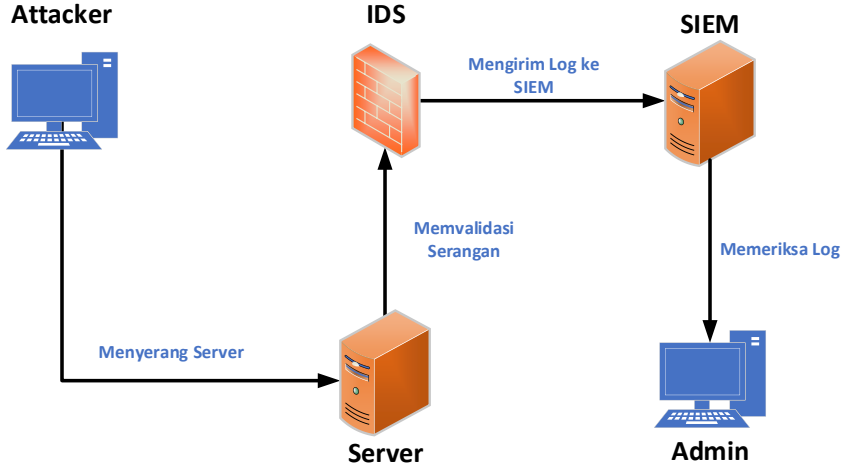
Gambar 4. Topologi Jaringan

Tabel 3. Alamat IP *Address* dan *Port*

Nama Perangkat Keras	Keterangan IP dan Port
ISP	ISP : 172.25.1.1/24
Router Mikrotik	ether1 : 172.25.1.71/24 (Bridge) ether4 : 192.168.10.99/24 (Bridge)
Komputer Penyerang	eth0 : 172.25.1.72/24
IDS	enp5s0 :172.25.1.246/24 enp3s0 : 192.168.10.200/24
Server	enp3s0 : 192.168.10.116/24

2.5 Alur Kerja Sistem

Cara kerja sistem IDS suricata dapat dilihat pada gambar 4, ketika komputer penyerang melakukan serangan ke server, IDS akan melakukan proses memvalidasi atau menyesuaikan serangan tersebut dengan *signature rules* yang ada. Setelah IDS berhasil memvalidasi bahwa adanya serangan, selanjutnya SIEM akan mencatat seluruh aktivitas yang ditemukan. Terakhir admin dapat mengecek semua kejadian yang terjadi dan melakukan *maintenance*.



Gambar 5. Cara Kerja Sistem IDS

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**

2.6 Evaluasi Kinerja IDS

Perhitungan keakuratan dan kinerja IDS dihitung dengan metrik dalam evaluasi sistem deteksi intrusi seperti *accuracy*, *precision*, *recall*, *specificity*, dan f1-*score* [9]. Perhitungan ini bertujuan untuk mengukur seberapa baik rule IDS suricata mendeteksi serangan dan apakah *rule* ini bisa membedakan serangan yang benar terjadi serta apakah bisa mengurangi terjadinya kesalahan deteksi atau *alarm* palsu. Berikut adalah penjelasan metrik yang digunakan untuk perhitungan evaluasi kinerja IDS :

1. *True Positive* (TP) : merupakan jumlah intrusi yang dideteksi dengan benar oleh sistem IDS.
2. *False Positive* (FP) : jumlah terjadinya kesalahan identifikasi sebagai serangan oleh sistem atau ketika adanya aktivitas serangan berbeda dan aktivitas normal alarm yang muncul tidak sesuai.
3. *True Negative* (TN): jumlah kejadian yang benar-benar bukan intrusi dan diidentifikasi dengan benar oleh sistem atau sistem berhasil dengan baik mengetahui bahwa benar tidak ada aktivitas mencurigakan.
4. *False Negative* (FN) : merupakan jumlah intrusi yang seharusnya terdeteksi tetapi tidak diketahui oleh sistem IDS.

Kemudian untuk masing-masing rumus evaluasi IDS dapat diukur dengan hasil pengujian sistem IDS dan dihitung dengan rumus berikut :

1. *Accuracy*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. *Precision*

$$Precision = \frac{TP}{TP + FP}$$

3. *Recall*

$$Recall = \frac{TP}{TP + FP}$$

4. *Specificity*

$$Specificity = \frac{TN}{TN + FP}$$

5. F1-*Score*

$$F1 - Score = \frac{2 \times (precision \times recall)}{precision + recall}$$

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data (Studi Literatur, Observasi, dan Wawancara)

Pada tahap awal peneliti akan mempelajari lebih dalam terkait penelitian yang sejenis dengan pembuatan dan rancangan sistem IDS. Proses ini dapat membuat pemahaman yang lebih baik terhadap konsep-konsep penelitian, teori, dan model yang relevan dengan topik penelitian. Kemudian tahap observasi dilakukan dengan mengunjungi langsung tempat UPA TIK Universitas Pendidikan Ganesha untuk mengamati keberadaan *data center* serta memahami komponen dan garis besar topologi jaringannya. Hal ini dapat melengkapi hal apa saja yang perlu dipertimbangkan dalam membuat sistem IDS beserta hal apa saja yang perlu dipersiapkan untuk merancang sistem. Observasi ini dapat menggambarkan penentuan hardware yang tepat untuk melakukan uji coba simulasi sistem IDS. Selanjutnya adalah tahap wawancara, tahapan

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**

ini dilakukan kepada pihak staf UPA TIK yang mengelola dan memelihara jaringan *data center* atau server utama. Peneliti akan berfokus pada rancangan sistem IDS dan menanyakan serangan apa saja yang paling rentan atau serangan yang sering terjadi pada server UPA TIK. Selain itu peneliti akan menyesuaikan kebutuhan yang diperlukan UPA TIK terkait implementasi sistem IDS yang akan diterapkan pada UPA TIK serta website apa yang diperbolehkan untuk diuji.

3.2 Proses Instalasi dan Konfigurasi

3.2.1 Mikrotik dan Suricata

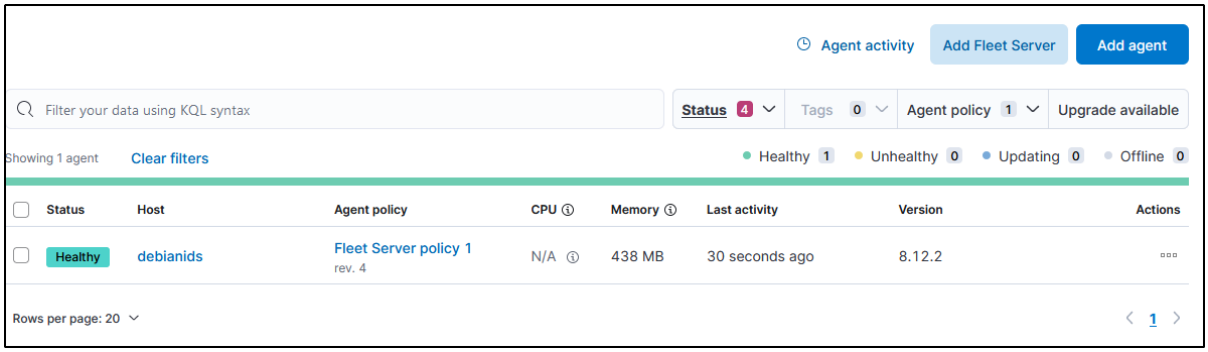
Proses ini diawali dengan melakukan konfigurasi pada mikrotik dengan mengaktifkan fitur *port mirroring* dan menghubungkan komputer server dengan komputer IDS. Kemudian setelah kedua komputer terhubung, proses selanjutnya adalah melakukan instalasi dan konfigurasi pada IDS suricata. Pada konfigurasi ini suricata akan diatur untuk mendeteksi lalu lintas jaringan yang terhubung ke server. Dibawah ini adalah tabel 4 yang menjelaskan proses konfigurasi yang dilakukan pada IDS suricata :

Tabel 4. Proses Konfigurasi Suricata

Proses	*Command*
Konfigurasi Suricata	- nano /etc/suricata/suricata.yml input : HOME_NET: “[192.168.10.0/24]” af-packet: interface: ether5 default-rule-path: /var/lib/suricata/rules rule-files: suricata.rules local.rules (Penambahan *rules* yang sesuai dengan simulasi serangan)
Aktifkan suricata setelah di reboot	systemctl enable suricata
Jalankan Suricata	systemctl start suricata

3.2.2 SIEM (*Security Information and Event Management*)

Selanjutnya adalah proses instalasi dan konfigurasi SIEM elasticsearch hasilnya dapat dilihat pada gambar 5. Proses ini membutuhkan nama domain yang sudah dikonfigurasi dengan sertifikat SSL/TLS. Kegunaanya agar ketika komputer IDS mati dan melakukan koneksi ulang untuk mengirim *log*, elasticsearch secara otomatis akan mengenali host tanpa harus konfigurasi ulang lagi. Pada proses implementasi ini nama domain yang digunakan adalah “myidscyberlocal46.site”. Kemudian komputer IDS dihubungkan menggunakan server *fleet* dan melakukan proses menambahkan integrasi dengan suricata (Gambar 6).



Gambar 6. Proses Instalasi SIEM dan *Agent* Elasticsearch

Gambar 7. Proses Integrasi Sistem SIEM dengan IDS

3.3 Simulasi Serangan

1. Serangan DOS Menggunakan DDoS *Ripper*
   Ketika melakukan percobaan serangan DOS pada halaman web sipepeng undiksha, IDS suricata hanya berhasil mendeteksinya dengan *rule anomaly* dari konfigurasi *default* suricata. Jumlah paket serangannya adalah 2081 dan terdeteksi secara akurat dengan jumlah yang sama.
2. Serangan *Path Traversal* dan SQL *Injection* Dengan ZAP *Scanning*
   Kemudian untuk serangan dari ZAP dengan menggunakan *scanning* secara otomatis, suricata tidak berhasil mendeteksi serangan *path traversal* dan SQL *injection* dari ZAP. Jumlah serangan yang dilakukan sebanyak 3 kali dan 15 kali.
3. Serangan XSS *Stored* Dengan Burpsuite
   Selanjutnya adalah serangan XSS *stored* dengan menggunakan "javascript", dari hasil percobaan IDS suricata juga tidak dapat mendeteksi serangan ini. Karena konten "javascript" tidak ada pada *rule* suricata. Jumlah serangan yang diuji adalah sebanyak 50 kode.

3.4 Analisis Paket Serangan dan Konfigurasi Signature Rules

1. Serangan DOS

```
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"DDoS Attacks Detected inbound"; flow:established,to_server; content:"GET / HTTP/1.1|0a|"; classtype:web-application-attack; sid:10000001; rev:4;)
```

2. Serangan *Path Traversal* dan SQL *Injection* Dengan ZAP *Scanning*

```
#Path Traversal

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"Possible Path Traversal Attack"; flow:established,to_server; http.method; content:"POST"; http.request_body; content:"..%2F"; fast_pattern; classtype:web-application-attack; sid:10000002; rev:3;)

#SQL Injection

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"Possible Path Traversal Attack"; flow:established,to_server; http.method; content:"POST"; http.request_body; content:"then+1+else+1+end"; fast_pattern; classtype:web-application-attack; sid:10000002; rev:3;)
```

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**

3. Serangan XSS *Stored*

```
alert http $EXTERNAL_NET any -> $HOME_NET any
(msg:"Possible XSS Attempt"; flow:established,to_server;
http.method; content:"POST"; http.request_body;
content:"javascript";

fast_pattern; classtype:web-application-attack;
sid:10000004; rev:2; )
```

3.5 Hasil Kinerja IDS

Dari hasil pengujian yang dilakukan sebanyak 10 kali pengulangan yang dapat dilihat pada tabel 4, serangan DOS yang biasanya menggunakan *option threshold* untuk mendeteksi serangan diubah menjadi deteksi konten saja. Hal ini terjadi jika menggunakan *option threshold* maka hasil *recall* akan menurun dibawah 50%, karena banyak serangan yang lolos. Jadi dengan memanfaatkan fokus deteksi ke konten tertentu *rule* serangan DOS ini membuat deteksi serangan menjadi lebih akurat dan sama sekali tidak terjadinya kesalahan deteksi. Kemudian untuk pengujian kedua dengan menggunakan konten atau *tool* yang berbeda yang dapat dilihat pada tabel 5, IDS suricata hanya berhasil mendeteksi serangan SQL *injection* dan XSS *stored* dengan kode "*script*". Pada percobaan serangan *path traversal* dengan konten "../" hasil *accuracy* merupakan hasil perhitungan saja, karena nilai *specificity* semuanya menggunakan nilai yang sama. *Specificity* didapatkan dari hasil pengujian dengan semua *rule* aktif jadi diambil nilai yang sama untuk semua serangan. Hal yang menentukan *rule* IDS suricata baik atau tidaknya adalah nilai akhir f1-*score*. Jika nilai dari f1-*score* adalah 1 maka *rule* tersebut sangat baik untuk mendeteksi serangan dan tidak adanya *alarm* palsu. Sebaliknya jika menurun atau dibawah nilai 1 maka IDS suricata sudah cukup baik atau masih kurang untuk mendeteksi serangan.

Tabel 5. Hasil Evaluasi Percobaan Pertama

Serangan	*Accuracy*	*Precision*	*Recall*	*Specificity*	F1-*Score*
DOS *Ripper*	100%	100%	100%	100%	1
Path Traversal ZAP *Scanning*	100%	100%	100%	100%	1
SQL *Injection* ZAP *Scanning*	100%	100%	100%	100%	1
XSS *Stored* javascript	100%	100%	100%	100%	1

Tabel 6. Hasil Evaluasi Percobaan Kedua

Serangan	*Accuracy*	*Precision*	*Recall*	*Specificity*	F1-*Score*
DOS dengan Slowlaris	3.4%	0%	0%	100%	0
Path Traversal konten "../"	93.8%	0%	0%	100%	0
SQL *Injection* dengan sqlmap	100%	100%	100%	100%	1
XSS *Stored* konten "*script*"	80%	100%	71%	100%	0.83

Setelah di uji coba, hanya serangan XSS *stored* dan SQL *injection* pada percobaan kedua yang bisa dideteksi oleh IDS suricata secara *default*. Karena percobaan kedua ini membuktikan bahwa IDS suricata perlu dibuatkan *rule* untuk setiap serangan baru atau dibuatkan *rule* yang bisa mendeteksi serangan lebih dari satu serangan. Berikut adalah *rule* perbaikan untuk mendeteksi serangan dengan *tool* atau konten serangan yang berbeda :

1. Serangan DOS Untuk Mendeteksi Lebih Dari 1 Serangan DOS

```
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"DDoS
Attacks Detected inbound"; flow:established,to_server;
threshold:type both, track by_src, count 50, seconds 10;
classtype:web-application-attack; sid:10000001; rev:4;)
```

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**

2. Serangan *Path Traversal* Untuk Mendeteksi Lebih Dari 1 Serangan *path*

```
alert http $EXTERNAL_NET any -> $HOME_NET any
(msg:"Possible Path Traversal Attack";
flow:established,to_server; pcre:"/(\.\.\/|\.\.%2F)/";
classtype:web-application-attack; sid:10000002; rev:3;)
```

Tabel 7. Hasil Evaluasi Percobaan Ketiga

Serangan	*Accuracy*	*Precision*	*Recall*	*Specificity*	F1-*Score*
DOS dengan Slowlaris	86.1%	100%	83.9%	100%	0.912
Path Traversal konten "../"	100%	100%	100%	100%	1
SQL *Injection* dengan sqlmap	100%	100%	100%	100%	1

4. KESIMPULAN

Berdasarkan hasil penelitian dan uji coba yang dilakukan, dapat disimpulkan bahwa IDS suricata dengan metode *signature rules* dapat mendeteksi serangan berbahaya seperti DOS, *path traversal*, SQL *injection* dan serangan XSS *stored* serta berfungsi dengan baik. Pada pengaturan *default* suricata tidak dapat mendeteksi serangan dari tool DOS *ripper*, ZAP, dan serangan XSS *stored* dengan konten "javascript" karena tidak ada rules atau penerapan deteksi yang identik untuk mengetahui aktivitas-aktivitas *web application attack*. Kemudian setelah menambahkan atau memperbaiki *rules* IDS suricata agar bisa mendeteksi lebih dari satu serangan, dengan hasil evaluasi yang didapatkan dari nilai f1-*score*, dapat dikatakan dengan metode *signature rules* sistem IDS suricata dapat dipilih untuk melindungi server. Dari hasil evaluasi kinerja IDS, suricata sudah mampu untuk mengidentifikasi serangan-serangan tersebut. Pada pengujian sistem, suricata juga tidak menggunakan *resouces* komputer terlalu tinggi. Selain itu suricata memiliki integrasi sistem yang bagus dalam melakukan sistem monitoring dan mudah dalam melakukan instalasi. Sistem IDS ini tidak hanya dapat melindungi server dari depan atau sebagai *firewall*, sistem IDS bisa ditempatkan dimana saja asalkan aliran data yang melewati server bisa dibaca oleh sistem IDS.

5. SARAN

Penelitian ini masih perlu dikembangkan lebih lanjut karena terdapat kekurangan pada penerapan rules IDS suricata. Metode yang digunakan hanya dapat mendeteksi serangan yang diketahui saja, jika ada serangan yang tidak diketahui seperti serangan DOS dengan *tool* slowlaris, *path traversal* dengan konten "../" dan serangan XSS *stored* dengan konten yang berbeda, maka IDS suricata tidak akan bisa mendeteksi serangan tersebut kecuali terdapat *rules anomaly* yang tepat untuk mendeteksi aktivitas yang mencurigakan. Selain itu *rule* untuk mengatasi serangan DOS masih belum cukup baik, karena IDS suricata masih melewatkan beberapa serangan yang harusnya dideteksi secara benar. Kemudian sistem IDS hanya bisa melakukan pemberitahuan saja pada admin lewat *log* IDS yang diberikan tanpa adanya sistem notifikasi yang membantu secara cepat untuk menanggulangi serangan tersebut. Selain itu sistem ini perlu dikembangkan agar bisa memblokir serangan, jadi pekerjaan administrator tidak terbebani karena sudah ditangani oleh sistem.

DOI: 10.33050/cices.v8i2 ISSN: **2356-5209**
pp. 1~10 *Online* ISSN: **2655-3058**

DAFTAR PUSTAKA

[1] W. A. Sulaksono and C. E. Suharyanto, "Implementasi Honeypot Sebagai Sistem Keamanan Jaringan Pada Virtual Private Server," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 90–95, 2020.

[2] E. Stephani, Fitri Nova, and Ervan Asri, "Implementasi dan Analisa Keamanan Jaringan IDS (Intrusion Detection System) Menggunakan Suricata Pada Web Server," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 1, no. 2, pp. 67–74, 2020, doi: 10.30630/jitsi.1.2.10.

[3] Y. N. Kholisho and Marfuatun, "EDUMATIC : Jurnal Pendidikan Informatika," *J. Pendidik. Inform.*, vol. 3, no. 2, pp. 99–108, 2019, doi: 10.29408/edumatic.v5i1.3278.

[4] G. K. Bada, W. K. Nabare, and D. K. K. Quansah, "Comparative Analysis of the Performance of Network Intrusion Detection Systems: Snort, Suricata and Bro Intrusion Detection Systems in Perspective," *Int. J. Comput. Appl.*, vol. 176, no. 40, pp. 39–44, 2020, doi: 10.5120/ijca2020920513.

[5] A. Thakkar and R. Lohiya, *A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions*, vol. 55, no. 1. Springer Netherlands, 2022.

[6] F. Tanang Anugrah, S. Ikhwan, and J. Gusti A.G, "Implementasi Intrusion Prevention System (IPS) Menggunakan Suricata Untuk Serangan SQL Injection," *Techné J. Ilm. Elektrotek.*, vol. 21, no. 2, pp. 199–210, 2022, doi: 10.31358/techne.v21i2.320.

[7] J. H. Jeong and S. G. Choi, "Hybrid System to Minimize Damage by Zero-Day Attack based on NIDPS and HoneyPot," *Int. Conf. ICT Converg.*, vol. 2020-Octob, pp. 1650–1652, 2020, doi: 10.1109/ICTC49870.2020.9289589.

[8] R. M. Davison, M. G. Martinsons, and J. Malaurent, "Research perspectives: Improving action research by integrating methods," *J. Assoc. Inf. Syst.*, vol. 22, no. 3, pp. 851–873, 2021, doi: 10.17705/1jais.00682.

[9] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614.