ANALISIS KINERJA *INTRUSION DETECTION SYSTEM* MENGGUNAKAN SNORT PADA *VIRTUAL PRIVATE SERVER*

OLEH

EKAGUSTIMAN NOHO

T3114210

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana

PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2021

PENGESAHAN SKRIPSI

ANALISIS KINERJA *INTRUSION DETECTION SYSTEM* MENGGUNAKAN SNORT PADA *VIRTUAL PRIVATE SERVER*

Oleh

EKAGUSTIMAN NOHO

T3114210

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana
Program Studi Teknik Informatika,
ini telah disetujui oleh Tim Pembimbing
Gorontalo, 27 Juni 2021

Pembimbing I	Pembimbing II
Zohrahayaw, M.Kom	**Serwin, M.Kom**
NIDN: 0912 17702	**NIDN: 0918078802**

PERSETUJUAN SKRIPSI

ANALISIS KINERJA *INTRUSION DETECTION SYSTEM* MENGGUNAKAN SNORT PADA *VIRTUAL PRIVATE SERVER*

Oleh

EKAGUSTIMAN NOHO

T3114210

Diperiksa oleh Panitia Ujian Strata Satu (S1)
Universitas Ichsan Gorontalo
Gorontalo, 8 Juli 2021

1.	Ketua Penguji	
	Yasin Aril Mustofa, M.Kom	…….
2.	Anggota	
	Sudirman Melangi, M.Kom	…….
3.	Anggota	
	Warid Yunus, M.Kom	…….
4.	Anggota	
	Zohrahayaty, M.Kom	…….
5.	Anggota	
	Serwin, M.Kom	…….

Mengetahui :

Dekan Fakultas Ilmu Komputer	Ketua Program Studi
Zohrahayaty, M.Kom	**Irvan Abraham Salihi, M.Kom**
NIDN: 0912117702	**NIDN: 0928028101**

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa :

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah di publikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 8 Juli 2021

Yang membuat pernyataan

Materai
10000

Ekagustiman Noho

ABSTRACT

EKAGUSTIMAN NOHO. T3114210. PERFORMANCE ANALYSIS OF INTRUSION DETECTION SYSTEM USING SNORT ON VIRTUAL PRIVATE SERVER

The internet has revolutionized communication. Its eases and uses come with various threats. Without online defenses, we leave ourselves open to being victims of fraud and even theft. Security and vigilance are very important to keep ourselves safe because everyone is a potential target, including governments and private companies. IDS is a piece of hardware or software that monitors a network or computer system for malicious activity or policy violations. The IDS used in this study is NIDS using Snort as its software. Snort is able to recognize attack patterns based on the rules created. Snort uses signature-based (characteristic) and anomaly-based (behavior) methods. Snort rules applied to the site http://ekagustiman have succeeded in detecting Low Level and Medium Level SQL Injection attacks. SQL requests with the GET method can be seen by the user in the URL browser and can also be seen in the Apache2 server's access.log. The Snort rules can be developed by experimenting with several other web attack techniques such as XSS, CSRF, and RFI. To improve the performance of Snort, it is necessary to store logs that use a database to be more structured in viewing logs on a regular basis.

Keywords: Snort, Intrusion Detection System, Virtual Private Server, Signature-Based, Anomaly Based

ABSTRAK

EKAGUSTIMAN NOHO. T3114210. ANALISIS KINERJA INTRUSION DETECTION SYSTEM MENGGUNAKAN SNORT PADA VIRTUAL PRIVATE SERVER

Internet telah merevolusi komunikasi, kemudahan dan penggunaannya datang dengan berbagai ancaman. Tanpa pertahanan online kita membiarkan diri terbuka untuk menjadi korban penipuan bahkan pencurian. Keamanan dan kewaspadaan sangatlah penting untuk menjaga diri kita aman karena setiap orang adalah target potensial, termasuk pemerintah dan perusahaan swasta.IDS adalah perangkat keras maupun perangkat lunak yang memonitoring jaringan atau sistem komputer untuk mengetahui aktifitas jahat atau pelanggaran kebijakan. Adapun IDS yang digunakan pada penelitian ini adalah NIDS menggunakan Snort sebagai perangkat lunaknya. Snort mampu mengenali pola serangan berdasarkan rule yang dibuat. Snort menggunakan metode signature based (ciri khas) dan anomaly based (tingkah laku).Snort rules yang diterapkan pada situs http://ekagustiman sudah berhasil mendeteksi serangan SQL Injection Low Level dan Medium Level. Request SQL dengan method GET dapat terlihat oleh user di url browser dan bisa dilihat pula pada access.log apache2 server. Snort rules [illegible]angkan dengan melakukan percobaan pada beberapa teknik se[illegible]ya seperti XSS, CSRF, dan RFI. Untuk meningkatkan pe[illegible]aka diperlukan penyimpanan log yang menggunakan database a[illegible]r dalam melihat log secara berkala.

Kaca Kunci: Snort, *Intrusion Detection System, Virtual Private Server, Signature Based, Anomaly Based*

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan skripsi ini dengan judul "ANALISIS KINERJA KINERJA INTRUSION DETECTION SYSTEM MENGGUNAKAN SNORT PADA VIRTUAL PRIVATE SERVER" untuk memenuhi salah satu syarat guna memperoleh gelar sarjana komputer pada Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa skripsi ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis, mengucapkan banyak terima kasih dan penghargaan yang setinggi – tingginya kepada:

1. Bapak Muhamad Ichsan Gaffar S.E M.AK, selaku Ketua Yayasan Pengembangan Ilmu Pengatahuan dan Teknologi (YPIPT) Ichsan Gorontalo,
2. Bapak Dr. Abd. Gaffar Latjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo,
3. Ibu Zohrahayaty, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo sekaligus Pembimbing Utama yang telah membimbing penulis selama penyusunan penelitian ini,
4. Bapak Sudirman S. Panna, M.Kom, selaku Wakil Dekan 1 Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo,
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo,
6. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan III Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo,
7. Bapak Irvan Abraham Salihi, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo,

8. Bapak Serwin, M.Kom, selaku Pembimbing Pendamping yang telah membimbing penulis selama penyusunan penelitian ini,
9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis,
10. Ibu dan Alm. Bapak saya tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis,
11. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis,
12. Kepada semua pihak yang ikut membantu dalam penyelesaian skripsi ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah, SWT melimpahkan balasan atas jasa-jasa mereka kepada kamu. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, 27 Juni 2021

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Internet merupakan istilah yang sudah tidak asing lagi bagi hampir semua orang yang pernah mendengar atau menggunakannya, mulai dari anak-anak hingga orang dewasa di zaman dimana hampir semua aktivitas serga digital.

Perkembangan Internet (*Interconnection-networking*) berupa jaringan komputer yang terhubung menggunakan standar *Internet Protocol Suite* (TCP/IP) yang melayani jutaan bahkan milyaran pengguna di seluruh dunia sebagai media bertukar informasi tanpa batasan ruang dan waktu. Internet telah merevolusi komunikasi, kemudahan dan penggunaannya datang dengan berbagai ancaman. Tanpa pertahanan online kita membiarkan diri terbuka untuk menjadi korban penipuan bahkan pencurian.

Ancaman di Internet atau sering disebut *Cyber Crime* merupakan tindakan yang dilakukan oleh seseorang atau kelompok yang dapat merusak, mencuri, bahkan memanipulasi data untuk diambil keuntungannya oleh mereka. Keamanan dan kewaspadaan sangatlah penting untuk menjaga diri kita aman karena setiap orang adalah target potensial, termasuk pemerintah dan perusahaan swasta.

Banyak pengguna internet tidak peduli akan keselamatan mereka. Contoh mengunjungi berbagai situs tanpa membaca *Term and Condition* terlebih dahulu atau mengijinkan situs tersebut untuk mengakses lokasi kita pada saat ini. Pada akhirnya yang dirugikan adalah pengguna internet yang menyepelekan hal-hal diatas tadi. Begitu pula dengan sistem administrator yang tidak berhati-hati dalam mengamankan sistem atau jaringan sepertikasusyang terjadi pada BPJS Kesehatan kemarin yang mengalami kebocoran hingga 279 juta data penduduk Indonesia[1].

Dengan berkembangnya teknologi keamanan jaringan ternyata belum bisa menjamin sepenuhnya untuk melindungi sistem dari serangan peretas. Berbagai cara akan dilakukan untuk dapat mengganggu sistem, seperti dengan cara membanjiri trafik pada jaringan sehingga dapat menghabiskan *resource* pada server dan mengakibatkan server tidak bisa menjalankan fungsinya dengan baik.

Beberapa keamanan jaringan default yang sudah disediakan oleh berbagai

vendor, contoh Windows *Firewall* oleh Microsoft Windows, dan IPTables oleh Linux. *Firewall* merupakan program perangkat lunak yang berfungsi untuk mencegah akses ilegal dari dalam maupun dari luar jaringan komputer. *Firewall* merupakan fokus dari segala kebijaksanaan sekuritas, karena *Firewall*adalahtempat keluar masuknya internet atau akses dari luar dalam jaringan komputer. Tidak hanya itu, *Firewall* dapat mencatat segala aktivitas yang berkaitan dengan alur data secara efisien. Namun dibalik kelebihannya yang sudah disebutkan, *Firewall* tidak bisa mendeteksi jenis serangan terbaru yang belum dikenal. Maka sangat diperlukan sistem keamanan yang lebih efektif dari *Firewall*, IDS menjadi alternatif berikutnya. Seperti yang dikatakan oleh Saiyan Saiyod dari Khon Kaen University pada penelitiannya, "Saat ini, sistem IDS memainkan peran penting dalam menjaga keamanan jaringan. IDS mendeteksi dan memantau data lalu lintas jaringan pada sistem jaringan dan memperingatkan pengguna ketika serangan jahat terjadi[2]."

IDS adalah perangkat keras maupun perangkat lunak yang memonitoring jaringan atau sistem komputer untuk mengetahui aktifitas jahat atau pelanggaran kebijakan. Setiap kegiatan jahat atau pelanggaran akan dilaporkan kepada administrator atau dikumpulkan secara terpusat menggunakan SIEM (*Security Information and Event Management*). Sistem SIEM menggabungkan hasil data yang dikumpulkan dari berbagai sumber dan melakukan penyaringan untuk tipe pengingatan yang berbeda yang akan dikirimkan kembali ke sistem administrator.

IDS menurut klasifikasinya terbagi menjadi 2, yakni HIDS (*Host-based Intrusion Detection System*) dan NIDS (*Network-based Intrusion Detection System*). Perbedaan mendasar antara klasifikasi IDS terdapat pada ruang lingkup komputer atau jaringan yang terpasang IDS. HIDS berfungsi memonitor dan menganalisa trafik jaringan yang dari berasal dan keluar dari host dimana IDS tersebut diimplementasikan. Sedangkan NDIS memonitor dan menganalisa seluruh subnet jaringan dimana IDS ini akan menangkap semua paket seperti melakukan sniffing. Beberapa contoh perangkat keras IDS yaitu Huawei NIP 2000/5000 dan contoh lain perangkat lunak IDS yakni Snort dan AIDE (*Advanced Intrusion Detection Environment*).

Adapun IDS yang digunakan pada penelitian ini adalah NIDS menggunakan Snort sebagai perangkat lunaknya. Snort mampu mengenali pola serangan berdasarkan rule yang dibuat. Snort menggunakan metode *signature based* (ciri khas) dan *anomaly based* (tingkah laku). Untuk mendeteksi setiap gejala serangan tersebut, sistem menggunakan pengenalan terhadap sumber yang didapat dari pihak yang dianggap sebagai ancaman dalam sistem jaringan komputer. Snort sendiri menjadi pilihan peneliti karena beberapa alasan berikut : berukuran kecil mudah dikonfigurasi, dan mendukung banyak sistem operasi dan yang paling utama snort bersifat *Open Source*yang dirilis dibawah lisensi di mana pemegang hak cipta memberikan hak kepada pengguna untuk menggunakan, mempelajari, mengubah, dan mendistribusikan perangkat lunak dan kode sumbernya kepada siapa pun dan untuk tujuan apa pun[3].

Penelitian ini dilakukan pada VPS layanan Alibaba Cloud yang sudah terinstall Ubuntu Server. Pada penelitian ini penulis akan melakukan analisa terhadap trafik jaringan yang legal maupun ilegal kemudian melakukan implementasi sistem keamanan jaringan menggunakan Snort sebagai IDS yang di install pada *Ubuntu Server*, kemudian melakukan pengujian analisis kinerja untuk mengetahui kemampuan *IDS*.

Menurut penelitian yang dilakukan oleh Ervin Kusuma Dewi dan Patmi Kasih, 2017. Analisis *Log* Snort Menggunakan *Network Forensic*. Berdasarkan implementasi dengan menggunakan *tools* keamanan jaringan Snort maka dapat disimpulkan Snort yang dibangun dapat memantau lalu lintas paket di dalam jaringan serta mampu mendeteksi serangan berdasarkan rule yang di set, sehingga serangan jaringan kopmuter tersebut dapat segera ditangani sesegara mungkin oleh administrator karena terdapat *alert*[4].

Berdasarkan uraian permasalahan diatas, maka peneliti tertarik untuk melakukan penelitian dengan judul **"Analisis Kinerja *Intrusion Detection System* menggunakan SNORT pada *Virtual Private Server*."**

1.2 Identifikasi Masalah

Berdasarkan uraian diatas, maka dapat diidentifikasi permasalahannya sebagai berikut :

1. Tidak ada sistem deteksi serangan otomatis pada *Virtual Private Server*dalam mengamankan *Ubuntu Server*.
2. Adanya kesulitan dalam mengetahui apakah sistem sedang diserang atau tidak.

1.3 Rumusan Masalah

Dari latar belakang di atas, maka masalah yang akan dibahas dapat dirumuskan sebagai berikut :

1. Bagaimana implementasi *IDS* pada *VPS* sehingga dapat melakukan monitoring serangan pada jaringan komputer?
2. Bagaimana mencegah terjadinya penyusupan atau penyerangan pada jaringan komputer?

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut:

1. Untuk mengetahui implementasi *IDS* pada *VPS* sehingga dapat melakukan monitoring serangan pada jaringan komputer.
2. Untuk membantu sistem *administrator*dalam mencegah terjadinya penyusupan atau penyerangan pada jaringan komputer.

1.5 Manfaat Penelitian

Beberapa manfaat penelitian:

1. Manfaat Teoritis

Secara teoritis diharapkan dapat memberikan wawasan tentang konsep dan cara kerja sistem pengamanan jaringan, mengetahui lebih rinci tentang istilah IDS, *IPTables*, *Firewall*, *NAT*, dan *Linux Server*.

2. Manfaat Praktis

Secara praktis diharapkan melalui analisis penelitian ini bisa menyadarkan pengguna *Internet* atau sistem *administrator* jaringan lebih berhati-hati dalam melakukan komunikasi *via Internet* dalam hal ini bermain di *Social Media*, transaksi perbankan, atau pun mengakses dan membeli dari toko online.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Berikut adalah penelitian terdahulu terkait dengan keamanan sistem jaringan komputer, yaitu :

Tabel 2.1: Tinjauan Studi

No	PENELITI	JUDUL	HASIL
1.	Sahid Aris Budiman, Catur Iswahyudi, Muhammad Sholeh (2014)	Implementasi *Intrusion Detection System* (IDS) Menggunakan Jejaring Sosial Sebagai Media Notifikasi	1. Hasil pengujian menunjukkan bahwa setiap ada serangan yang datang dari luar menuju *host* atau *server* yang didalamnya terdapat IDS yang sedang berjalan, maka secara otomatis akan mendeteksi dan memberitahukan kepada *administrator* berupa notifikasi yang dikirim melalui jejaring sosial *Facebook*, *Twitter*, dan *Whatsapp* dengan rentang waktu yang relatif cepat, sehingga *administrator* dapat melakukan tindak lanjut terhadap jenis serangan yang dilakukan oleh *intruder*[5].
2.	Moh. Dahlan, S.T., M.T, Anastasya Latubessy, S.Kom., M.Cs, Lelly Hidayah	Pengujian dan Analisa Keamanan Website Terhadap Serangan *SQLInjection*	Menyimpulkan bahwa : 1. Tidak ada sistem yang dikatakan benar-benar aman, sehingga aktivitas jaringan perlu dipantau setiap saat dengan mengamati setiap paket

No	PENELITI	JUDUL	HASIL
	Anggraini, S.Kom., M.Cs, Mukhamad Nurkamid, S.Kom., M.Cs (2014)		data yang berjalan didalam jaringan. 2. Dengan adanya IDS Snort, aktifitas jaringan berjalan di Pusat Sistem Informasi (PSI) Universitas Muria Kudus dapat dipantau secara berkala dan dianalisis sebagai upaya deteksi dini terhadap serangan. 3. Pemberian aturan Snort (*rule)* yang tepat dapat memberikan peringatan/*alert* sehingga serangan dari *intruder* terhadap jaringan dapat diketahui oleh *sysadmin*[6].
3.	Ervin Kusuma Dewi, Patmi Kasih (2017)	Analisis *Log* Snort Menggunakan *Network Forensic*	Berdasarkan implementasi dengan menggunakan *tools* keamanan jaringan Snort maka dapat disimpulkan Snort yang dibangun dapat memantau lalu lintas paket di dalam jaringan serta mampu mendeteksi serangan berdasarkan *rule* yang di *set*, sehingga serangan jaringan komputer tersebut dapat segera ditangani sesegera mungkin oleh

No	PENELITI	JUDUL	HASIL
			administrator karena terdapat *alert*[4].

2.2 Tinjauan Pustaka

2.2.1 Jaringan Komputer

Jaringan komputer adalah jaringan telekomunikasi yang memungkinkan antar komputer saling berkomunikasi dengan bertukar data. Tujuan dari jaringan komputer adalah untuk mencapai tujuannya, setiap bagian dari jaringan komputer dapat meminta dan memberikan layanan. Pihak yang meminta/menerima layanan disebut *client* dan pihak yang menyediakan/mengirim layanan disebut *server*. Desain ini disebut sistem c*lient-server* dan digunakan di hampir semua aplikasi jaringan komputer[7].

Kemudian dua buah komputer yang masing-masing memiliki kartu jaringan dihubungkan dengan kabel atau tanpa kabel sebagai media transmisi data, dan terdapat perangkat lunak sistem operasi jaringan yang akan membentuk jaringan komputer sederhana. Jika ingin membuat jaringan komputer yang jangkauannya lebih luas, diperlukan peralatan tambahan seperti *Hub, Bridge, Switch, Router, Gateway* sebagai peralatan interkoneksi[7].

Jaringan komputer berdasarkan skalabilitasnya terbagi menjadi :

1. ***PERSONAL AREA NETWORK* (PAN)**

Merupakan jaringan komunikasi antara satu perangkat dengan perangkat lainnya dengan jarak yang sangat dekat, yaitu hanya beberapa meter saja. *Personal Area Network* atau sering disebut PAN, menggunakan teknologi WAP (*Wireless Application Protocol*) dan Bluetooth. PAN terhubung melalui bus yang ada pada komputer, seperti Firewire dan USB[8].

2. ***LOCAL AREA NETWORK* (LAN)**

LAN adalah jaringan komputer yang jaringannya hanya mencakup area kecil, seperti jaringan komputer di rumah, sekolah, kampus, dll. *Local Area Network* atau LAN tidak memerlukan jaringan telekomunikasi yang disewa dari

operator telekomunikasi dan tidak menggunakan fasilitas komunikasi publik seperti telepon, tetapi pemilik LANlah yang mencoba mengelola media komunikasi tersebut[8].

Untuk LAN umumnya berbasis teknologi *IEEE 802.3 Ethernet* menggunakan perangkat *switch*. Kemudian ada juga teknologi 802.11b (*Wi-Fi*) yang sering digunakan untuk membentuk LAN. Tempat yang menyediakan koneksi LAN dengan teknologi *Wi-Fi* disebut *Hotspot*[8].

3. ***METROPOLITAN AREA NETWORK* (MAN)**

Merupakan jaringan komunikasi di kota dengan transfer data berkecepatan tinggi.menghubungkan beberapa lokasi seperti kantor, kampus,pemerintah, dll. MAN mencakup area antara 5 dan 50 kilometer[8].

Umumnya, MAN tidak dimiliki oleh suatu organisasi, tetapi oleh salah satupenyedia layanan jaringan yang menjual layanan kepada pengguna. Alih-alih LAN,

MAN lebih besar dan memiliki jangkauan yang lebih luas, tetapi untuk teknologi yangpenggunaannya sama dengan LAN[8].

4. ***WIDE AREA NETWORK* (WAN)**

WAN adalah jaringan komputer yang mencakup area yang luas, yaitu antar komputer daerah, kota bahkan negara. *Internet* adalah contoh jaringan WAN. Saat ini, hampir setiap komputer yang kita temukan adalah bagian dari jaringankomputer yang kompleks. Misalnya, ketika laptop atau komputer yang *standalone* (mandiri) kemudian kita tambahkan *modem* USB agar terhubung ke *Internet*, maka komputer kita terhubung dan merupakan bagian dari Jaringan yang sangat besar yaitu WAN (*Wide Area Network*)[8].

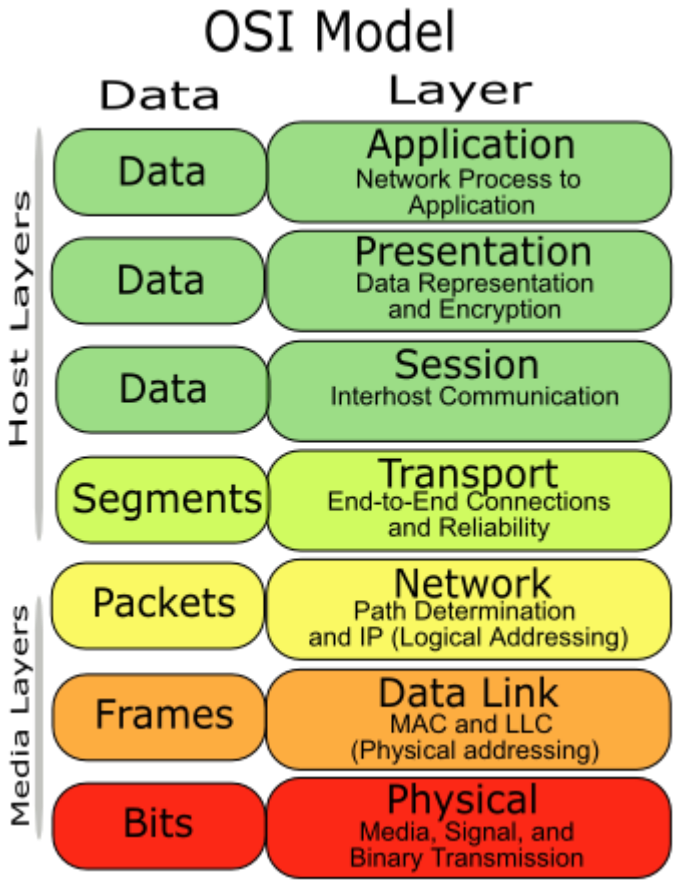
2.2.2 *TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL* (TCP/IP)

Banyak digunakan untuk jaringan Internet, dikembangkan secara terpisah dari protokol model lapisan OSI (Open System Interconnection). Akibatnya, struktur lapisan sangat tidak kompatibel dengan protokol model OSI standar. Seperti yang telah dijelaskan di atas, protokol model OSI terdiri dari 7 (tujuh) layer, yaitu physical, data link, network, transport, session, presentation dan application.

Sedangkan protokol TCP/IP hanya terdiri dari 5 (lima) layer yaitu physical, data link, network, transport dan application. Empat lapisan pertama (bawah) TCP / IP mewakili empat lapisan terendah dari model OSI: fisik, tautan data, jaringan, dan transportasi. Sedangkan lapisan atas TCP/IP (aplikasi) mewakili tiga lapisan atas model OSI, yaitu session, presentation dan application[9].

2.2.3 *OSI LAYER*

Lapisan OSI adalah model referensidigunakan untuk memahami jaringan komputer secara umum. Secara *de facto*, lapisan OSI telah digunakan sebagai acuan saat mempelajari jaringan yang dibangun dengan perangkat *Cisco*. Model referensi OSI atau model referensi OSI terdiri dari 7 lapisan (*layers*)[10].



Gambar 2.1: OSI Model [11]

Layer 7 : *Application Layer*

Application Layer adalah lapisan atas model OSI dan bertanggung jawab untuk menyediakan antarmuka antara protokol jaringan dan aplikasi di komputer. Lapisan aplikasi menyediakan layanan yang dibutuhkan oleh aplikasi, seperti menyediakan antarmuka ke *Simple Mail Transfer Protocol* (SMTP), *telnet*, dan *File Transfer Protocol* (FTP). Di sinilah aplikasi saling berhubungan dengan jaringan[10].

Layer 6 : *Presentation Layer*

Presentation Layer bertanggung jawab untuk mendefinisikan sintaks yang digunakan *host* di jaringan untuk berkomunikasi. Lapisan presentasi juga melakukan proses enkripsi/dekripsi informasi atau data sehingga dapat digunakan pada lapisan aplikasi[10].

Layer 5 : *Session Layer*

Session Layer bertanggung jawab untuk mengontrol dialog koneksi sesi, seperti membangun, mengelola, dan mengakhiri koneksi antar komputer. Untuk membangun sesi komunikasi, lapisan sesi menggunakan sirkuit virtual yang dibuat oleh *Transport Layer*[10].

Layer 4 : *Transport Layer*

Transport Layer bertanggung jawab untuk mengirim pesan antara dua atau lebih host di jaringan. *Transport Layer* juga menangani pemisahan dan penggabungan pesan dan juga mengontrol keandalan jalur koneksi yang diberikan. Protokol TCP adalah contoh *Transport Layer* yang paling banyak digunakan[10].

Layer 3 : *Network Layer*

Network Layer bertanggung jawab untuk menentukan jalur yang akan digunakan untuk mentransfer data antar perangkat di jaringan. *Router* jaringan beroperasi pada lapisan ini, yang juga merupakan fungsi utama dari lapisan jaringan dalam hal perutean. Routing memungkinkan paket untuk berpindah antar komputer yang terhubung satu sama lain. Untuk mendukung proses perutean ini, lapisan jaringan menyimpan alamat logis sebagai alamat IP untuk setiap perangkat di jaringan. Lapisan jaringan juga mengelola pemetaan antara alamat logis dan alamat fisik. Pada jaringan IP, pemetaan ini dilakukan dengan menggunakan *Address ResolutionProtocol* (ARP)[10].

Layer 2 : *Data-link Layer*

Data-link Layer bertanggung jawab untuk memeriksa kesalahan yang mungkin terjadi selama proses transmisi data dan juga untuk membungkus *bit*dalam bingkai data. *Data-link Layer* juga mengelola skema pengalamatan fisik, seperti alamat MAC pada jaringan. Lapisan data link merupakan salah satu lapisan OSI yang cukup kompleks, oleh karena itu lapisan ini dibagi menjadi dua sub lapisan yaitu lapisan *Media Access Control* (MAC) dan lapisan *Logical Link Layer*(LLC). Lapisan *Media Access Control* (MAC) bertanggung jawab untuk mengontrol bagaimana perangkat di jaringan memperoleh akses ke media dan izin untuk mengirimkan data. Layer *Logical Link Control* (LLC) bertanggung jawab untuk mengidentifikasi dan mengenkapsulasi protokol lapisan jaringan dan mengendalikan pengecekan kesalahan dan juga menyinkronkan *frame*[10].

Layer 1 : *Physical Layer*

Physical Layer adalah lapisan pertama atau terendah dari model OSI. Lapisan ini bertanggung jawab untuk mentransmisikan bit data digital dari lapisan fisik perangkat pengirim (sumber) ke lapisan fisik perangkat penerima (tujuan) melalui media komunikasi jaringan. Pada lapisan fisik, data ditransmisikan menggunakan jenis sinyal yang kompatibel dengan media fisik, seperti tegangan listrik, kabel, frekuensi radio, atau inframerah atau cahaya biasa[10].

2.2.4 Keamanan Jaringan

Sistem keamanan jaringan komputer merupakan mesin yang digunakan dalam melakukan identifikasi dan melakukan pencegahan dari penggunaan yang tidak sesuai atau tidak sah pada jaringan komputer. Melalui sistem jaringan inilah dapat membantu untuk melakukan pencegahan dengan cara menghentikan pengguna yang tidak sesuai atau seringkali disebut sebagai penyusup [12].

Secara umum keamanan komputer terdiri dari lima aspek:

1. ***Privacy***

Privacy, adalah sesuatu yang rahasia. Intinya adalah untuk mencegah orang yang tidak berwenang mengakses informasi tersebut. Contohnya adalah *email* atau file lain yang tidak boleh dibaca orang lain, bahkan *administrator*. Pencegahan yang

dapat dilakukan adalah dengan menggunakan teknologi enkripsi, sehingga hanya pemilik informasi yang dapat mengetahui informasi yang sebenarnya[13].

2. ***Confidentiality***

Confidentiality, adalah data yang diberikan kepada pihak lain untuk tujuan khusus tetapi tetap didistribusikan. Misalnya, data pribadi seperti: nama, alamat, nomor identitas, telepon, dll. *Confidentiality* akan terlihat ketika Anda diminta untuk membuktikan kejahatan seseorang, apakah pemilik informasi memberikan informasi kepada orang yang memintanya atau merawat kliennya[13].

3. ***Integrity***

Integrity, penekanannya adalah bahwa informasi tersebut tidak boleh diubah kecuali oleh pemilik informasi tersebut. Terkadang data yang telah dienkripsi tidak terjaga keutuhannya karena ada kemungkinan *ciphertext* dapat berubah. Contoh: Integritas menyerang ketika email yang dikirim di tengah jalan disadap dan isinya berubah, sehingga email yang sampai ke tujuannya berubah[13].

4. ***Authentication***

Authentication, dilakukan ketika pengguna login dengan username dan password, apakah cocok atau tidak, jika cocok akan diterima dan tidak ditolak. Hal ini umumnya berkaitan dengan hak akses seseorang, apakah itu akses yang sah atau tidak[13].

5. ***Availability***

Availability, aspek ini berkaitan dengan apakah data tersedia pada saat dibutuhkan atau dibutuhkan. Jika suatu data atau informasi terlalu ketat, keamanan akan menyulitkan untuk mengakses data tersebut. Selain itu, akses yang lambat juga menyulitkan pemenuhan aspek *availability*. Serangan yang biasanya dilakukan dalam hal ini adalah *denial of service* (DoS), yaitu kegagalan layanan ketika ada permintaan data sehingga komputer tidak dapat melayaninya. Contoh lain dari penolakan layanan ini adalah mengirimkan permintaan yang berlebihan sehingga komputer tidak dapat lagi mendukung beban dan akhirnya komputer mati[13].

2.2.5 Jenis-jenis serangan pada jaringan komputer

Berikut adalah jenis-jenis serangan pada jaringan komputer yang biasa dilakukan oleh attacker:

1. **Virus**

Virus tidak dapat dijalankan sendiri; itu membutuhkan interaksi pengguna untuk menginfeksi komputer dan menyebar di jaringan. Contohnya adalah email dengan tautan berbahaya atau lampiran berbahaya. Ketika penerima membuka lampiran atau mengklik tautan, kode berbahaya akan diaktifkan dan mengelak dari kontrol keamanan sistem dan membuatnya tidak dapat dioperasikan. Dalam kasus ini, pengguna secara tidak sengaja merusak perangkat[14].

2. ***Malware***

Serangan malware adalah salah satu serangan cyber paling parah yang dirancang khusus untuk menghancurkan atau mendapatkan akses tidak sah melalui sistem komputer yang ditargetkan. Sebagian besar malware menggandakan diri, yaitu, ketika menginfeksi sistem tertentu, ia masuk melalui internet dan dari sana, menginfeksi semua sistem yang terhubung ke internet di jaringan. Perangkat titik akhir eksternal jika terhubung, juga akan terinfeksi. Ia bekerja lebih cepat dari jenis konten berbahaya lainnya[14].

3. ***Worm***

Worm dapat memasuki perangkat tanpa bantuan pengguna. Saat pengguna menjalankan aplikasi jaringan yang rentan, penyerang di koneksi *internet* yang sama dapat mengirim *malware* ke aplikasi itu. Aplikasi dapat menerima *malware* dari *internet* dan menjalankannya, sehingga menciptakan *worm*[14].

4. ***Phishing***

Phishing adalah jenis serangan jaringan yang paling umum. Ini adalah singkatan dari mengirim email yang mengaku dari sumber atau bankir yang dikenal dan menciptakan rasa urgensi untuk menarik pengguna untuk menindaklanjutinya. *Email* tersebut mungkin berisi tautan atau lampiran berbahaya atau mungkin meminta untuk membagikan informasi rahasia[14].

5. ***Botnet***

Ini adalah jaringan komputer pribadi yang menjadi korban perangkat lunak berbahaya. Penyerang mengontrol semua komputer di jaringan tanpa sepengetahuan pemiliknya. Setiap komputer di jaringan dianggap sebagai zombie

karena mereka melayani tujuan menyebarkan dan menginfeksi sejumlah besar perangkat atau seperti yang dipandu oleh penyerang[14].

6. **DoS (*Denial of Service*)**

Denial of Service adalah serangan krusial yang menghancurkan sebagian atau seluruh jaringan korban atau seluruh infrastruktur TI untuk membuatnya tidak tersedia bagi pengguna yang sah[14].

Serangan DoS dapat dikategorikan dalam tiga bagian berikut -

1. ***Connection Flooding***

Penyerang menghambat *host* dengan membuat koneksi TCP dalam jumlah besar pada *host* yang ditargetkan. Koneksi palsu ini memblokir jaringan dan membuatnya tidak tersedia untuk pengguna yang sah[14].

2. ***Vulnerability Attack***

Dengan mengirimkan beberapa pesan yang dibuat dengan baik ke sistem operasi atau aplikasi yang rentan yang berjalan pada host yang ditargetkan, menghentikan layanan atau memperburuk keadaan sampai host tersebut mogok[14].

3. ***Bandwidth Flooding***

Penyerang mencegah paket yang sah mencapai *server* dengan mengirimkan paket yang membanjiri. Paket yang dikirim berjumlah besar sehingga tautan target diblokir untuk diakses orang lain[14].

7. ***Distributed Denial of Service*(DDoS)**

Ini adalah versi serangan *DoS* yang kompleks dan jauh lebih sulit untuk dideteksi dan dipertahankan dibandingkan dengan serangan *DoS*. Dalam serangan ini, penyerang menggunakan beberapa sistem yang dikompromikan untuk menargetkan satu sistem target serangan *DoS*. Serangan *DDoS* juga memanfaatkan *botnet*[14].

8. ***Man-In-The-Middle* (MITM)**

Serangan *man-in-the-middle* adalah seseorang yang berdiri di antara percakapan yang terjadi antara Anda dan orang lain. Dengan berada di tengah, penyerang menangkap, memantau, dan mengontrol komunikasi Anda secara

efektif. Misalnya, ketika lapisan bawah jaringan mengirimkan informasi, komputer di lapisan tersebut mungkin tidak dapat menentukan penerima yang bertukar informasi[14].

9. ***Packet Sniffer***

Ketika penerima pasif ditempatkan di wilayah pemancar nirkabel, ia mencatat salinan dari setiap paket yang dikirimkan. Paket-paket ini dapat berisi informasi rahasia, data sensitif dan penting, rahasia dagang, dll. Yang ketika diterbangkan melalui penerima paket akan dapat melewatinya. Penerima paket kemudian akan bekerja sebagai sniffer paket, mengendus semua paket yang dikirim memasuki jangkauan. Pertahanan terbaik melawan *packet sniffer* adalah kriptografi[14].

10. ***DNS Spoofing***

Ini tentang membahayakan komputer dengan merusak data *Domain Name System* (DNS) dan kemudian memasukkan cache resolver. Ini menyebabkan server nama mengembalikan alamat IP yang salah[14].

11. ***IP Spoofing***

Ini adalah proses menginjeksi paket di internet menggunakan alamat sumber palsu dan merupakan salah satu cara untuk menyamar sebagai pengguna lain. Otentikasi titik akhir yang memastikan kepastian pesan yang berasal dari tempat yang kami tentukan akan membantu melindungi dari *spoofing* IP[14].

12. ***Compromised Key***

Penyerang mendapatkan akses tidak sah ke komunikasi yang aman menggunakan kunci yang disusupi. Kunci mengacu pada nomor atau kode rahasia yang diperlukan untuk menafsirkan informasi yang diamankan tanpa pemberitahuan kepada pengirim atau penerima. Ketika kunci diperoleh oleh penyerang, itu disebut sebagai kunci yang disusupi yang berfungsi sebagai alat untuk mengambil informasi[14].

2.2.6 *Firewall*

Firewall (Tembok Api) adalah suatu sistem yang dirancang untuk mencegah akses yang tidak diinginkan dari atau ke dalam suatu jaringan internal[15].

Tembok api bekerja dengan cara melacak dan mengendalikan jalannya data serta memutuskan aksi untuk melewatkan (pass), menolak (reject), mengenkripsi atau melakukan pencatatan aktivitas (log) data. Firewall menjamin agar data sesuai dengan aturan (rule) yang terdapat di dalam kebijakan keamanannya (security policy) yaitu seperangkat aturan yang telah didefinisikan di dalam keamanan jaringan internal[15].

2.2.6.1 Manfaat*Firewall*

Dengan adanya firewall, manfaatnya adalah:

1. Menjadi pengatur lalu lintas atau trafik data terhadap jaringan satu dengan jaringan yang lain[16].
2. Mengatur port ataupun paket data yang di izinkan atau ditolak[16].
3. Mengautentifikasi terhadap akses[16].
4. Menjadi pemantau dan pencatat lalu lintas jaringan [16].

2.2.6.2 Filter Rules

Filter rule biasanya digunakan untuk melakukan kebijakan boleh atau tidaknya sebuah trafik ada dalam jaringan, identik dengan accept atau drop. Filter Rules terdapat 3 macam chain yang tersedia. Chain tersebut antara lain adalah Forward, Input, Output. Adapun fungsi dari masing-masing chain tersebut adalah sebagai berikut[17]:

- *Forward* :

Digunakan untuk memproses lalu lintas data paket yang hanya melewati router. Misalnya lalu lintas dari jaringan publik ke lokal atau sebaliknya dari jaringan lokal ke publik, contoh kasus seperti saat kita browsing. Lalu lintas laptop yang menjelajah Internet dapat dikelola oleh firewall menggunakan chain forward[17].

- *Input* :

Digunakan untuk memproses lalu lintas data paket yang masuk ke router melalui interface router dan memiliki alamat IP tujuan berupa IP yang terdapat pada router. Jenis lalu lintas ini dapat berasal dari jaringan publik atau dari jaringan lokal dengan router itu sendiri. Contoh: Mengakses router menggunakan winbox, webfig, telnet baik dari publik maupun lokal[17].

- *Output* :

Digunakan untuk memproses lalu lintas data paket yang meninggalkan router. Dengan kata lain, ini adalah kebalikan dari 'Input'. Jadi lalu lintas yang datang dari dalam router adalah untuk jaringan publik atau jaringan lokal. Misalnya, dari terminal winbox baru, kami melakukan ping ke ip Google. Lalu lintas ini kemudian dapat ditangkap dalam rantai keluar[17].

2.2.6.3 NAT (Network Address Translation)

Terdapat 2 macam opsi chain yang tersedia, yaitu dst-nat dan src-nat. Dan fungsi dari NAT sendiri adalah untuk mengubah Source Address maupun Destination Address. Kemudian fungsi dari masing-masing chain tersebut adalah sebagai berikut[17]:

- *dstnat* :

Memiliki fungsi untuk mengubah *Destination Address*dalam sebuah data. Biasa digunakan untuk membuat host dalam jaringan lokal dapat diakses dari luar jaringan (internet) dengan cara NAT akan menggantikan alamat IP tujuan paket dengan alamat IP lokal. Jadi intinya chain ini adalah untuk mengubah/mengganti IP tujuan dalam sebuah paket data[17].

-srcnat :

Memiliki fungsi untuk mengubah *Source Address*suatu data. Sebagai contoh, fungsi chain ini banyak digunakan ketika kita mengakses website dari jaringan LAN. Secara aturan untuk IP Address local tidak dapat masuk ke jaringan WAN, jadi konfigurasi 'srcnat' ini diperlukan. Sehingga IPlokal diganti dengan IP publik yang terpasang pada router[17].

2.2.6.4 MANGLE

Ada 4 jenis pilihan untuk chain, yaitu *Forward*, *Input*, *Output*, *Prerouting*, dan *Postrouting*. Mangle sendiri memiliki fungsi untuk menandai koneksi atau

paket data, yang melewati router, masuk ke router, atau keluar dari router. Dalamimplementasinya, Mangle sering dikombinasikan dengan fitur lain seperti *Management Bandwidth*, *Routing policy*, dll. Fungsi dari masing-masing chain di mangle adalah sebagai berikut[17]:

- *Forward, Input, Output* :

Untuk penjelasan mengenai Forward, Input, dan Output sebenarnya tidak jauh berbeda dengan apa yang telah diuraikan pada Filter rules diatas. Namun pada Mangle, semua jenis trafik paket data forward, input, dan output bisa ditandai berdasarkan koneksi atau paket atau paket data[17].

- *Prerouting* :

Adalah koneksi yang akan masuk ke router dan melewati router. Berbeda dengan input yang hanya menangkap lalu lintas yang menuju ke router. Lalu lintas yang melewati router dan lalu lintas yang masuk kerouter dapat ditangkap di chain prerouting[17].

- *Postrouting* :

Kebalikan dari prerouting, postrouting merupakan koneksi yang akan keluar dari router, baik untuk trafik yang melewati router ataupun yang keluar dari router [17].

2.2.7 NDLC (Network Development Life Cycle)

NDLC memiliki 6 tahap yaitu:

1. Analisis

Tahap awal ini dilakukan analisa kebutuhan, analisa permasalahan yang muncul, analisa keinginan user, dan analisa topologi / jaringan yang sudah ada saat ini. Metode yang biasa digunakan pada tahap ini diantaranya wawancara, survey, dan membaca manual atau blueprint dokumentasi[18].

2. Desain

Dari data-data yang didapatkan sebelumnya, tahap Design ini akan membuat gambar design topology jaringan interkoneksi yang akan dibangun, diharapkan dengan gambar ini akan memberikan gambaran seutuhnya dari kebutuhan yang ada. Desain bisa berupa design struktur topology, design akses

data, desain tata layout perkabelan, dan sebagainya yang akan memberikan gambaran jelas tentang project yang akan dibangun[18].

3. Simulasi

Beberapa networkers akan membuat dalam bentuk simulasi dengan bantuan Tools khusus di bidang network seperti BOSON, PACKET TRACER, NETSIM, dan sebagainya, hal ini dimaksudkan untuk melihat kinerja awal dari network yang akan dibangun dan sebagai bahan presentasi dan sharing dengan team work lainnya. Namun karena keterbatasan perangkat lunak simulasi ini, banyak para networkers yang hanya menggunakan alat Bantu tools VISIO untuk membangun topology yang akan didesain[18].

4. Implementasi

Di tahapan ini akan memakan waktu lebih lama dari tahapan sebelumnya. Dalam implementasi networkers akan menerapkan semua yang telah direncanakan dan di design sebelumnya. Implementasi merupakan tahapan yang sangat menentukan dari berhasil / gagalnya project yang akan dibangun dan di tahap inilah Teamwork akan diuji di lapangan untuk menyelesaikan masalah teknis dan non teknis[18].

5. Monitoring

Setelah tahap implementasi tahapan monitoring merupakan tahapan yang penting, agar jaringan komputer dan komunikasi dapat berjalan sesuai dengan keinginan dan tujuan awal dari user pada tahap awal analisis, maka perlu dilakukan kegiatan monitoring[18].

6. Manajemen

Di manajemen atau pengaturan, salah satu yang menjadi perhatian khusus adalah masalah Policy, kebijakan perlu dibuat untuk membuat / mengatur agar sistem yang telah dibangun dan berjalan dengan baik dapat berlangsung lama dan unsur Reliability terjaga. Policy akan sangat tergantung dengan kebijakan level management dan strategi bisnis perusahaan tersebut. IT sebisa mungkin harus dapat mendukung atau alignment dengan strategi bisnis perusahaan [18].

2.2.8 Penerapan IDS menggunakan Snort

Penelitian yang dilakukan oleh Sahid Aris Budiman, Catur Iswahyudi, dan Muhammad Sholeh, 2014. Judul penelitian Implementasi Intrusion Detection System (IDS) Menggunakan Jejaring Sosial Sebagai Media Notifikasi. Berdasarkan hasil pengujian yang dilakukan dapat diketahui kemampuan dari sistem yang dibuat mampu untuk mengolah data output dari IDS snort serta dapat mengenali segala aktifitas yang dilakukan intruder dalam usaha untuk menyusup ke dalam sistem dengan menggunakan ping flood, syn attack, port scanner, SSH dan FTP berdasarkan rule yang telah diterapkan. Proses selanjutnya adalah dilakukan blocking terhadap IP address yang dianggap sebagai intruder dan sistem akan memberikan laporan kepada administrator melalui media jejaring sosial dan web monitoring mengenai adanya intruder yang mencoba masuk ke dalam sistem. Tabel 2.2 menunjukkan kemampuan dari sistem untuk mengelola hasil output dari snort untuk mengenali terjadinya serangan sampai terjadinya proses blocking menggunakan iptables dari beberapa sampel yang telah diujicobakan[5].

Tabel 2.2: Selisih waktu yang dibutuhkan untuk blocking

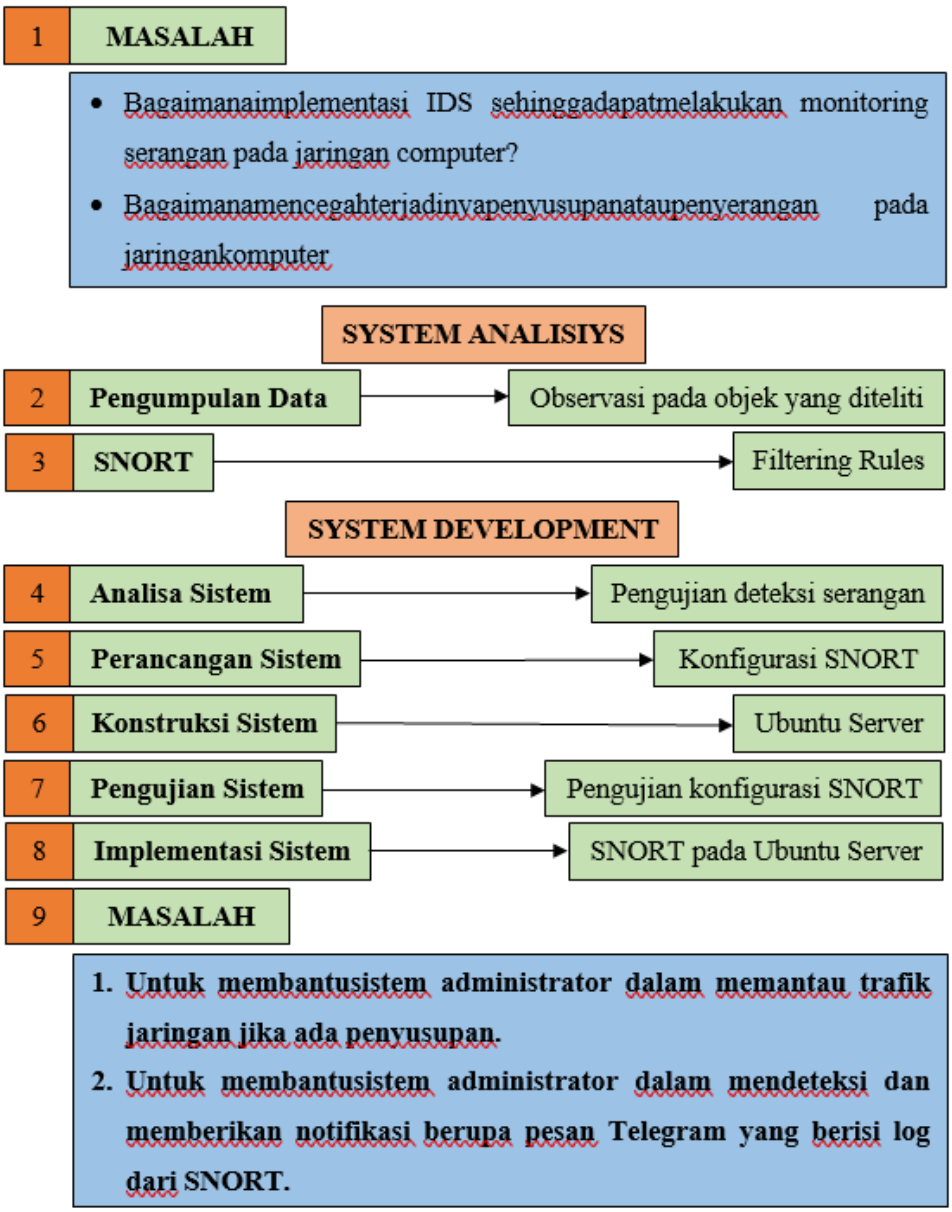
No	IP Adress	Waktu Serangan	Waktu Blocking	Selisih (detik)
1	202.91.10.214	08:36:24	08:36:30	6 detik
2	202.67.40.13	22:30:54	22:30:59	5 detik
3	10.15.74.157	18:08:37	18:08:44	7 detik
4	114.79.19.46	09:10:18	09:10:27	9 detik
5	202.67.40.12	23:34:24	23:34:31	7 detik
6	202.67.40.6	12:15:31	12:15:35	4 detik
7	202.67.40.11	14:14:35	14:14:39	4 detik
8	114.79.19.76	07:51:39	07:51:45	6 detik

2.2.9 Pengujian Sistem

Pengujian sistem adalah elemen penting dari jaminan kualitas perangkat lunak dan merupakan tinjauan utama dari spesifikasi, desain, dan pengkodean. Tujuan dari pengujian ini adalah untuk menunggu dengan sedikit usaha dan waktu untuk menemukan berbagai potensi *bug* dan konfigurasi pada jaringan komputer.

Pada tahap ini dilakukan pengujian sistem yang telah dilakukan. Pengujian berfokus pada logika internal perangkat lunak dan bagian eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan *bug* dan memastikan bahwa *input*terbatas memberikan hasil aktual yang sesuai dengan hasil yang diperlukan. Pada tahap ini, pengujian operasi juga dilakukan, yang mengarah pada kesiapan implementasi.

2.3 Kerangka Pikir



Gambar 2.2: Kerangka Pikir

BAB III

METODE PENELITIAN

3.1 Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian

Dalam penelitian ini penulis menggunakan metode eksperimental untuk mendapatkan data secara langsung dengan menerapkan teknik *scanning serta teknik filtering rules* untuk memantau log terhadap server yang telah di pasang IDS.

Berdasarkan latar belakang dan kerangka pikir seperti yang diuraikan diatas maka yang menjadi objek penelitian adalah analisis kinerja IDS menggunakan Snort. Penelitian ini dimulai dari 28 Februari 2021 s/d 4 April 2021 di VPS milik peneliti pribadi.

3.2 Pengumpulan Data

1. Data primer

Melakukan observasi langsung pada VPS pada objek yang diteliti

2. Data Sekunder

Diperoleh dengan cara mengumpulkan data atau keterangan melalui berbagai macam referensi seperti hasil penelitian terdahulu dan jurnal terkait dari internet yang berhubungan dengan analisis kinerja IDS menggunakan Snort.

3.3 Pengembangan Sistem

Prosedur dalam menganaliss dan merancang sistem menggunakan *tools NMAP* dalam hal uji coba serangan scanning dan tools snort untuk memonitoring log serangan yang masuk, dan dilakukan simulasi untuk menguji kinerja sistemnya.

3.3.1 Analisis Sistem

Sebelum melakukan analisis log IDS dalam sistem keamanan jaringan, ada beberapa tahan yang akan dilakukan oleh peneliti diantaranya:

1. Instalasi dan konfigurasi snort untuk menerapkan filtering rules pada VPS.
2. Melakukan pemantauan log IDS.

3.3.2 Desain Sistem

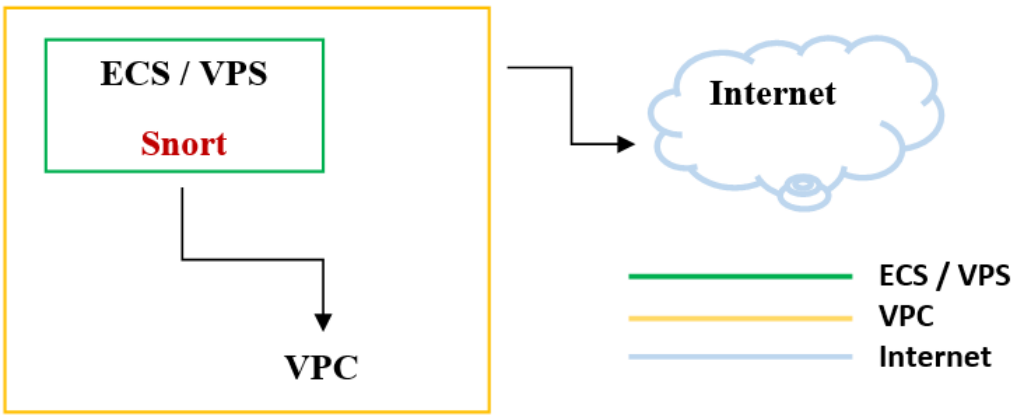
Desain sistem menggunakan pendekatan konfigruasi dalam hal menentukan kebijakan rules yang harus diterapkan pada sistem keamanan dengan teknik filtering.

Architecture Design, dalam bentuk:

- Konfigurasi Snort
- Spesifikasi hardware dan software yang digunakan:

1. Sistem Operasi: Ubuntu Linux 20.04.2 LTS
2. Processor: Intel(R) Xeon(R) CPU 2.50Ghz
3. RAM: 1GiB
4. Harddisk: 40GB

3.3.3 Topologi Jaringan



Gambar 3.1: Skema jaringan yang digunakan

3.3.4 Konstruksi Sistem

Pada tahap ini dilakukan perancangan sistem menggunakan Snort dengan teknik Filtering Rules untuk memantau log serta menguji kinerja Snort menggunakan simulasi dengan menyerang langsung pada server. Tahap ini juga penulis melakukan analisa terhadap hasil dan desain sistem sebelumnya.

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Berikut ini adalah log apache2 melalui observasi langsung pada websitehttp://gustiman.space yang kemudian datanya di *capture* dan bisa dilihat pada gambar berikut.

Gambar 4.1: access.log dari Apache2

Dari informasi gambar diatas dapat disimpulkan bahwa akses protokol http terdapat *error* pada beberapa URI, sehingganya perlu dilakukan analisis dan diidentifikasi lebih lanjut.

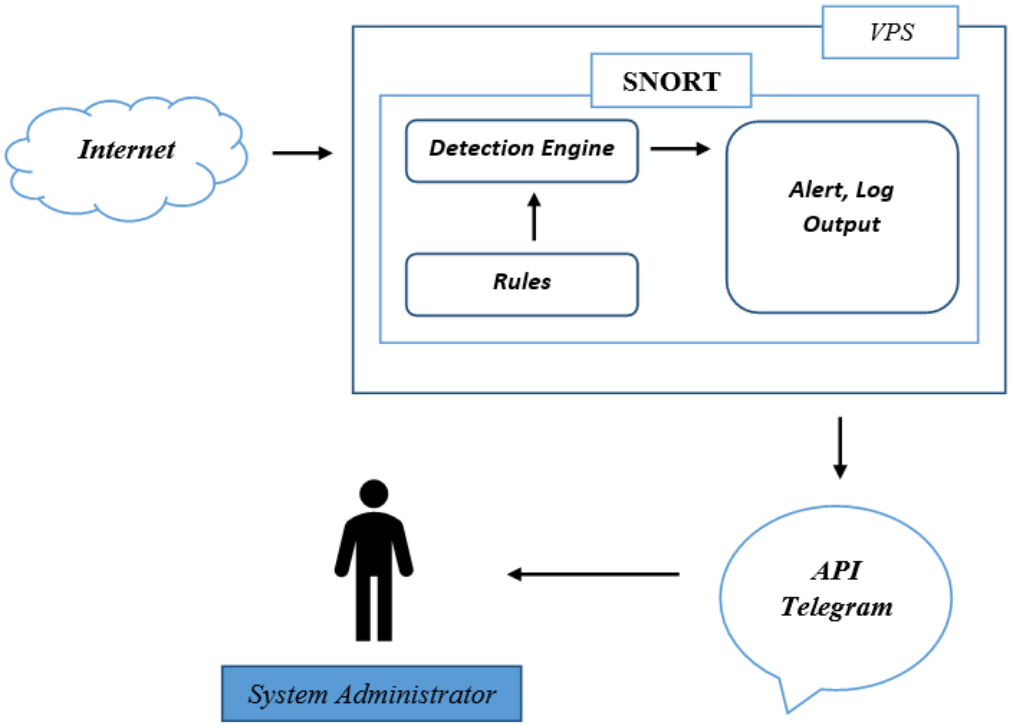
Gambar 4.2: error pada beberapa URI yang dicurigai sebagai sql injection

Nampak pada gambar 4.2 yang mana pada bagian yang diberi kotak merah tersebut merupakan kondisi akses protokol http yang mengalami error. Error

tersebut disebabkan oleh request URI dari client yang seharusnya disanitasi terlebih dahulu sebelum dieksekusi oleh server.

4.2 Hasil Pemodelan

Dalam menggambarkan pemodelan sistemdari NIDS Snort, maka peneliti membuat alur kerja dari penelitian iniyang dapat dilihat pada gambar 4.3. Paket dari internet masuk ke *VPS* akan difilter terlebih dahulu di Snort. Kemudian *Detection Engine* memeriksa apakah ada *rule* yang sesuai untuk paket yang melewati *Detection Engine*. Jika demikian, paket tersebut akan di*log* dan mengeluarkan *alert* kemudian *alert* tersebut akan di *trigger* ke *API Telegram* yang sudah terpasang pada sebuah script node js dan diteruskan langusng ke *System Admnistrator* untuk identifikasi lebih lanjut. Berikut ini merupakan penggambaran skema dari kondisi yang dimaksudkan, bisa dilihat langsung pada gambar 4.2



Gambar 4.3: Pemodelan Sistem

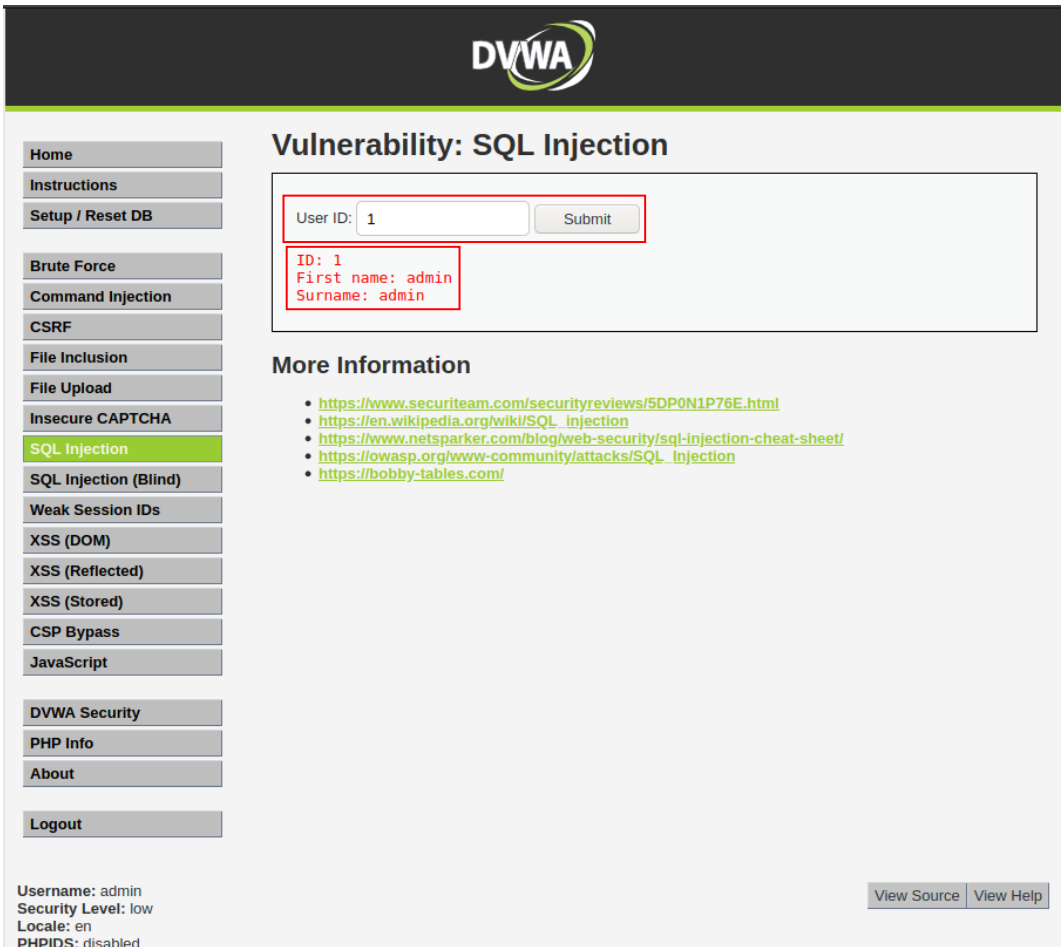
4.3 Hasil Pengembangan Sistem

Pada pengembangan sistem peneliti melakukan pengujian deteksi pada Snort dengan menggunakan *tools* WireShark, Burp Suite dan teknik penetrasi web *SQL Injection* agar bisa menciptakan rule yang sesuai dengan serangan yang diterima oleh server.

4.3.1 Perancangan Rule Detection untuk *SQL Injection Low Level*

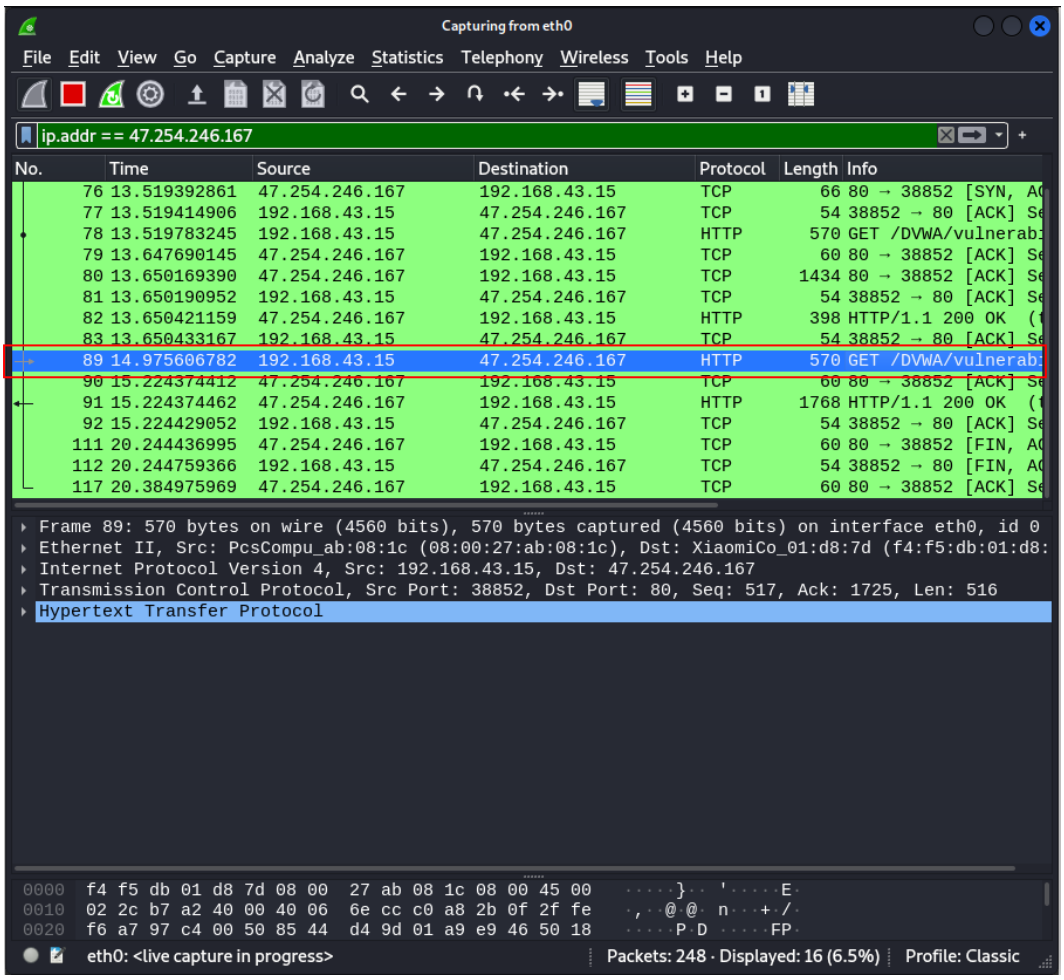
Serangan *SQL Injection* umumnya terdapat pada aplikasi web, dimana tidak adanya *filtering* atau sanitasi pada *query SQL* yang dimasukkan oleh client melalui web browser.

Dalam melakukan perancangan *rule detection* tersebut, diperlukan percobaan serangan dan *sniffing* menggunakan WireShark untuk menyadap paket yang keluar dan masuk dari komputer penyerang ke website target.



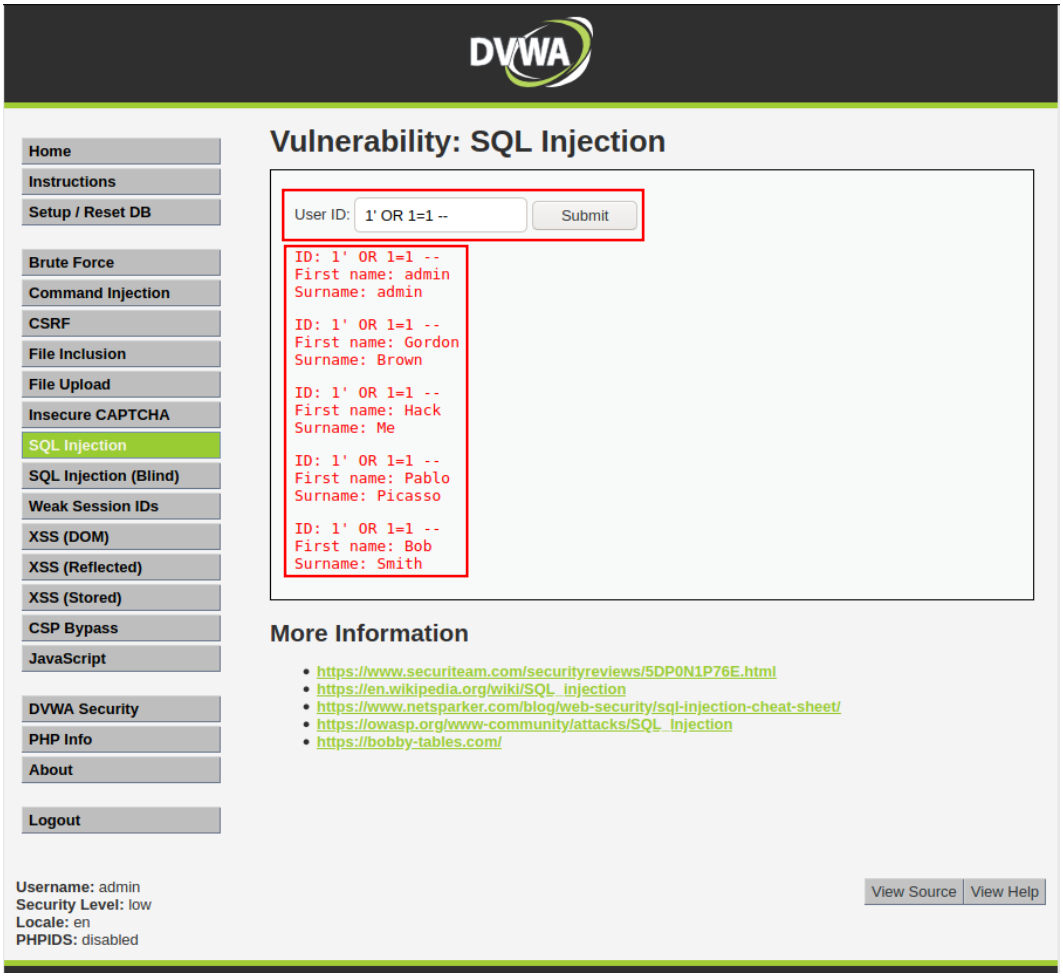
Gambar 4.4: Percobaan serangan dengan memasukkan User ID 1

Pada tahap ini (tampak pada gambar 4.4) penulis melakukan pengujian terhadap form User ID yang terdapat pada halaman website target untuk melihat hasil query SQL ke dalam database. Setelah dimasukkan angka 1, keluarlah user dengan ID 1 yang merupakan hasil query SQL (***SELECT first_name,last_name FROM users WHERE id = 1***).



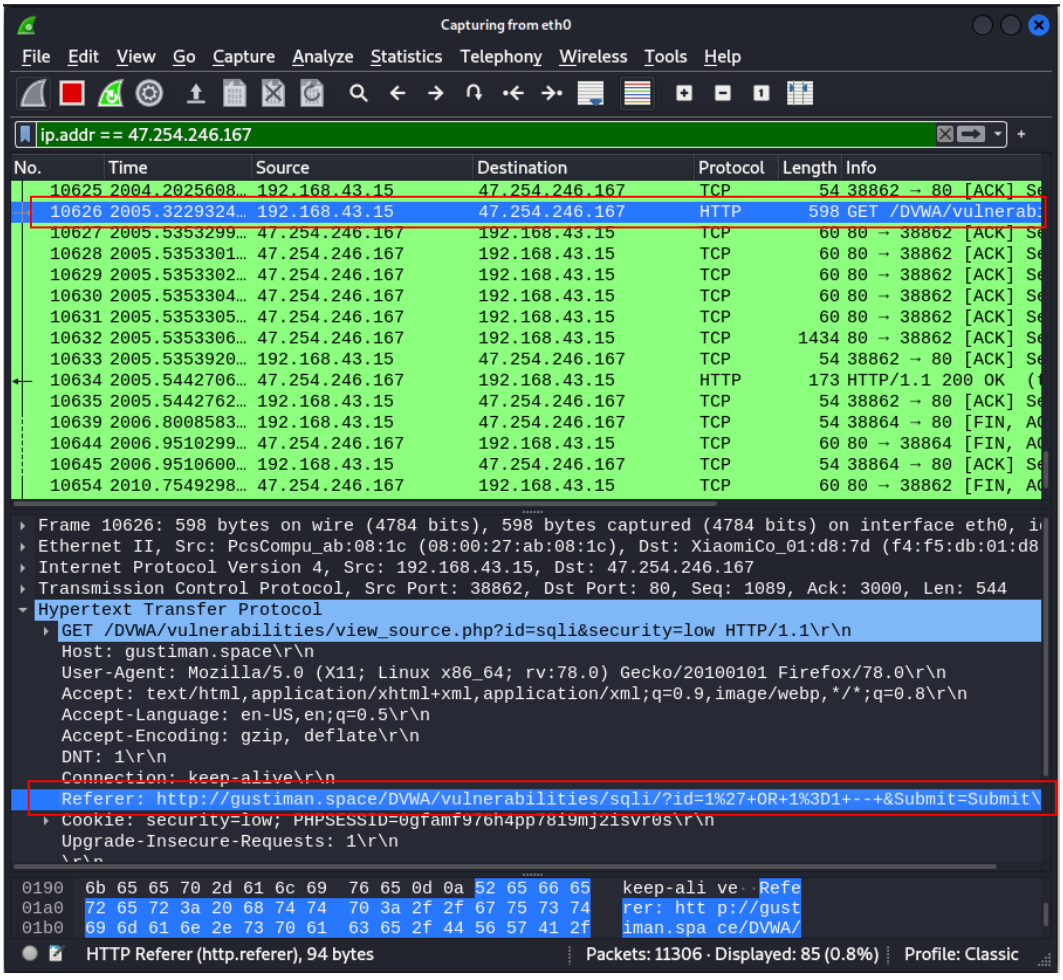
Gambar 4.5: *Tools* WireShark dengan memfilter IP dari website target

Bisa dilihat pada gambar (tampak pada gambar 4.5) diatas, terdapat request dengan method GET yang tertuju ke server beberapa saat setelah penulis memasukkan User ID 1. Apabila server menerima query SQL dengan HTTP GET maka request tersebut dapat dilihat langsung oleh user pada kolom url browser.



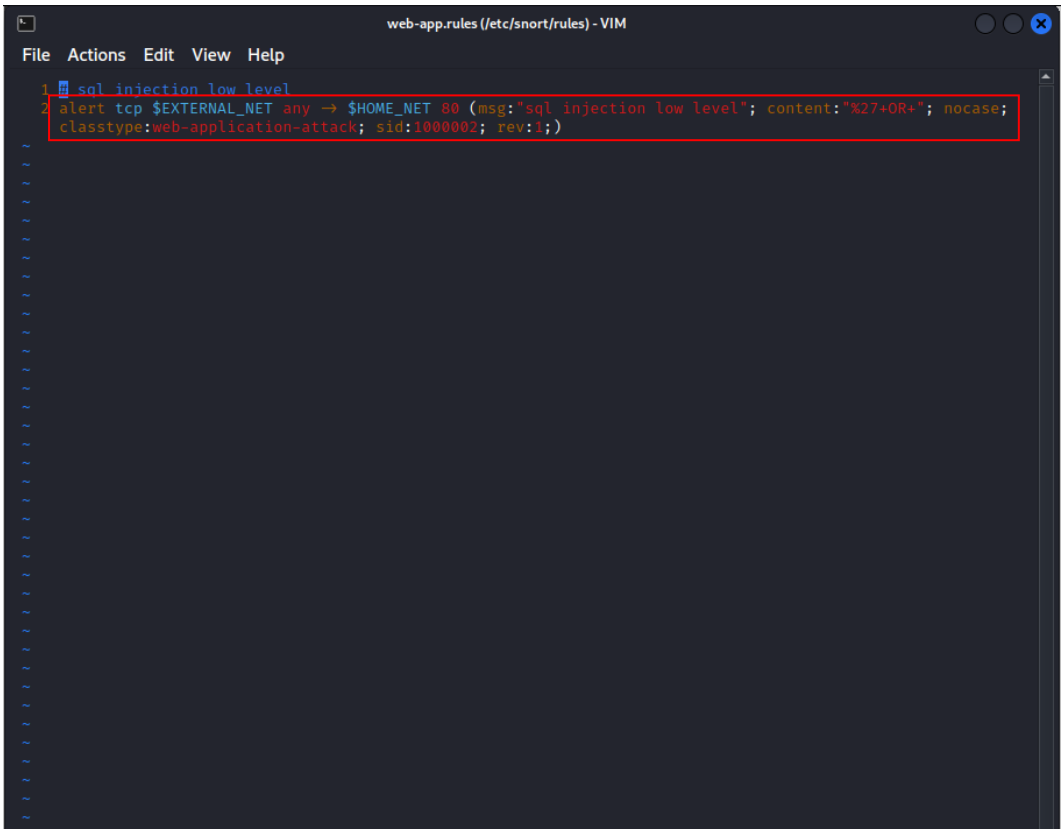
Gambar 4.6: Pengujian terhadap form User ID dengan memasukkan query SQL 1’ or 1=1 --

Gambar 4.6 menunjukkan bahwa SQL melakukan query tersebut tanpa melakukan sanitasi terhadap form User ID sehingga MySQL mencetak semua user yang terdapat pada database ke halaman tersebut. Penulis memasukkan 1’ untuk mengecek user dengan id 1 dan menambahkan operator logika OR 1=1 untuk mendapatkan hasil true karena logika OR akan menjadi true ketika salah satu input bernilai true dan – (spasi terakhir) untuk komen pada SQL sehingga hasil dari query tersebut adalah tampilkan user dengan id 1 atau 1=1 yang menghasilkan nilai true. Kemudian penulis melakukan analisa pada *tools* WireShark untuk melihat paket apa saja yang keluar dan masuk pada sisi client.



Gambar 4.7: Hasil packet capture dari WireShark

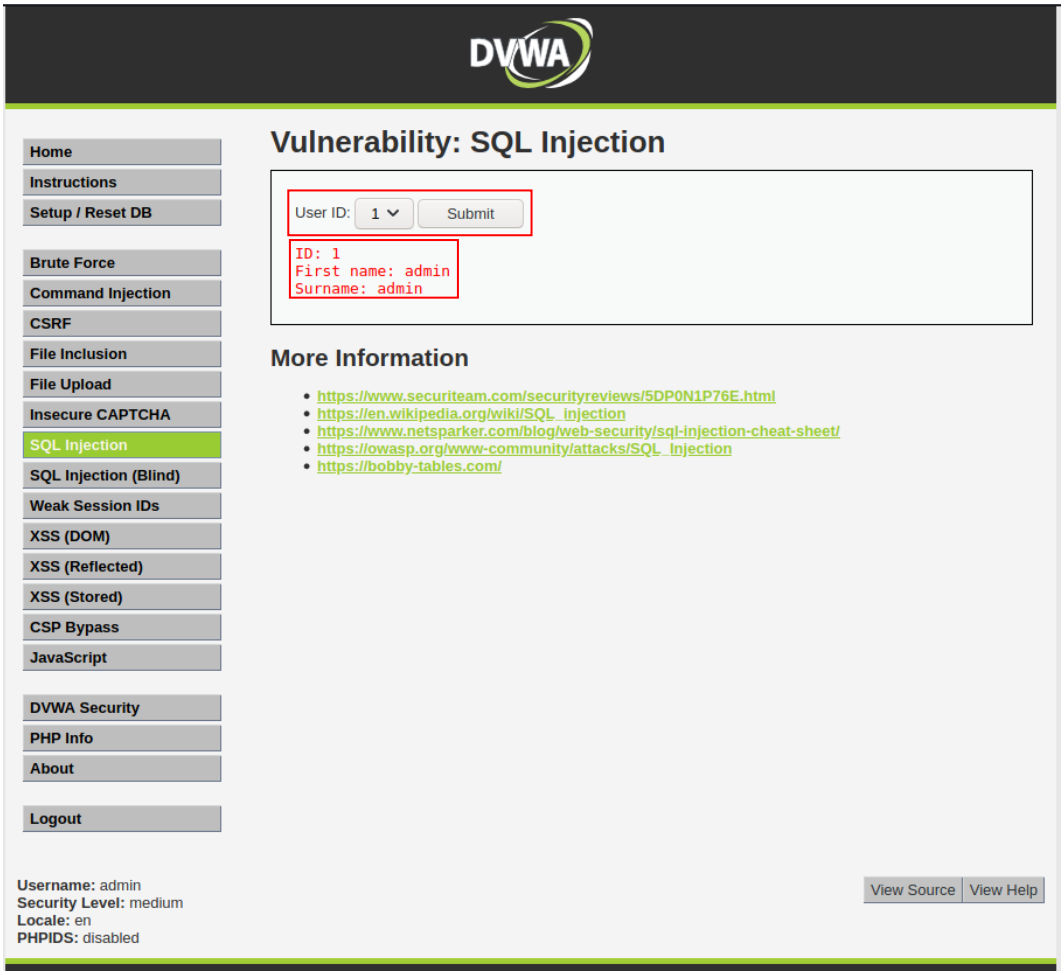
Dari hasil *packet caputure* tersebut kita bisa melihat ada request HTTP dengan method GET dari client dengan URI ***GET /DVWA/vulnerabilites/sqli/?id=1%27+or+1%3D1+--+&Submit=Submit***. Setelah dianalisa packet caputre tersebut maka penulis membuat rule untuk SQL Injection Low Level seperti pada gambar dibawah ini.



Gambar 4.8: Rule untuk SQL Injection Low Level

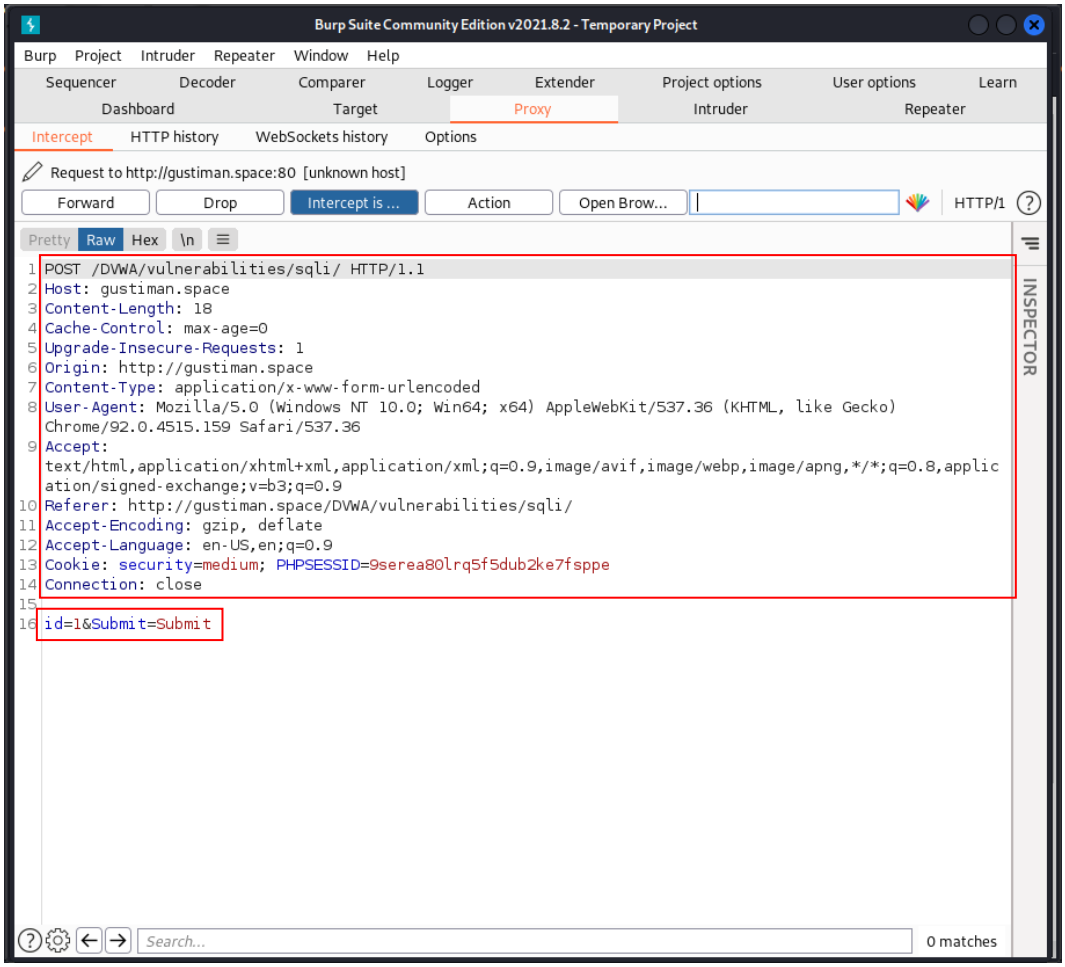
4.3.2 Perancangan rule detection untuk *SQL Injection Medium Level*

Pada perancangan rule untuk SQL Injection Medium Level penulis melakukan *intercept* dengan *tools* Burp Suite. Langkah pertama melihat tampilan awal dari website target dan melakukan percobaan pada beberapa fungsi yang ada pada website target.



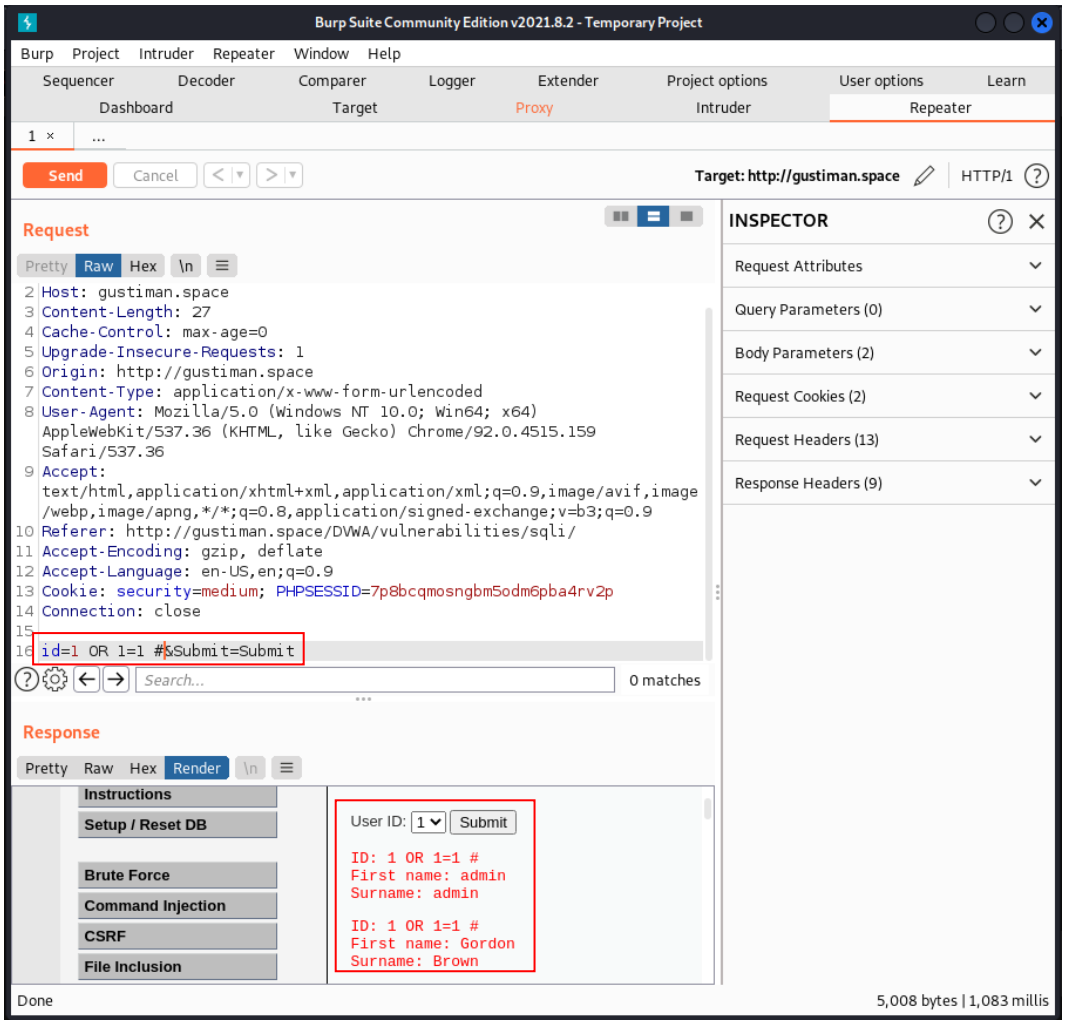
Gambar 4.9: Tampilan awal website target untuk pengujian SQL Injection Medium Level

Tampak pada gambar 4.9 bahwa hanya terdapat select option dan tombol submit untuk memilih serta menampilkan user berdasarkan ID yang dipilih. Penulis melakukan percobaan pada User ID 1 dan melihat apakah respon yang diberikan oleh server. Untuk SQL Injection Medium Level penulis menyimpulkan bahwa proses query SQL ke database server menggunakan method POST. Sedikit berbeda dengan SQL Injection Low Level yang menggunakan methode GET, pada level ini penulis menggunakan *tools* Burp Suite untuk intersepsi komunikasi yang terjadi pada client server dengan mem-*proxy* browser ke localhost terlebih dahulu sebelum diteruskan ke website target.

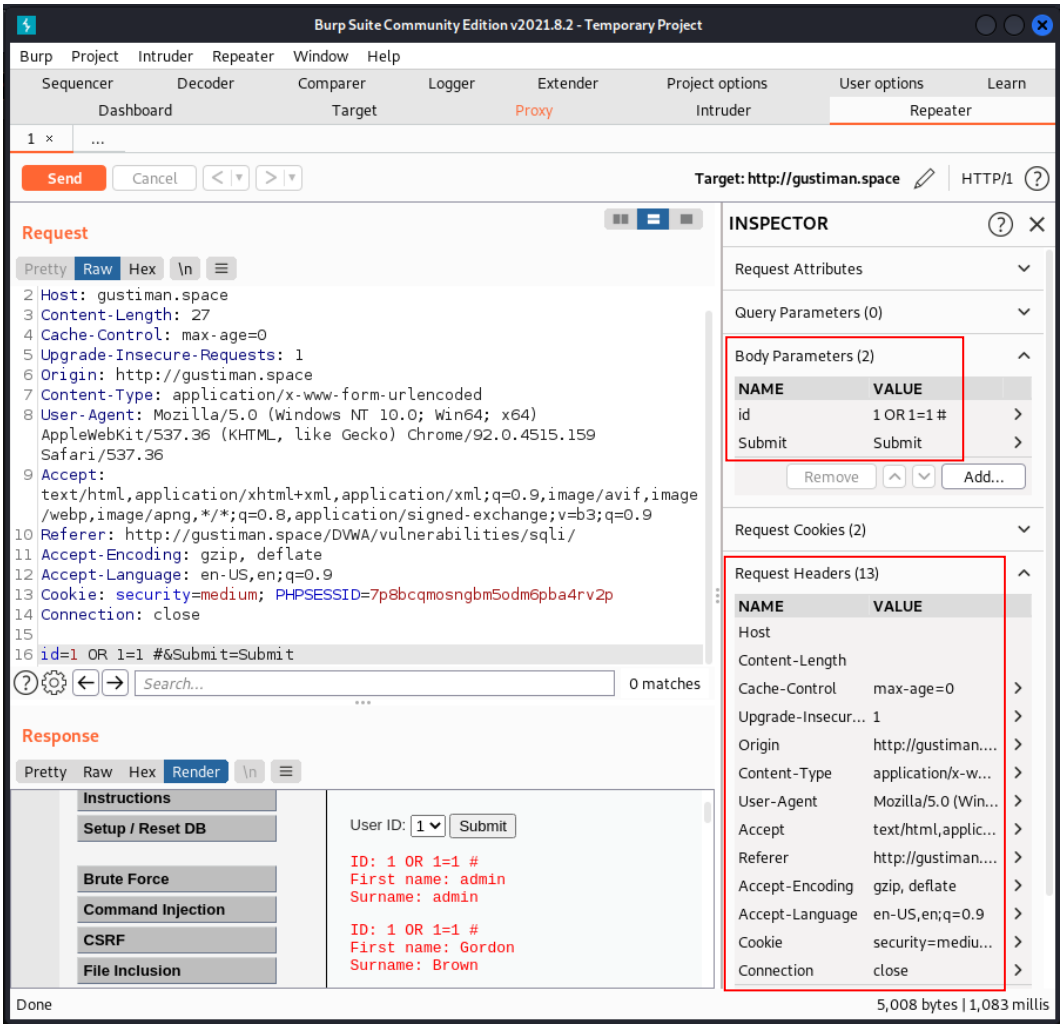


Gambar 4.10: Hasil intersepsi komunikasi client server

Setelah melakukan *intercept* pada website target dengan *tools* Burp Suite penulis melihat ada *request body* di proses query User ID ke server target. Penulis melakukan pengujian terhadap request body tersebut pada fitur Burp Suite yakni Repeater untuk melihat respon dari server target dengan memasukkan query yang sama pada pengujian level sebelumnya yakni ***id=1 OR 1=1 #&Submit=Submit*** pada request body.

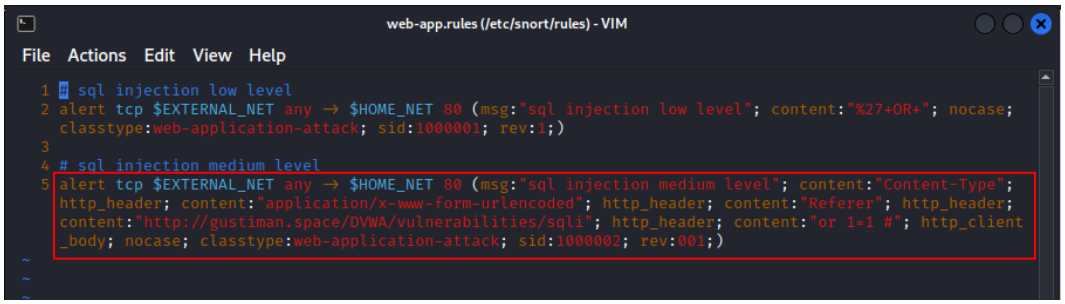


Gambar 4.11: Hasil pengujian *request body*



Gambar 4.12: *Body Parameters* dan *Request Headers*

Kemudian penulis membuat rule dari hasil pengujian sebelumnya dengan mengambil beberapa *request parameter* dan *request body* sebagai *content* untuk deteksi pada *rule options* yang akan dibuat. Maka rule final untuk pengujian SQL Injection Medium Level dapat dilihat pada gambar 4.13 dibawah ini.

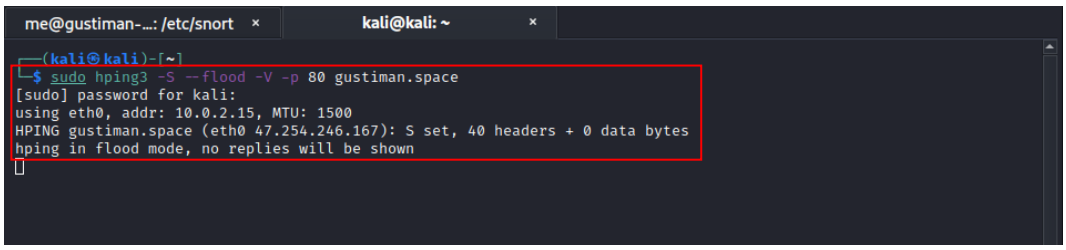


Gambar 4.13: Rule untuk SQL Injection Medium Level

4.3.3 Perancangan Rule Detection untuk Denial of Service

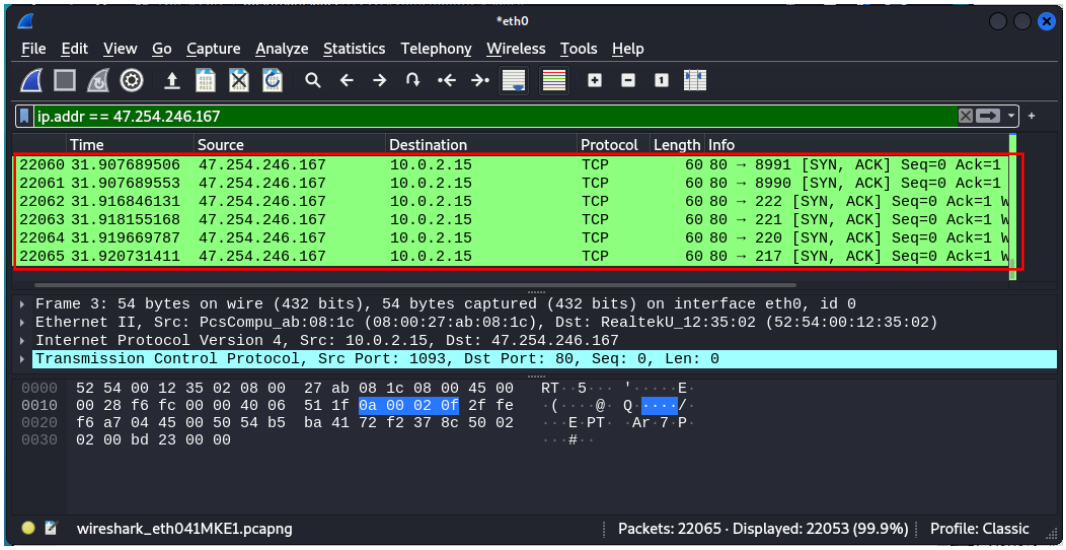
Denial of Service (DoS) merupakan serangan pada sisi server dimana dengan mengirimkan packet yang begitu banyak sehingga server kehabisan resource yang menyebabkan server *down* dan tidak bisa di akses.

Pada perancangan rule ini, seperti biasa penulis menggunakan *tools* WireShark untuk melihat packet apa saja yang dikirimkan oleh client ke server sehingga bisa menciptakan rule yang sesuai dengan serangan yang dimaksud.



Gambar 4.14: DoS menggunakan tools hping3 pada OS Kali Linux

Penulis melakukan percobaan DoS menggunakan *tools* hping3 dengan mengirimkan tcp flag syn dan menargetkan port 80, kemudian melakukan *sniffing* menggunakan *tools* WireShark untuk mengidentifikasi paket apakah yang dikirimkan oleh client tadi.



Gambar 4.15: *Packet Capture* dari client ke server

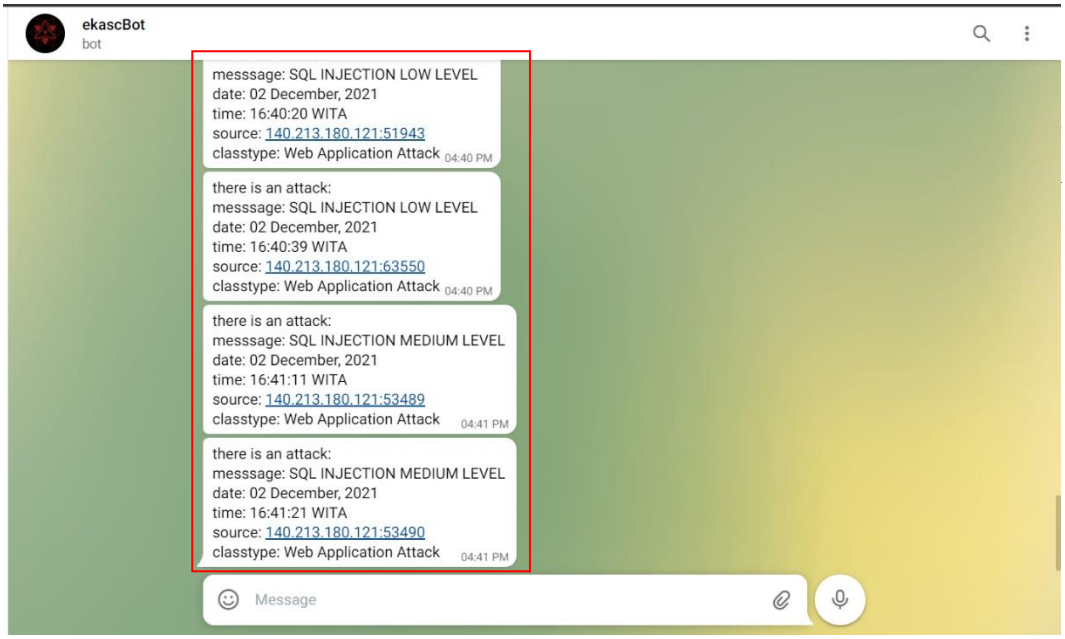
Tampak pada gambar 4.15, terlihat di *tools* WireShark bahwa client mengirimkan packet dengan TCP Flags Syn Ack yang begitu banyak yang menyebabkan server hampir kehabisan sumber daya. Sehingganya penguji mengambil flags syn untuk dijadikan pattern matching terhadap rule ini seperti gambar dibawah ini.

Gambar 4.16: Rule untuk Denial of Service

BAB V
PEMBAHASAN

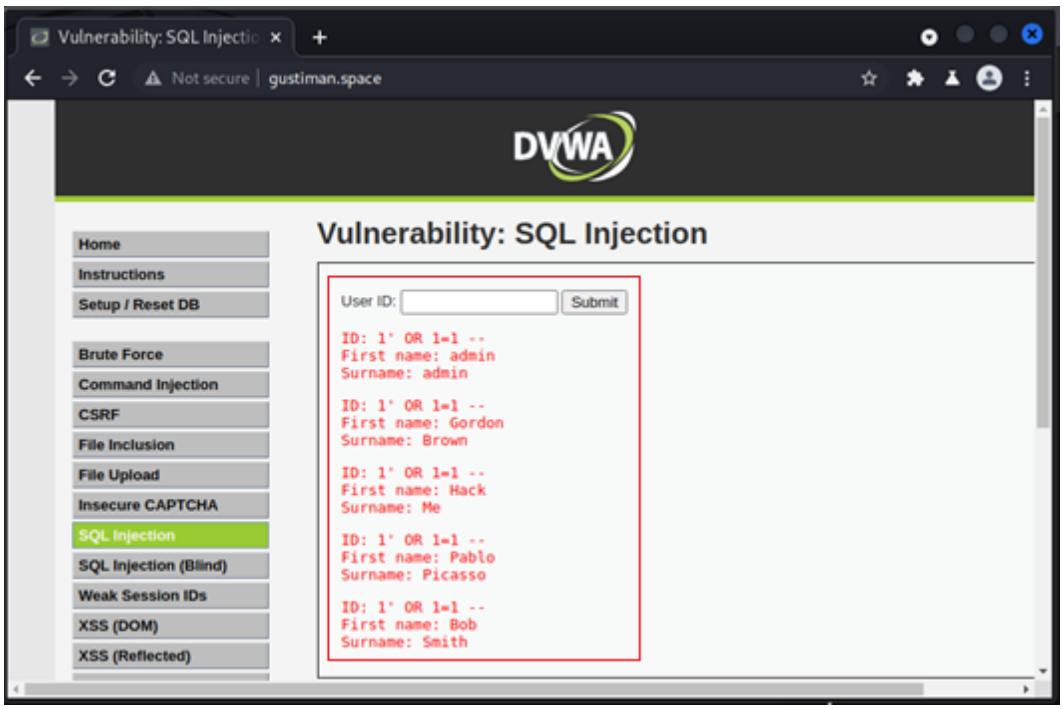
5.1 Pembahasan Model

Pada percobaan Snort IDS terhadap *SQL Injection* pada situs http://gustiman.space dilakukan monitoring Snort Log diperoleh berupa alert Telegram yang terdapat beberapa informasi seperti pesan, tanggal, waktu, sumber, dan tipe serangan yang sedang terjadi pada website tersebut.



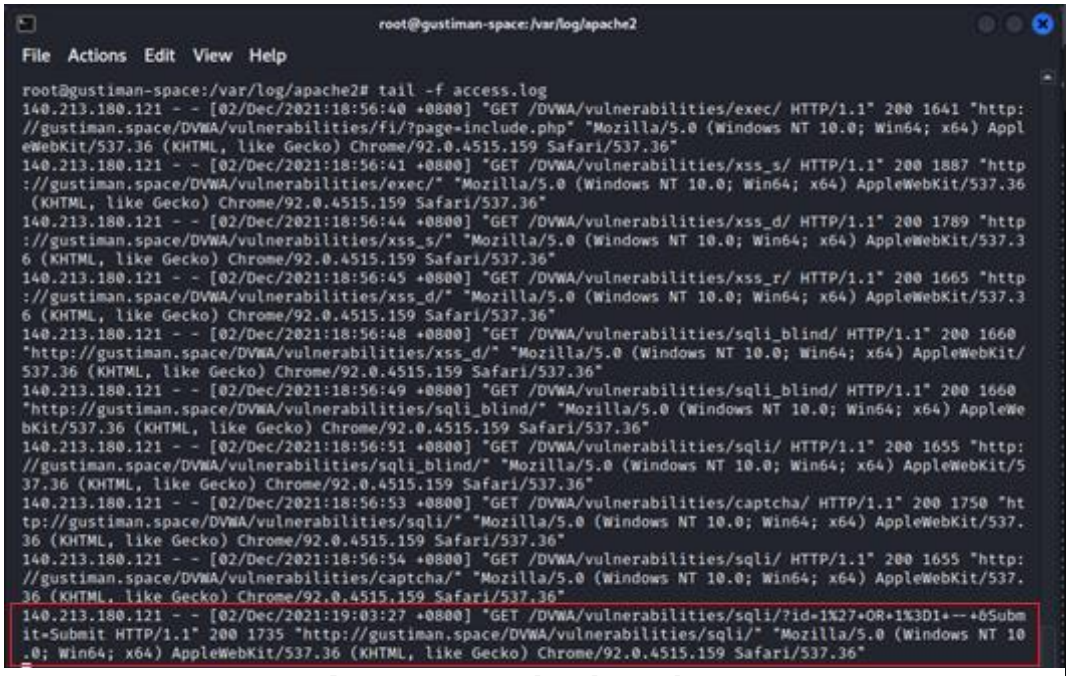
Gambar 5.1: Alert yang dihasilkan oleh Snort IDS

5.2.1 Pengjian *SQL Injection Low Level* tanpa *Rule Detection*



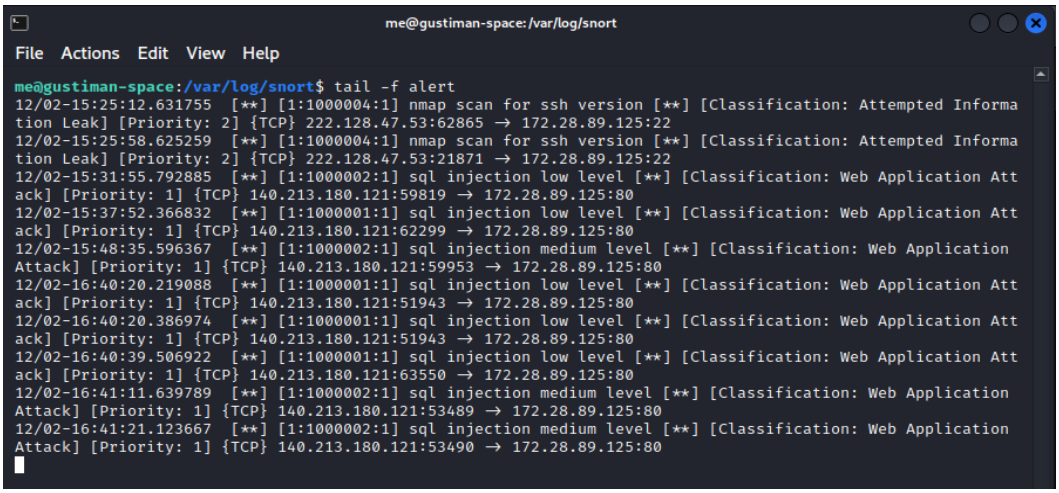
Gambar 5.2: Pengujian SQL Injection Low Level tanpa Rule Detection

Penguji melakukan Query 1’ OR 1=1 – untuk mengekstrak semua user yang terdapat dalam database pada server dan dapat dilihat pada browser.



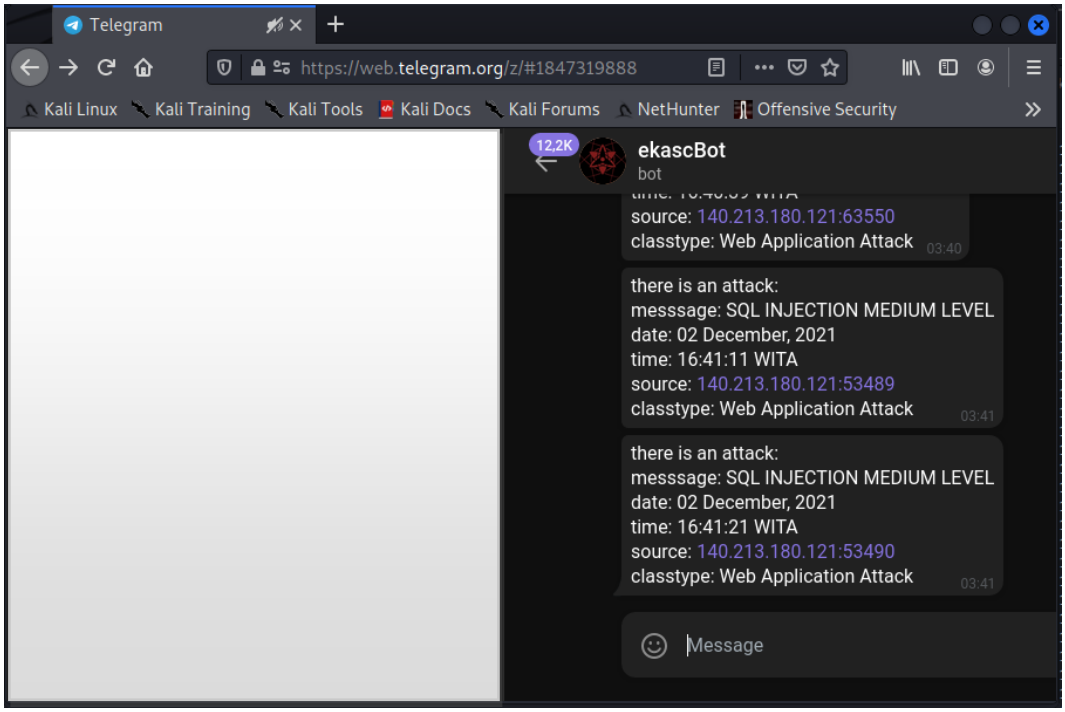
Gambar 5.3: access.log di apache2 server

Gambar diatas adalah log realtime dihasilkan oleh apache2 server yang dipicu oleh error pada SQL.



Gambar 5 4: Log di Snort IDS

Tidak terdapat alert pada Telegram dan Snort IDS seperti yang tampak pada gambar 5.4 dan gambar 5.5.



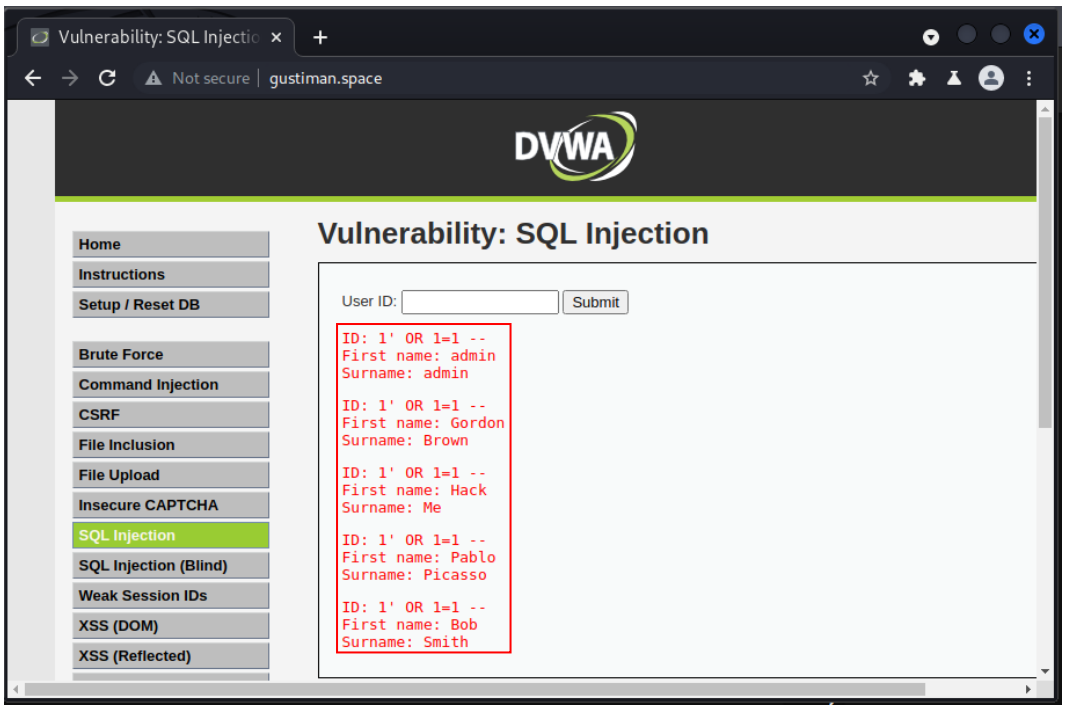
Gambar 5.5: Alert pada Telegram

Tabel 5.1: Hasil pengujian *SQL Injection Low Level tanpa Rule Detection*

IP Adrress	Apache2 Log	Snort Log	Telegram Alert
140.213.180.121	Yes	No	No

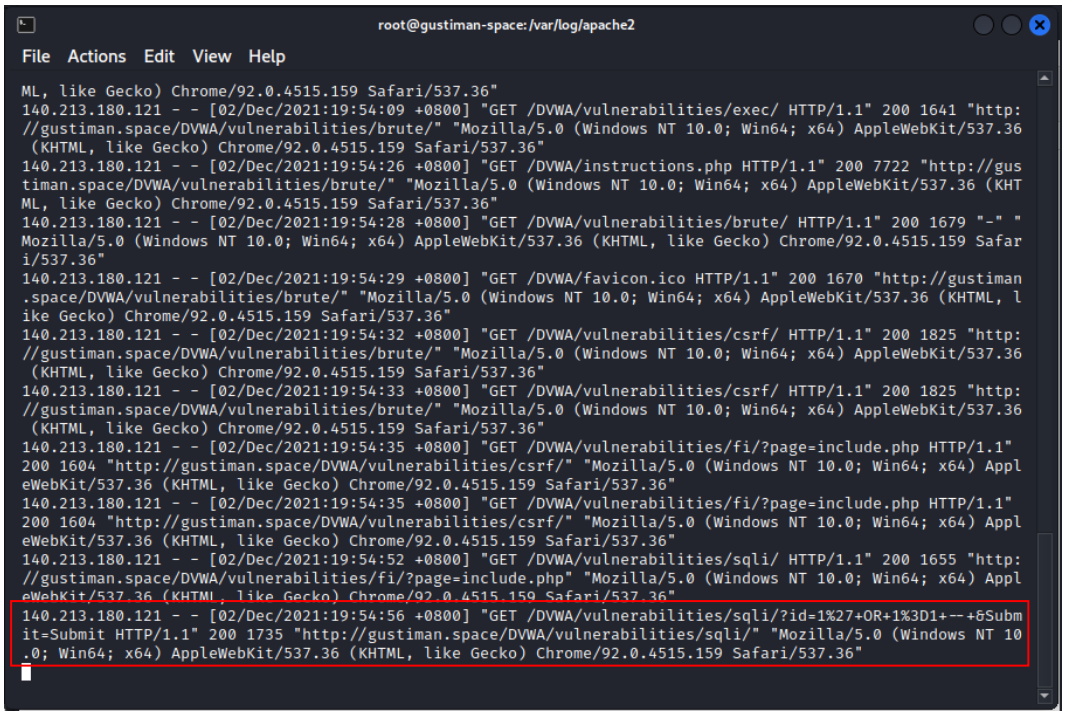
Hasil pengujian *SQL Injection Low Level* tanpa *Rule Detection* berupa serangan bisa terlihat jelas pada Apache2 Log namun tidak pada Snort Log dan Telegram Alert.

5.2.2 Pengujian SQL Injection Low Level dengan Rule Detection



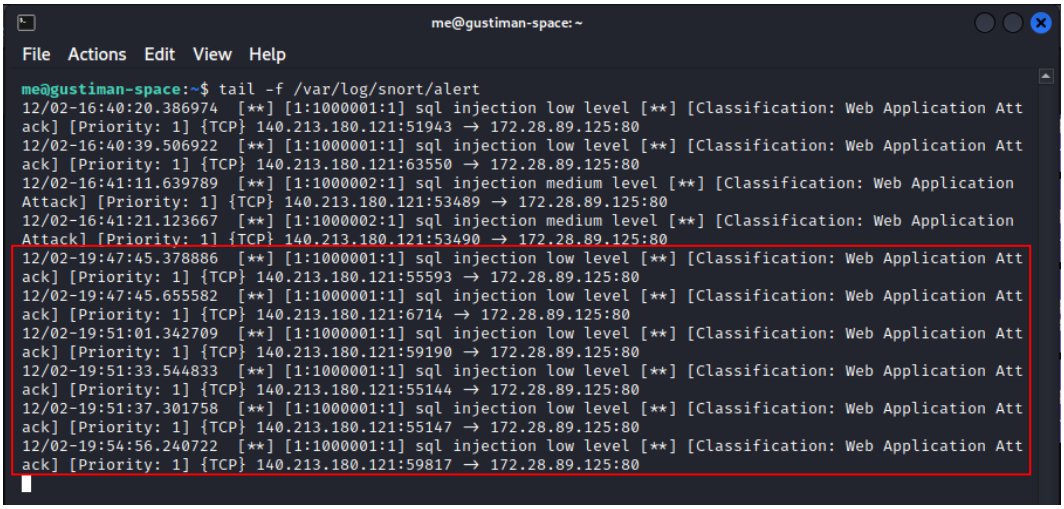
Gambar 5.6: Pengujian *SQL Injection Low Level* dengan *Rule Detection*

Seperti biasa, penulis melakukan pengujian seperti pada pengujian sebelumnya yakni dengan memasukkan Query 1’ OR 1=1 – untuk mengekstrak semua user yang terdapat dalam database server target.



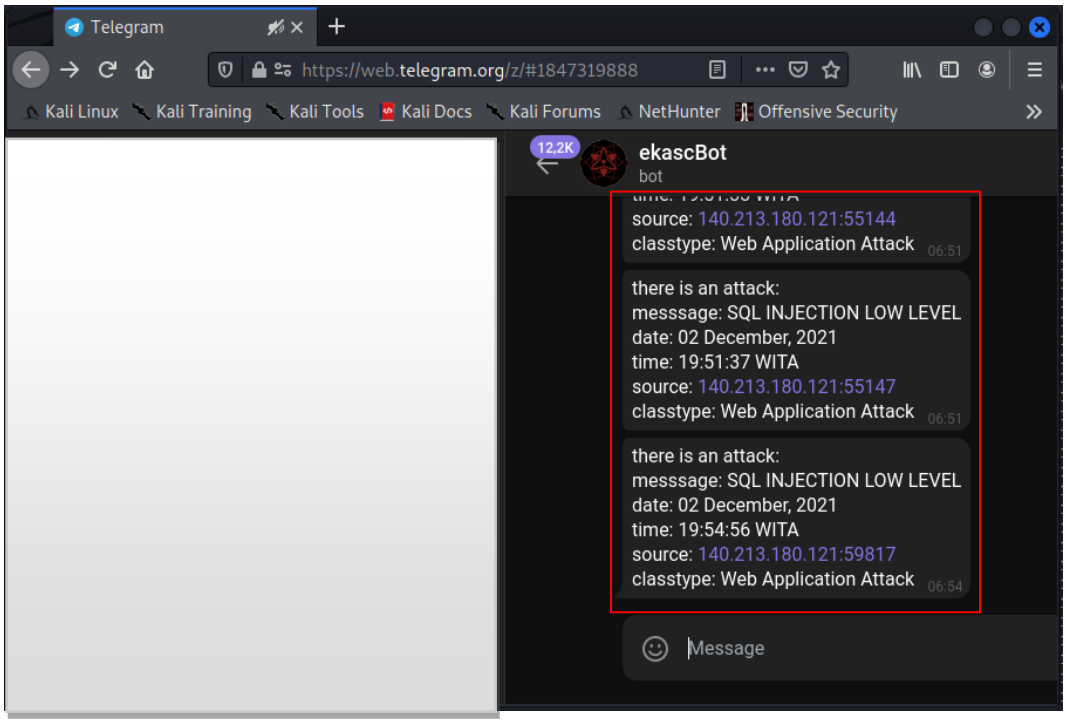
Gambar 5.7: access.log di apache2 server

Log yang terdapat di sisi server yang hasilkan oleh SQL error yang terdapat pada apache2 server.



Gambar 5.8: Log di Snort IDS

Log snort yang terdeteksi diserver dan dikirimkan langsung ke telegram dengan menggunakan bot.



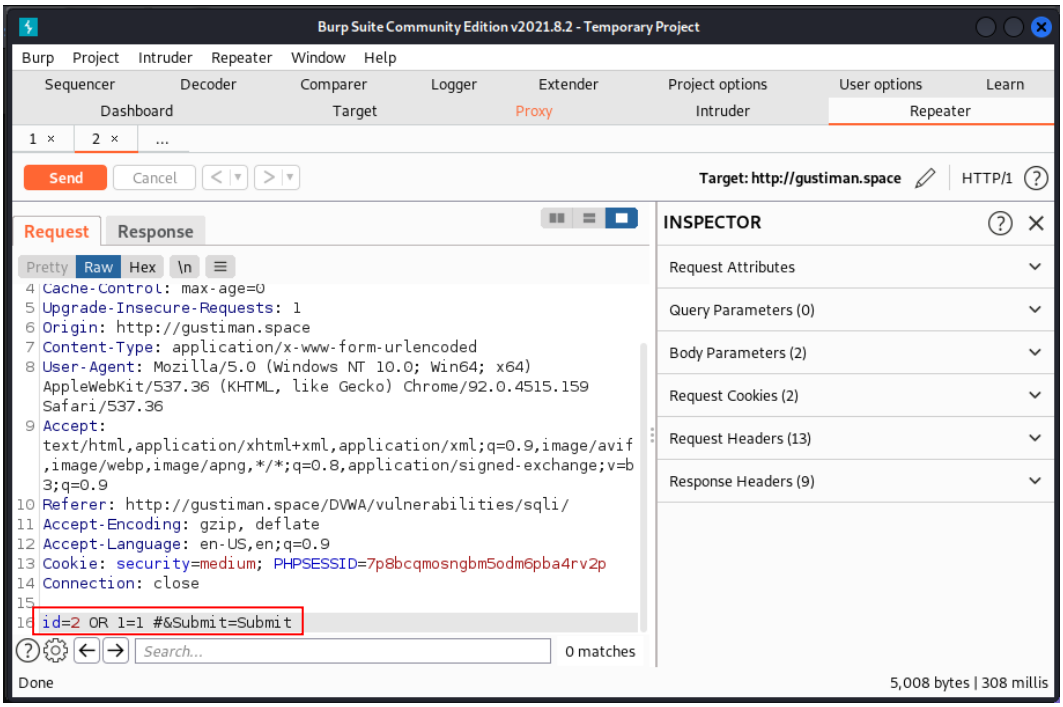
Gambar 5.9: Alert pada Telegram

Tabel 5.2: Hasil pengujian *SQL Injection Low Level* dengan *Rule Detection*

IP Adrress	Apache2 Log	Snort Log	Telegram Alert
140.213.180.121	Yes	Yes	Yes

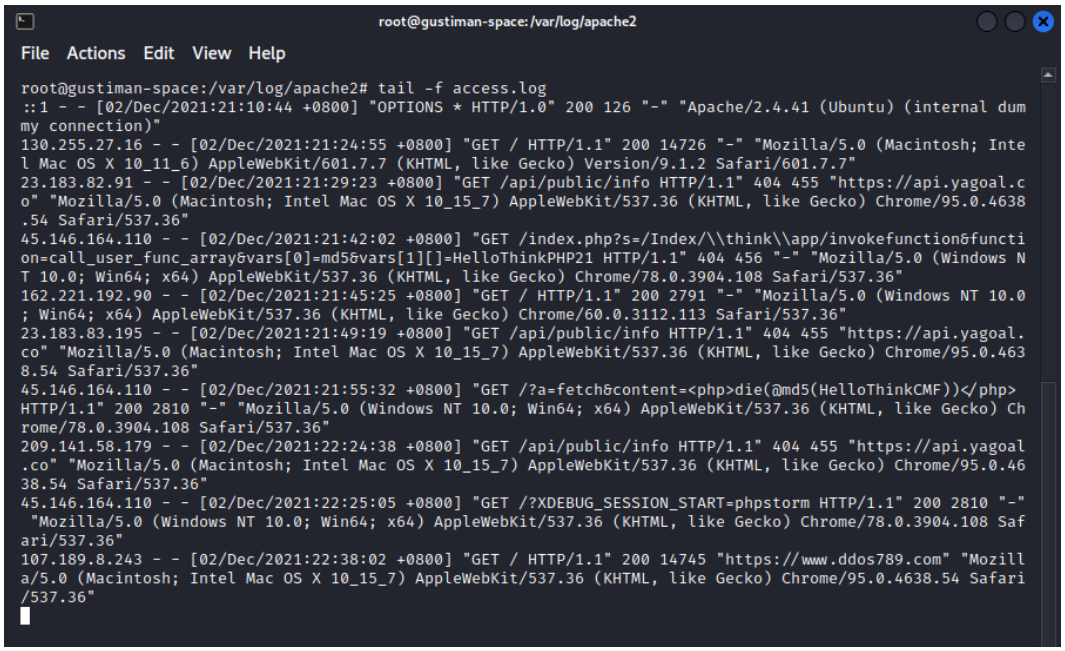
Hasil pengujian *SQL Injection Low Level* dengan *Rule Detection* berupa serangan dapat terlihat jelas pada Apache2 Log, Snort Log, dan Telegram Alert.

5.2.3 Pengujian SQL Injection Medium Level tanpa Rule Detection



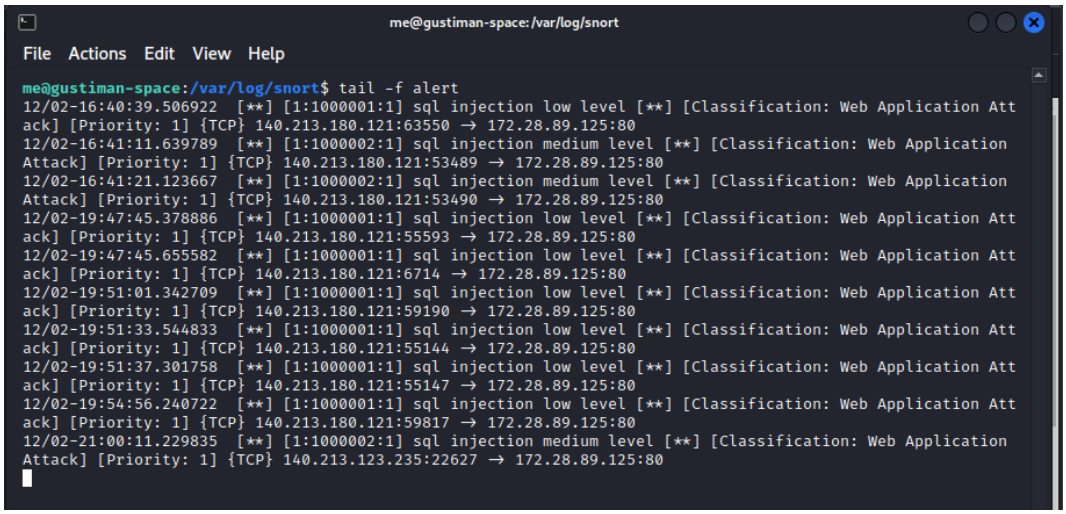
Gambar 5.10: Pengujian *SQL Injection Medium Level* tanpa *Rule Detection*

Penulis melakukan pengujian SQL Injection dengan menggunakan tools Burp Suite dengan cara memasukkan querry SQL 1 OR 1=1 #.



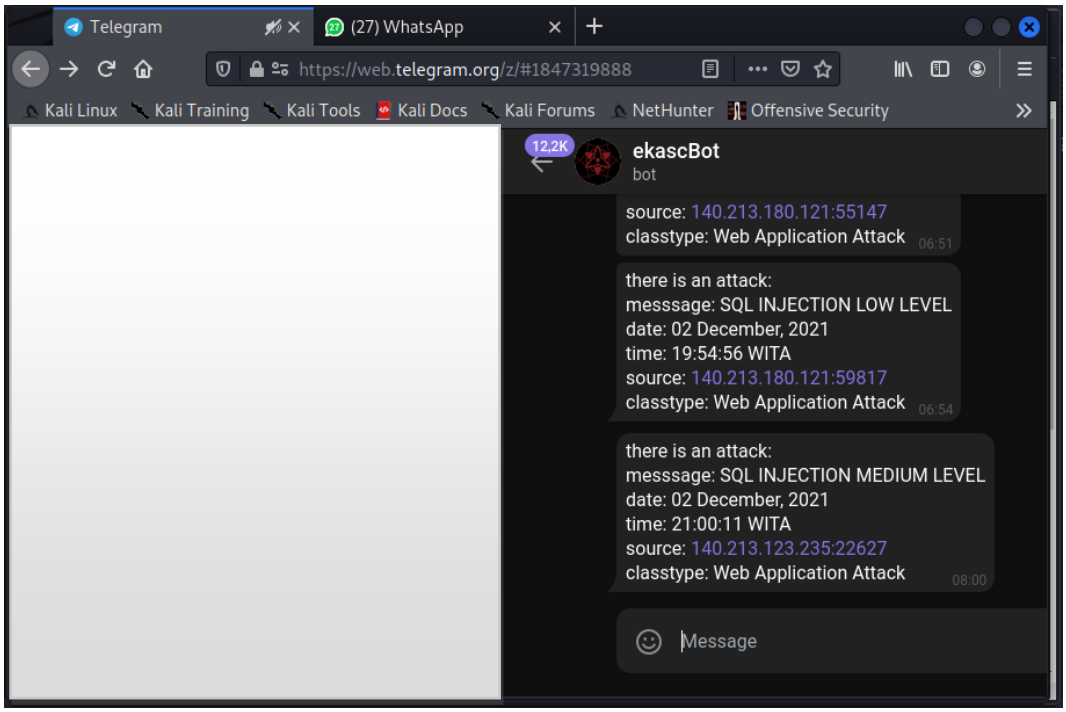
Gambar 5.11: access.log pada apache2 server

Log yang terdapat pada apache2 server.



Gambar 5.12: Log di Snort IDS

Log yang terdapat pada snort IDS dan alert pada telegram.



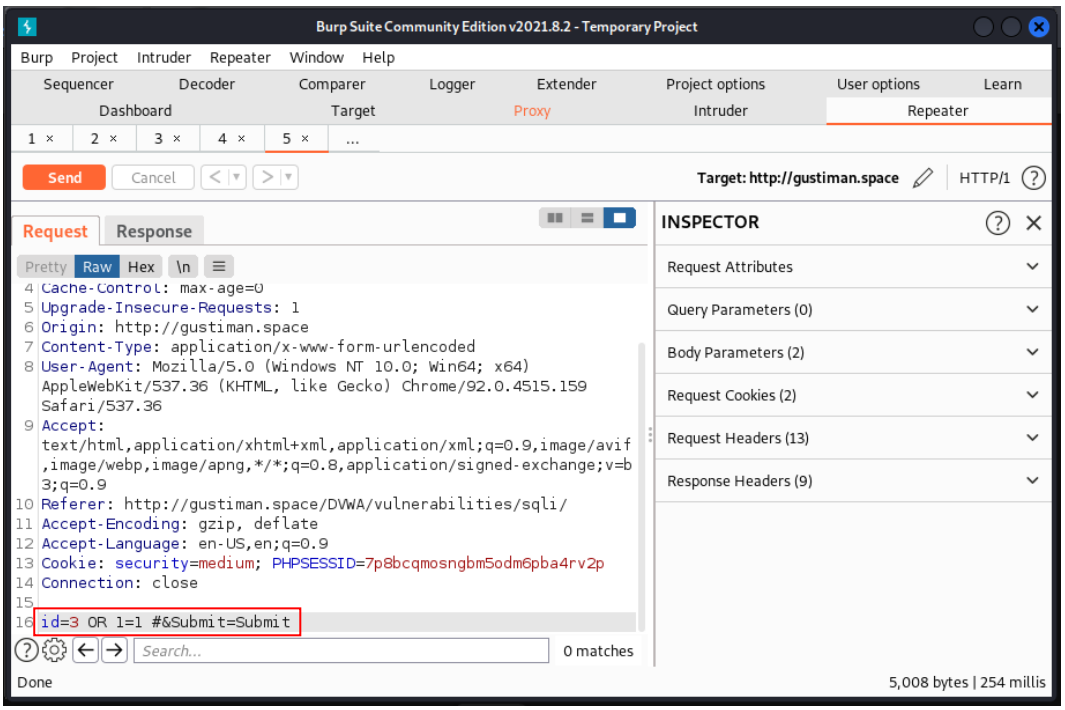
Gambar 5.13: *Alert* pada Telegram

Tabel 5.3: Hasil pengujian *SQL Injection Medium Level* tanpa *Rule Detection*

IP Adrress	Apache2 Log	Snort Log	Telegram Alert
140.213.123.45	No	No	No

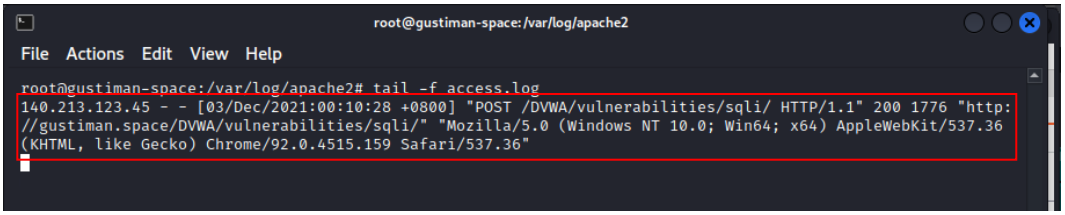
Hasil pengujian *SQL Injection Medium Level* tanpa *Rule Detection* tidak terdapat log pada Apache2, Snort, serta Telegram Alert.

5.2.4 Pengujian SQL Injection Medium Level dengan Rule Detection



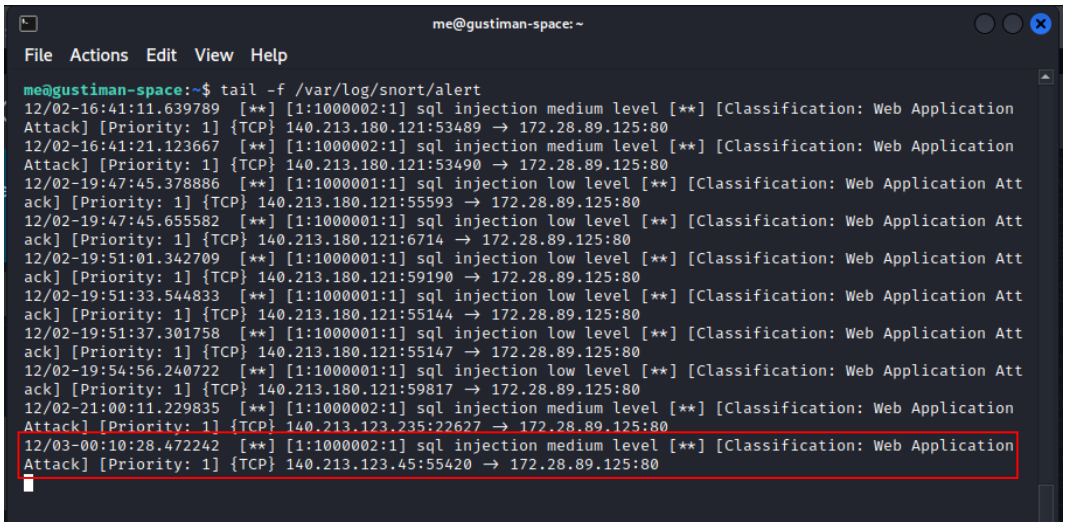
Gambar 5.14: Pengujian *SQL Injection Medium Level* dengan *Rule Detection*

Seperti biasa penulis melakukan pengujian SQL injection yang sama pada pengujian sebelumnya dengan memasukkan query yang tampak pada gambar diatas.

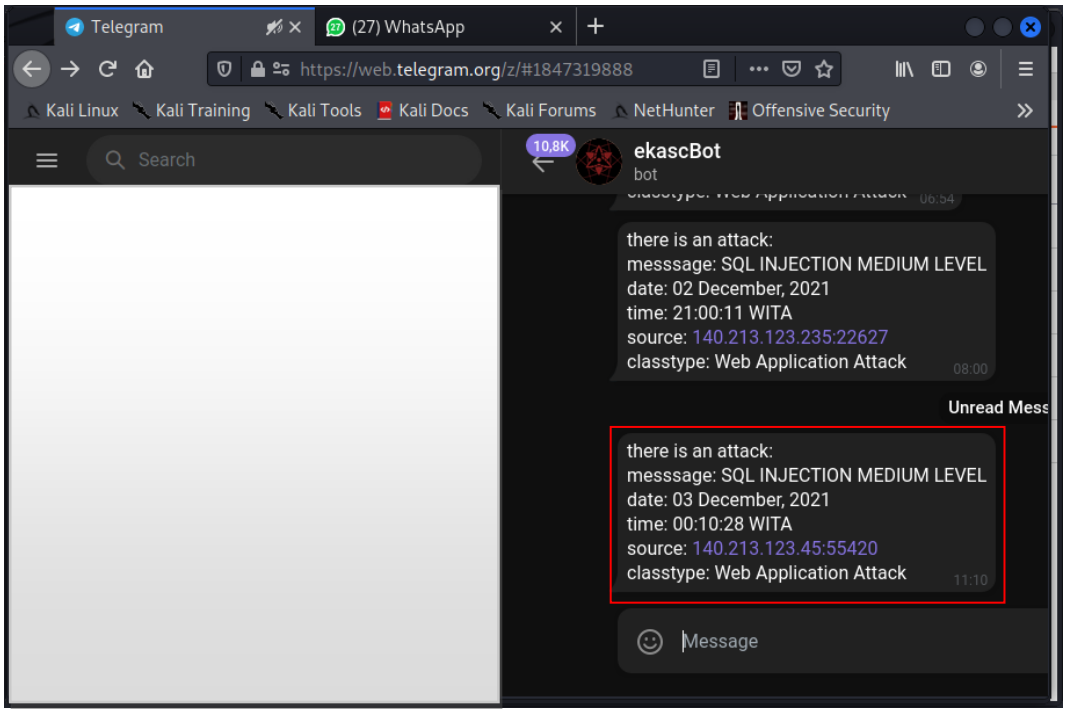


Gambar 5.15: access.log di apache2 server

Log yang terdapat pada apache2 server serta log yang terdapat pada Snort IDS tampak pada gambar 5.15 dan gambar 5.16.



Gambar 5.16: Log pada Snort IDS



Gambar 5.17: *Alert* pada Telegram

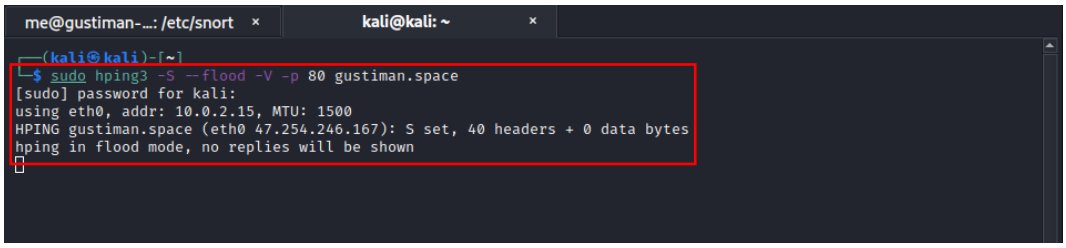
Alert pada Telegram yang mengandung informasi serangan SQL Injection Medium Level.

Tabel 5.4: Hasil pengujian *SQL Injection Medium Level* dengan *Rule Detection*

IP Adrress	Apache2 Log	Snort Log	Telegram Alert
140.213.123.45	Yes	Yes	Yes

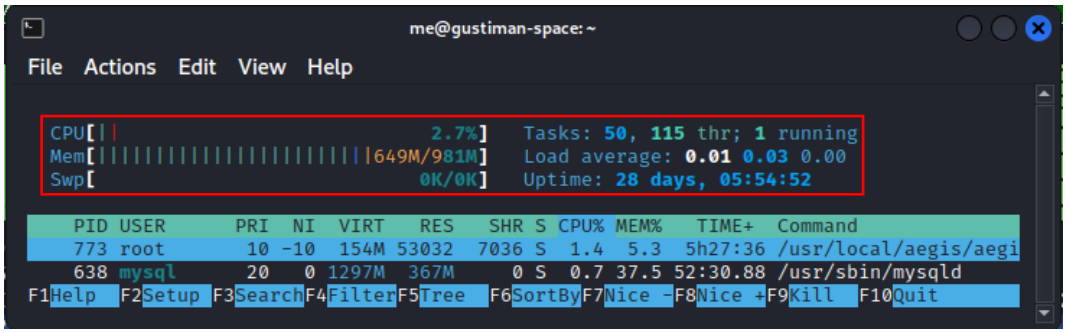
Hasil pengujian *SQL Injection Medium Level* dengan *Rule Detection* terdapat log pada Apache2, Snort, dan Telegram Alert.

5.2.5 Pengujian Denial of Service tanpa Rule Detection



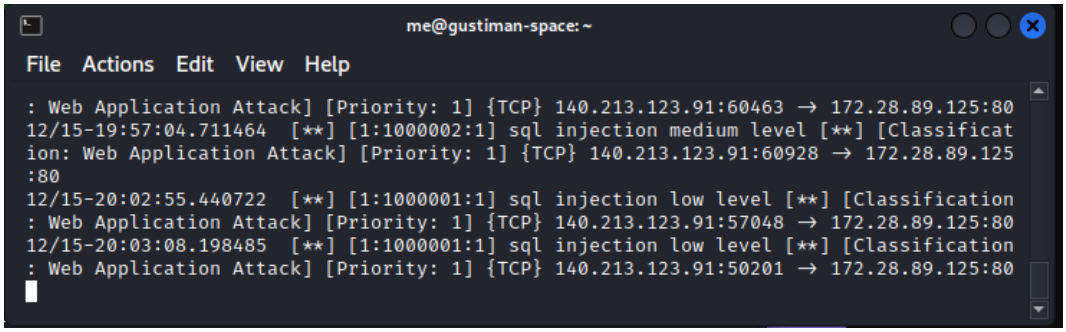
Gambar 5.18: DoS menggunakan tools hping3 pada OS Kali Linux

Pengujian DoS untuk menghabiskan resource pada server target.



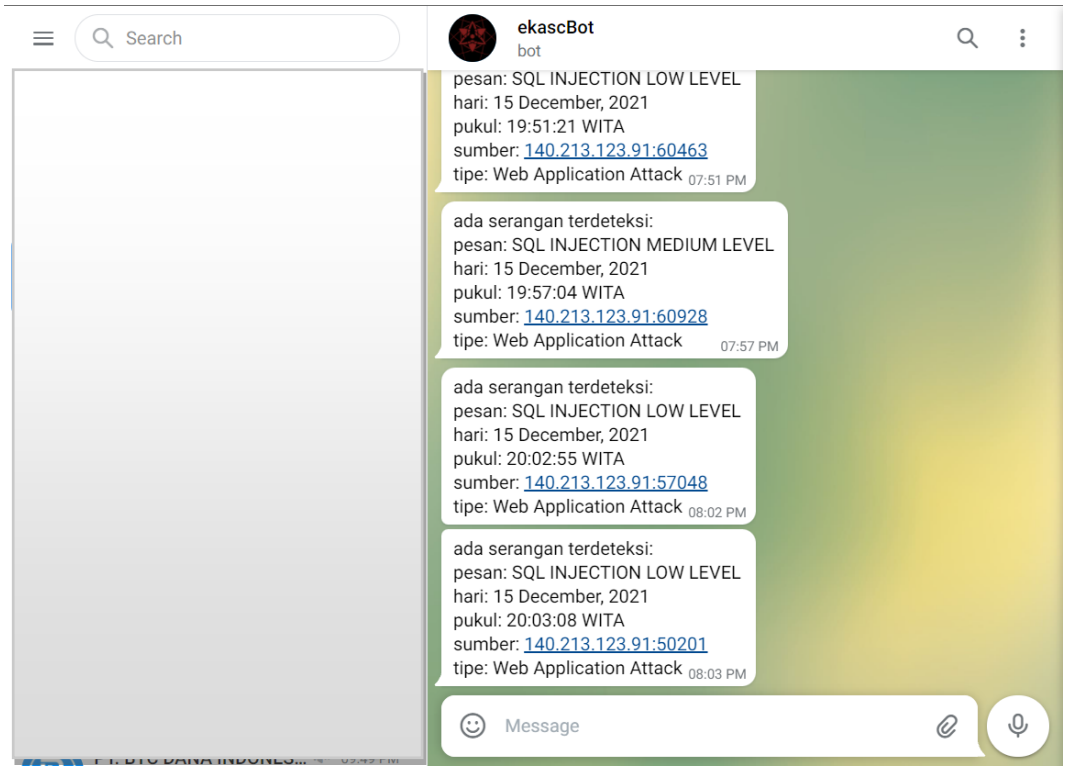
Gambar 5.19: Informasi resource pada server dengan tools htop

Ketika dilakukan pengujian sebelumnya yang tampak pada gambar 5.18, terlihat memori, CPU, dan swap normal.



Gambar 5.20: Log pada Snort IDS

Tidak terdeteksi serangan DoS pada log Snort IDS.



Gambar 5.21: Alert pada Telegram

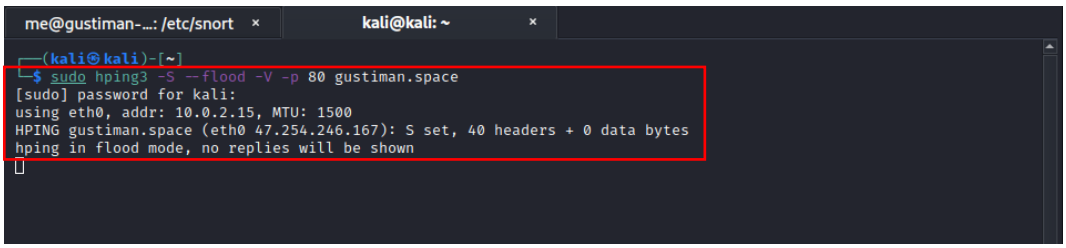
Alert pada Telegram dimana tidak terdapat notifikasi serangan DoS.

Tabel 5.5: Hasil pengujian Denial of Service tanpa Rule Detection.

IP Adrress	htop	Snort Log	Telegram Alert
140.213.123.91	Yes	No	No

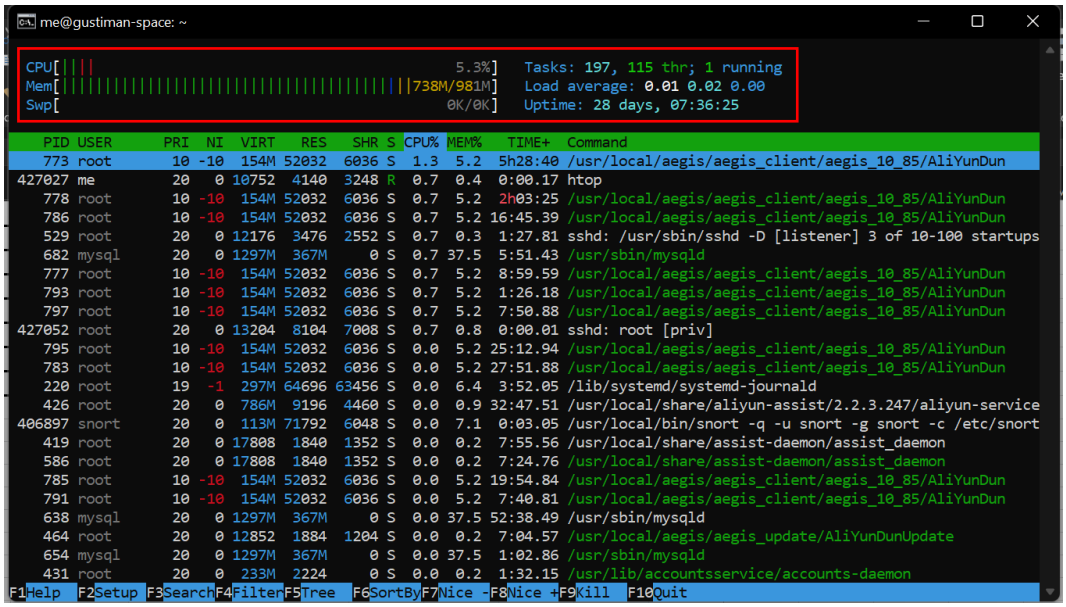
Hasil pengujian *Denial of Service* tanpa *rule detection* terlihat dengan jelas hanya terdapat pengaruh pada htop dan tidak terdeteksi pada Snort Log serta Telegram Alert.

5.2.6 Pengujian Denial of Service dengan Rule Detection



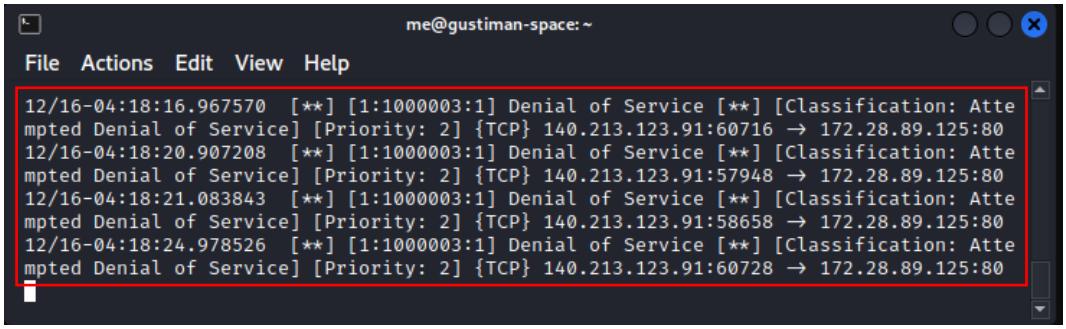
Gambar 5.22: DoS menggunakan tools hping3 pada OS Kali Linux

Pengujian DoS untuk menghabiskan resource pada server target dan membuat sever down.



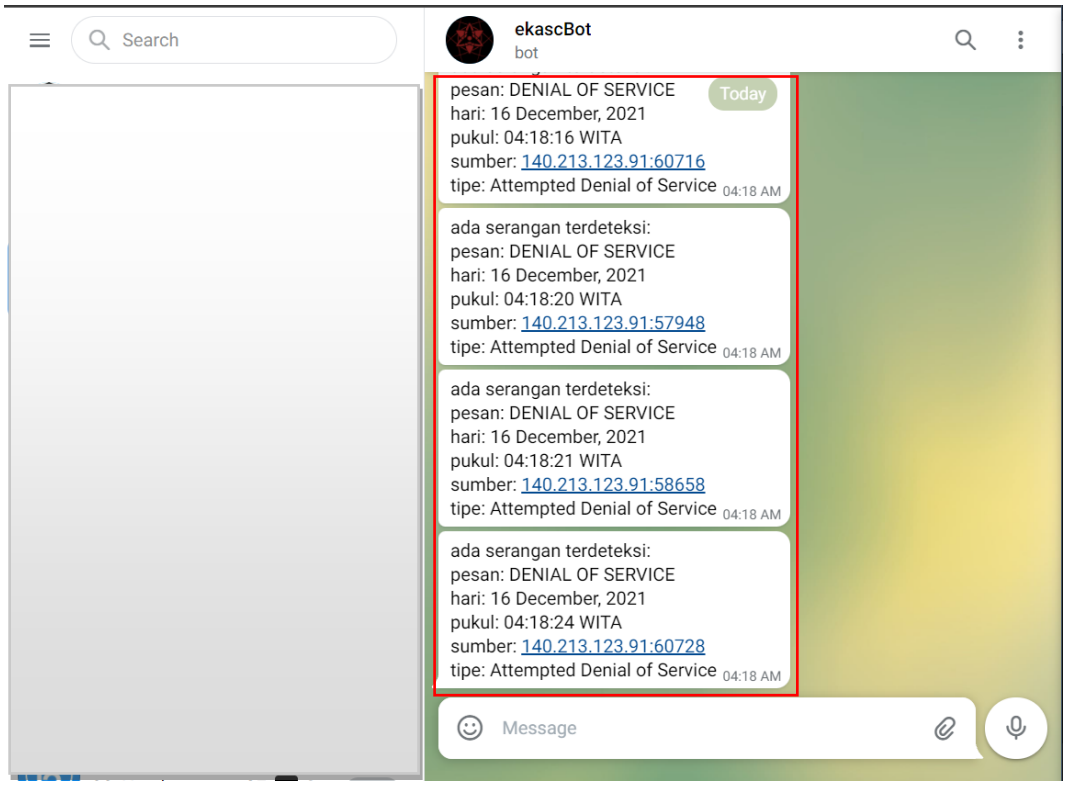
Gambar 5.23: Informasi resource pada server dengan tools htop

Ketika dilakukan pengujian sebelumnya yang tampak pada gambar 5.22, terlihat memori, CPU, dan swap mengalami kenaikan yang bisa dilihat langsung pada gambar tersebut.



Gambar 5.24: Log di Snort IDS

Pada log Snort IDS terdeteksi serangan DoS.



Gambar 5.25: Alert pada Telegram

Pengujian Denial of Service terdeteksi pada Alert Telegram.

Tabel 5.6: Hasil pengujian *Denial of Service* dengan *Rule Detection*

IP Adrress	htop	Snort Log	Telegram Alert
140.213.123.91	Yes	Yes	Yes

Hasil pengujian Denial of Service dengan rule detection dapat terdeteksi pada pemantauan resource server (htop), Snort Log, serta Telegram Alert.

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan pada situs http://gustiman.space dan pembahasan yang telah diuraikan sebelumnya, maka dapat ditarik kesimpulan bahwa:

1. Implementasi Snort IDS pada VPS untuk melakukan monitoring serangan pada jaringan komputer sudah berjalan sesuai yang diharapkan.
2. Sistem administrator dapat mencegah terjadinya penyusupan pada jaringan komputer karena saat ada serangan sistem administrator mendapatkan notifikasi tentang informasi serangan melalui Telegram.

6.2 Saran

Setelah melakukan penelitian dan perancangan snort rules pada sistus http://gustiman.space, ada beberapa saran yang perlu diperhatikan untuk mencapai tujuan yang diharapkan, yaitu sebagai berikut:

1. Snort *rules* ini dapat dikembangkan dengan melakukan percobaan pada beberapa teknik serangan web lainnya seperti XSS, CSRF, dan RFI.
2. Untuk meningkatkan performa snort maka diperlukan penyimpanan *log* yang menggunakan *database* agar lebih terstruktur dalam melihat *log* secara berkala.

DAFTAR PUSTAKA

[1] "Polri: Diduga Keras Data Kependudukan BPJS Kesehatan Bocor." https://nasional.kompas.com/read/2021/06/04/06300041/polri--diduga-keras-data-kependudukan-bpjs-kesehatan-bocor (accessed Jun. 19, 2021).

[2] S. Saiyod, Y. Chanthakoummane, N. Benjamas, N. Khamphakdee, and J. Chaichawananit, "Improving Intrusion Detection on Snort Rules for Botnet Detection," *Converg. Secur.*, vol. 2016, no. 1, pp. 19–40, 2016, doi: 10.13052/jcs2445-9992.2016.002.

[3] Wikipedia: Protection policy - Wikipedia." https://en.wikipedia.org/wiki/Wikipedia:Protection_policy#pending (accessed Sep. 03, 2021).

[4] E. K. Dewi, "Analisis Log Snort Menggunakan Network Forensic," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 2, no. 2, pp. 72–79, 2017, doi: 10.29100/jipi.v2i2.370.

[5] M. S. Sahid Aris Budiman, Catur Iswahyudi, "IMPLEMENTASI INTRUSION DETECTION SYSTEM (IDS) MENGGUNAKAN JEJARING SOSIAL SEBAGAI MEDIA NOTIFIKASI," vol. 70, no. 8. pp. 827–838, 2014.

[6] M. Dahlan *et al.*, "PENGUJIAN DAN ANALISA KEAMANAN WEBSITE TERHADAP SERANGAN SQL INJECTION," pp. 1–16, 2014.

[7] I. K. Astuti, "Jaringan Komputer," *Jar. Komput.*, 2020, doi: 10.31219/osf.io/p6ytb.

[8] "MENGENAL JENIS-JENIS JARINGAN KOMPUTER," vol. 148, pp. 148–162.

[9] E. Marpanji, "Protokol TCP/IP," *Protok. TCP/IP*, pp. 1–5, 2010, [Online]. Available: https://books.google.co.id/books?id=Q6wbyV05S3cC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.

[10] T. Ariyadi, "Mitigasi Keamanan Dynamic Host Control Protocl (DHCP) Untuk Mengurangi Serangan Pada Local Area Network (LAN)," vol. 3, no. 2, p. 147, 2018, doi: 10.35314/isi.v3i2.455.

[11] "Osi-model-jb.png (PNG Image, 349 × 449 pixels)." https://

citraweb.com/images/artikel/TCPIP/Osi-model-jb.png (accessed Apr. 28, 2021).

[12] “Mengenal Lebih Lengkap Sistem Keamanan Jaringan Nirkabel - Qwords.” https://qwords.com/blog/sistem-keamanan-jaringan-nirkabel/ (accessed Apr. 26, 2021).

[13] “aspek-aspek keamanan komputer.” https://si200.ilearning.me/2016/03/19/aspek-aspek-keamanan-komputer/ (accessed Apr. 26, 2021).

[14] “Types of Network Security Attacks | Network Security Training | EC-Council Blog.” https://blog.eccouncil.org/types-of-network-security-attacks/ (accessed Apr. 26, 2021).

[15] “Tembok api - Wikipedia bahasa Indonesia, ensiklopedia bebas.” https://id.wikipedia.org/wiki/Tembok_api (accessed Apr. 26, 2021).

[16] “Firewall adalah : Pengertian, Fungsi, Manfaat, Jenis.” https://www.seputarpengetahuan.co.id/2020/04/firewall-adalah.html#Manfaat_Firewall (accessed Apr. 26, 2021).

[17] “Mikrotik.ID : Penggunaan Custom Chain pada Firewall MikroTik.” http://www.mikrotik.co.id/artikel_lihat.php?id=146 (accessed Apr. 26, 2021).

[18] “Metode Pengembangan Perangkat Lunak NDLC - KETUTRARE.” https://www.ketutrare.com/2018/06/metode-pengembangan-perangkat-lunak-ndlc.html (accessed Apr. 26, 2021).

Lampiran 1. Kode Program

```
const { Telegraf } = require('telegraf')
const { watch, createReadStream, copyFileSync } = require('fs')
require('dotenv').config()

const chokidar = require('chokidar')
const readLastLines = require('read-last-lines')
const path = require('path')

const myPath = path.join(__dirname + '../../../../var/log/snort/alert')

const watcher = chokidar.watch(myPath, {
   persistent: true
})

const bot = new Telegraf(process.env.BOT_TOKEN)

bot.start(ctx => {
   ctx.reply(`hello there, type /help to see more commands.`)
})

bot.command('log', ctx => {
   watcher.on('ready', () => ctx.reply('log started.'))
   watcher.on('change', async path => {
      try {
         const lines = await readLastLines.read(path, 1)
         const arrMonth = [
            'January', 'February', 'March', 'April',
            'May', 'June', 'July', 'August',
            'September','October', 'November', 'December'
         ]

         const month = arrMonth[Number(lines.split('/')[0]) - 1]
         const day = lines.split('/')[1].slice(0, 2)
         const year = new Date().getFullYear()
         const time = lines.split('-')[1].slice(0,8)

         const attMessage = lines.split('[')[2].slice(13)
               const attSource = lines.split('}')[1].split('->')[0].slice(1)
         const attClassType = lines.split(':')[5].slice(1, -11)

            console.log(lines)
            ctx.reply(`
there is an attack:
```

```
messsage: ${attMessage.toUpperCase()}
date: ${day} ${month}, ${year}
time: ${time} WITA
source: ${attSource}
classtype: ${attClassType}
                `)
        } catch (err) {
            console.log(err.message)
        }
    })
})

bot.command('logStop', async ctx => {
    try {
        await watcher.close()
            .then(() => ctx.reply(`log stopped.`))
        // await watcher.unwatch(myPath)
        // ctx.reply('log stopped.')
    } catch (err) {
        console.log(err.message)
    }
})

bot.help(ctx => {
    ctx.reply(`
I can help you monitor snort logs from telegram.
You can control me by sending these commands:

# General
/start          start the bot
/help           show this help

# Snort
/log       start the SNORT NIDS mode
    `)
})

bot.launch()


{
  "name": "ekascbot",
  "version": "1.0.0",
  "description": "a simple bot for my graduation college",
  "main": "index.js",
  "scripts": {
```

```
    "start": "nodemon index.js"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/ekaxfrnd/ekascBot.git"
  },
  "keywords": [
    "node",
    "telegraf"
  ],
  "author": "Ekagustimann",
  "license": "MIT",
  "bugs": {
    "url": "https://github.com/ekaxfrnd/ekascBot/issues"
  },
  "homepage": "https://github.com/ekaxfrnd/ekascBot#readme",
  "dependencies": {
    "chokidar": "^3.5.2",
    "dotenv": "^10.0.0",
    "pm2": "^5.1.0",
    "read-last-lines": "^1.8.0",
    "telegraf": "^3.38.0"
  },
  "devDependencies": {
    "nodemon": "^2.0.9"
  }
}
```

Lampiran 2. Surat Rekomendasi Penelitian

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN GORONTALO
LEMBAGA PENELITIAN (LEMLIT)
Jln. Achmad Nadjamuddin No. 17 Kota Gorontalo, Telp: (0435) 8724466, 829975
Website: www.internal.lemlitunisan.ac.id, E-mail: lembagapenelitian@unisan.ac.id

SURAT KETERANGAN
NO : 3389/SK/LEMLIT-UNISAN/GTO/IV/2021

Yang bertanda tangan di bawah ini :

Nama	: Zulham, Ph.D
NIDN	: 0911108104
Jabatan	: Ketua Lembaga Penelitian

Menerangkan bahwa :

Nama Mahasiswa	: Ekagustiman Noho
NIM	: T3114210
Fakultas	: Fakultas Ilmu Komputer
Program Studi	: Teknik Informatika
Judul Penelitian	: ANALISIS KINERJA INTRUSION DETECTION SYSTEM (IDS) MENGGUNAKAN SNORT PADA VIRTUAL PRIVATE SERVER (VPS)

Adalah benar telah melakukan pengambilan data penelitian dalam rangka Penyusunan Proposal/Skripsi.

Gorontalo, 24 April 2021
Ketua

Zulham, Ph.D
NIDN 0911108104

Lampiran 3. Surat Rekomendasi Bebas Pustaka

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
UPT. PERPUSTAKAAN FAKULTAS
SK. MENDIKNAS RI NO. 84/D/0/2001
Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No : 023/Perpustakaan-Fikom/XI/2021

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota	:	Ekagustiman Noho
No. Induk	:	T3114210
No. Anggota	:	M202154

Terhitung mulai hari, tanggal : Selasa, 16 November 2021, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

Gorontalo, 16 November 2021
Mengetahui,
Kepala Perpustakaan

Apriyanto Alhamad , M.Kom
NIDN : 0924048601

Lampiran 4. Surat Rekomendasi Bebas Plagiasi

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO
SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI

No. 1030/UNISAN-G/S-BP/XII/2021

Yang bertanda tangan di bawah ini :

Nama	:	Sunarto Taliki, M.Kom
NIDN	:	0906058301
Unit Kerja	:	Pustikom, Universitas Ichsan Gorontalo

Dengan ini Menyatakan bahwa :

Nama Mahasisw	:	EKAGUSTIMAN NOHO
NIM	:	T3114210
Program Studi	:	Teknik Informatika (S1)
Fakultas	:	Fakultas Ilmu Komputer
Judul Skripsi	:	Analisis Kinerja Intrusion Detection System Menggunakan Snort pada Virtual Private Server

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 30%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/IX/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujiankan.

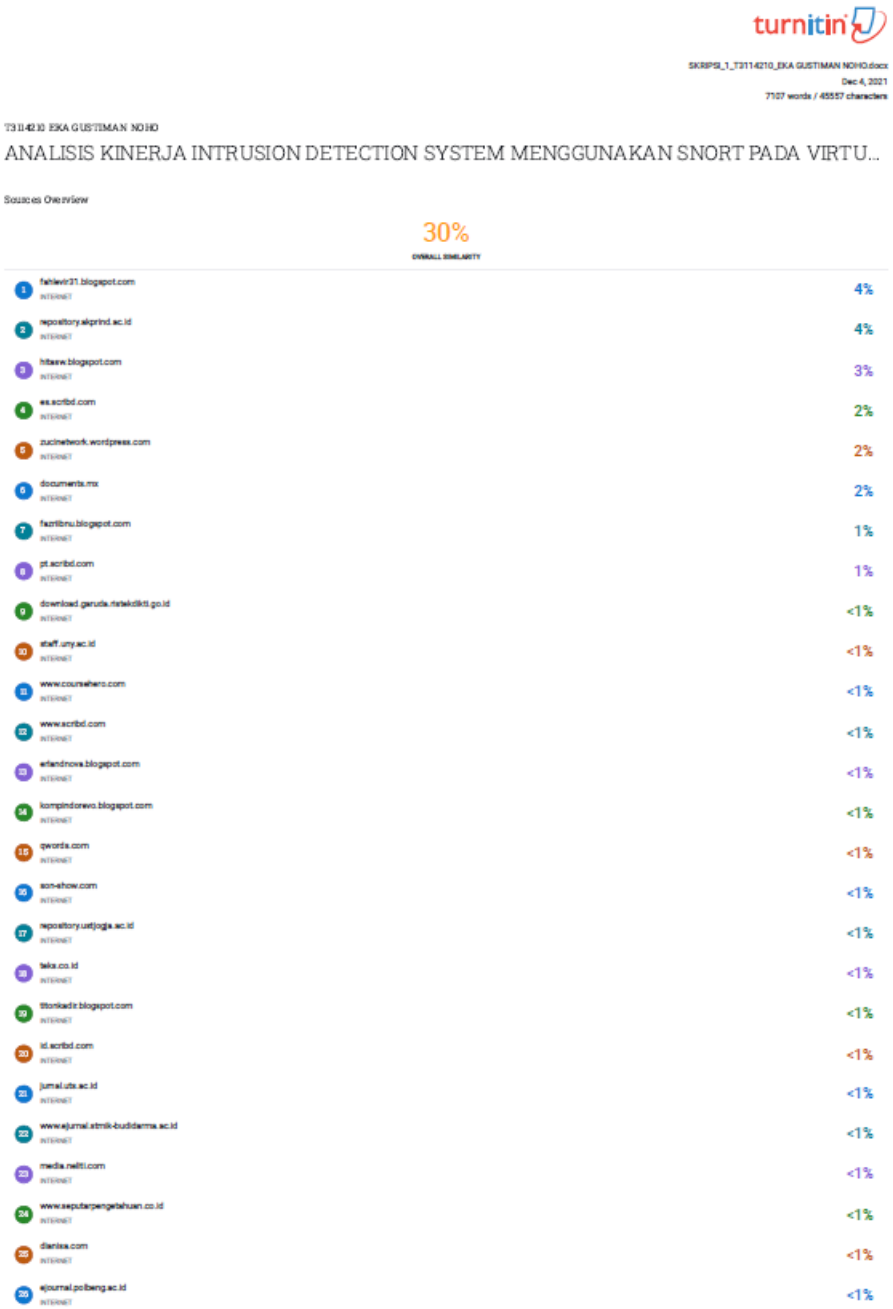
Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 07 Desember 2021
Tim Verifikasi,

Sunarto Taliki, M.Kom
NIDN. 0906058301

Tembusan :
1. Dekan
2. Ketua Program Studi
3. Pembimbing I dan Pembimbing II
4. Yang bersangkutan
5. Arsip

Lampiran 5.Hasil Uji Turnitin



turnitin

SKRIPSI_1_T3114210_EKA GUSTIMAN NOHO.docx
Dec 4, 2021
7107 words / 45557 characters

T3114210 EKA GUSTIMAN NOHO

ANALISIS KINERJA INTRUSION DETECTION SYSTEM MENGGUNAKAN SNORT PADA VIRTU...

Sources Overview

30%

OVERALL SIMILARITY

1	fahlevir31.blogspot.com INTERNET	4%
2	repository.ekprind.ac.id INTERNET	4%
3	hitaew.blogspot.com INTERNET	3%
4	es.scribd.com INTERNET	2%
5	zuclnetwork.wordpress.com INTERNET	2%
6	documents.mx INTERNET	2%
7	fazrilbnu.blogspot.com INTERNET	1%
8	pt.scribd.com INTERNET	1%
9	download.garuda.ristekdikti.go.id INTERNET	<1%
10	staff.uny.ac.id INTERNET	<1%
11	www.coursehero.com INTERNET	<1%
12	www.scribd.com INTERNET	<1%
13	erlandnova.blogspot.com INTERNET	<1%
14	kompindonewo.blogspot.com INTERNET	<1%
15	qwords.com INTERNET	<1%
16	son-ehow.com INTERNET	<1%
17	repository.uad.ac.id INTERNET	<1%
18	teka.co.id INTERNET	<1%
19	ittonkadir.blogspot.com INTERNET	<1%
20	id.scribd.com INTERNET	<1%
21	jurnal.uts.ac.id INTERNET	<1%
22	www.ejurnal.stmik-budidarma.ac.id INTERNET	<1%
23	media.neliti.com INTERNET	<1%
24	www.seputarpengetahuan.co.id INTERNET	<1%
25	dianisa.com INTERNET	<1%
26	ejournal.polteng.ac.id INTERNET	<1%

Lampiran 6. Riwayat Hidup

Nama : Ekagustiman Noho

NIM : T3114210

Tempat Tanggal Lahir : Gorontalo, 5 Agustus 1996

Agama : Islam

Email : ekaxfrnd[at]gmail.com

Riwayat Pendidikan :

1. Tahun 2008, menyelesaikan pendidikan di Sekolah Dasar Negeri 11 Tabongo, Kec. Tabongo, Kab. Gorontalo
2. Tahun 2011, menyelesaikan pendidikan di Madrasah Tsanawiyah Negeri 2 Kabupaten Gorontalo, Kec. Tabongo, Kab. Gorontalo
3. Tahun 2014, menyelesaikan pendidikan di Sekolah Menengah Kejuruan Negeri 1 Limboto jurusan Teknik Komputer Jaringan, Kec. Limboto, Kab. Gorontalo
4. Tahun 2014 diterima menjadi mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo