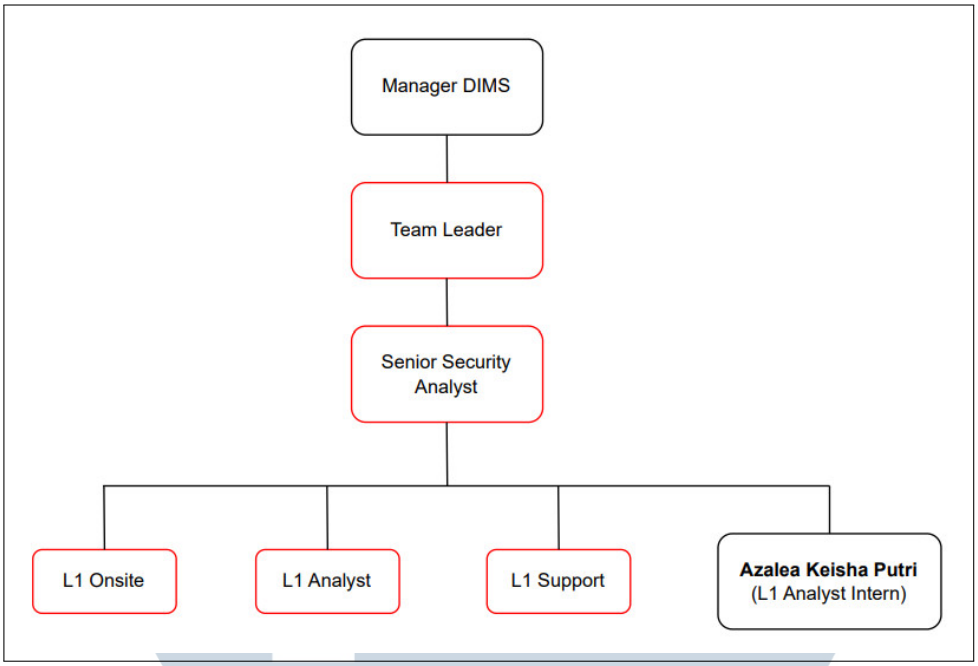
BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam struktur organisasi PT. Defender Nusa Semesta, posisi yang ditempati adalah L1 *Security Analyst Intern*, yang berada pada lapisan awal dalam tim operasional keamanan informasi. Sebagai bagian dari tim *Security Analyst*, peran ini memiliki fungsi penting sebagai garda terdepan dalam mendeteksi, memantau, dan menganalisis berbagai potensi insiden keamanan siber yang muncul dalam sistem. Fokus utama dari posisi ini adalah menjalankan proses analisis tingkat pertama (*first-level analysis*), melakukan eskalasi insiden sesuai prosedur, serta menyusun dokumentasi awal sebagai dasar bagi tim lanjutan dalam menindaklanjuti ancaman yang teridentifikasi.

Secara struktur, posisi ini berada dalam tim Level 1 (L1) bersama peran-peran operasional lainnya seperti *L1 Analyst*, *L1 Onsite*, dan *L1 Support*. Namun, sebagai L1 *Security Analyst Intern*, fokus pekerjaan sepenuhnya tertuju pada aspek keamanan informasi, yang membedakan kedudukan ini dari peran L1 lainnya. Fungsi koordinatif dan pengarahan operasional diberikan secara langsung oleh *Senior Security Analyst*, yang bertindak sebagai koordinator teknis dan pengarah utama dalam aktivitas harian tim *Security Analyst*. Setiap proses eskalasi, klarifikasi teknis, maupun pelaporan awal dilaksanakan terlebih dahulu melalui *Senior Security Analyst*, sebelum diteruskan ke *Team Leader* bila diperlukan.

Team Leader tetap memegang peran sentral dalam pengambilan keputusan dan pengawasan umum terhadap seluruh tim, namun dalam konteks operasional sehari-hari, kedudukan sebagai L1 *Security Analyst Intern* lebih intensif berinteraksi dan berkoordinasi dengan *Senior Security Analyst*. Di atas *Team Leader*, struktur organisasi dilanjutkan kepada *Manager DIMS*, yang memiliki tanggung jawab penuh atas manajemen strategis keamanan informasi digital perusahaan. Struktur kedudukan dan koordinasi ditampilkan pada Gambar 3.1 untuk memperjelas posisi dalam tim.



Gambar 3.1. Struktur kedudukan L1 Security Analyst dalam tim SOC Defenxor

Selama masa magang, koordinasi antar tim dan divisi di PT. Defender Nusa Semesta dilakukan melalui beberapa platform utama. *WhatsApp Group* digunakan sebagai media komunikasi harian untuk koordinasi internal dengan sesama anggota tim, serta komunikasi lintas divisi dalam konteks operasional. Selain itu, *Signal Group* berperan sebagai saluran komunikasi langsung dengan pihak *customer* sekaligus menjadi media pengiriman notifikasi hasil analisis keamanan yang dilakukan melalui sistem PWA (*Portal Warning Alert*). Di samping itu, *email* resmi Defenxor juga digunakan secara aktif sebagai sarana koordinasi formal, baik dalam menjadwalkan dan mengonfirmasi pertemuan (*meeting*), berkoordinasi dengan divisi lain, berkomunikasi dengan pihak CTI, maupun dalam pengiriman notifikasi hasil analisis yang lebih bersifat dokumentatif dan administratif. Dari kedudukan dan koordinasi tersebut, peran *Security Analyst* ini dijalankan secara terarah, kolaboratif, dan relevan dengan kebutuhan nyata di bidang keamanan informasi.

3.2 Tugas yang Dilakukan

Selama masa magang di PT. Defender Nusa Semesta, penempatan dilakukan di divisi *Security Operations Center* (*SOC*) dengan fokus utama pada aktivitas *monitoring* keamanan siber dan analisis insiden. Program magang berlangsung selama 25 minggu dan dibagi menjadi dua fase utama: fase orientasi dan pelatihan

teknis pada empat minggu pertama, diikuti dengan fase operasional hingga minggu ke-25. Seluruh kegiatan dilakukan secara sistematis dan dibimbing oleh mentor, dengan pendekatan kerja yang selaras dengan prinsip-prinsip pengamanan informasi berdasarkan standar *ISO/IEC 27001*. Berikut merupakan poin-poin utama aktivitas dan pembelajaran yang dilakukan selama program magang:

1. Pembelajaran Mandiri CompTIA Security+ dan Security Awareness

   Pemahaman dasar terkait keamanan informasi diperkuat melalui studi mandiri menggunakan materi *CompTIA Security+* dan *Security Awareness*. Fokus pembelajaran mencakup ancaman siber, teknik mitigasi risiko, serta prinsip dasar perlindungan data dan penanganan insiden.

2. Pengenalan Lingkungan Kerja SOC dan Simulasi Teknis

   Dilakukan pengenalan terhadap struktur organisasi, budaya kerja, serta perangkat operasional yang digunakan di *Security Operations Center* (*SOC*). Simulasi teknis dilakukan dengan menggunakan *Elastic/Kibana*, *Moloch*, serta konfigurasi akses melalui *OpenVPN*, *MobaXterm*, dan sertifikat keamanan *browser*.

3. Simulasi Kasus dan Analisis Insiden Teknis

   Mengikuti alur simulasi insiden siber untuk mempelajari proses identifikasi dan respons. Kegiatan meliputi analisis *log*, pendeteksian pola serangan seperti *brute force* atau *RDP outbound*, serta penyusunan laporan insiden berdasarkan pedoman *Playbook*.

4. Monitoring Infrastruktur dan Ketersediaan Sistem (Thruk & Log Availability

   Melakukan pemantauan terhadap infrastruktur TI menggunakan *Thruk* dan sistem *log availability*. Aktivitas berfokus pada deteksi dini terhadap gangguan layanan, kesalahan sistem, serta pembuatan notifikasi apabila ditemukan anomali.

5. Monitoring Keamanan Melalui SIEM dan Endpoint Security

   Aktivitas monitoring dilakukan menggunakan perangkat seperti *QRadar*, *Wazuh*, *Microsoft Sentinel*, serta *endpoint protection* seperti *CrowdStrike*, *SentinelOne*, dan *Kaspersky Security Center*. Fokusnya adalah korelasi *log*, identifikasi aktivitas mencurigakan, dan klasifikasi insiden secara *real-time*.

6. Penyusunan Notifikasi dan Drafting Laporan Insiden

   Setiap insiden yang ditemukan ditindaklanjuti dengan penyusunan notifikasi kepada pihak terkait serta dokumentasi kronologis ke dalam sistem internal. Termasuk pula analisis penyebab dan rekomendasi perbaikan sebagai bagian dari penguatan sistem keamanan.

Secara keseluruhan, pelaksanaan kegiatan magang ini memberikan kesempatan untuk memperoleh pengalaman langsung dalam memantau dan menangani sistem keamanan siber di lingkungan perusahaan. Proses pembelajaran dilakukan melalui pendekatan berbasis insiden dengan dukungan berbagai perangkat keamanan, serta mengacu pada kerangka kerja ISO/IEC 27001. Pengalaman ini menjadi bekal penting untuk memahami keamanan informasi dan meningkatkan keterampilan dalam mendeteksi, menganalisis, dan mencatat insiden secara sistematis. Berikut merupakan rangkuman aktivitas magang yang telah dilakukan, sebagaimana disajikan pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Mengikuti New Intern Orientation (NIO) dan onboarding perusahaan untuk memahami etika kerja, evaluasi Career Path Assessment (CPA), struktur organisasi, budaya kerja PT Defender Nusa Semesta, serta pengenalan Security Operations Center (SOC), Service Level Agreement (SLA), dan dasar-dasar keamanan siber.
2	Melakukan pembelajaran mandiri selama seminggu terkait CompTIA Security+ sebagai bekal sebelum masuk SOC, mencakup dasar keamanan informasi, jaringan, enkripsi, dan penanganan insiden, lalu mempresentasikan hasil dan berdiskusi studi kasus dengan mentor.
3	Melakukan pembelajaran mandiri selama satu minggu terkait materi CompTIA Security+ dan materi Security Awareness untuk meningkatkan kewaspadaan terhadap ancaman digital dan fisik, lalu mempresentasikan hasilnya dan berdiskusi kasus nyata bersama mentor.
4	Melakukan simulasi analisis kasus sebelum bertugas penuh sebagai Security Analyst menggunakan Elastic/Kibana, Sistem internal kantor, dan Moloch. Juga menyiapkan akses teknis dengan mengonfigurasi VPN (OpenVPN), MobaXterm, dan sertifikat keamanan browser.
5	Melakukan security monitoring dengan Thruk dan SIEM (Elastic, OpenSearch) untuk mendeteksi gangguan dan ancaman, serta membuat tiket insiden. Proses ini menggunakan berbagai kategori sistem keamanan seperti EDR/XDR, NIDS, HIDS, SOAR, firewall, WAF, email security, threat intelligence, dan cloud security.
	Lanjut pada halaman berikutnya

Tabel 3.2. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
6	Melakukan monitoring keamanan menggunakan XDR/EDR untuk menganalisis kasus "Wazuh Windows, Logon failure – Unknown user or bad password", mencakup identifikasi sumber pada index Wazuh, analisis potensi brute force atau penyalahgunaan kredensial, serta melakukan notifikasi ke customer sesuai prosedur.
7	Melakukan monitoring dengan CrowdStrike untuk menganalisis kasus "Machine Learning detected", mencakup identifikasi command line malicious, pengecekan file/path, dan notifikasi ke customer sesuai prosedur.
8	Melakukan monitoring dengan SentinelOne untuk menganalisis kasus "Administrative detected", mencakup review aktivitas administratif, histori endpoint, evaluasi akses pengguna, dan notifikasi ke customer sesuai prosedur.
9	Melakukan monitoring dengan Kaspersky Security Center untuk menangani kasus "Disinfection impossible", mencakup identifikasi penyebab, analisis file malicious, dan notifikasi ke customer sesuai prosedur.
10	Melakukan monitoring dengan Cortex XDR untuk kasus "Spyware Detected via Anti-Spyware profile", mencakup korelasi dengan Fortigate dan F5 ASM, validasi signature, serta notifikasi ke customer sesuai prosedur.
	Lanjut pada halaman berikutnya

Tabel 3.3. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

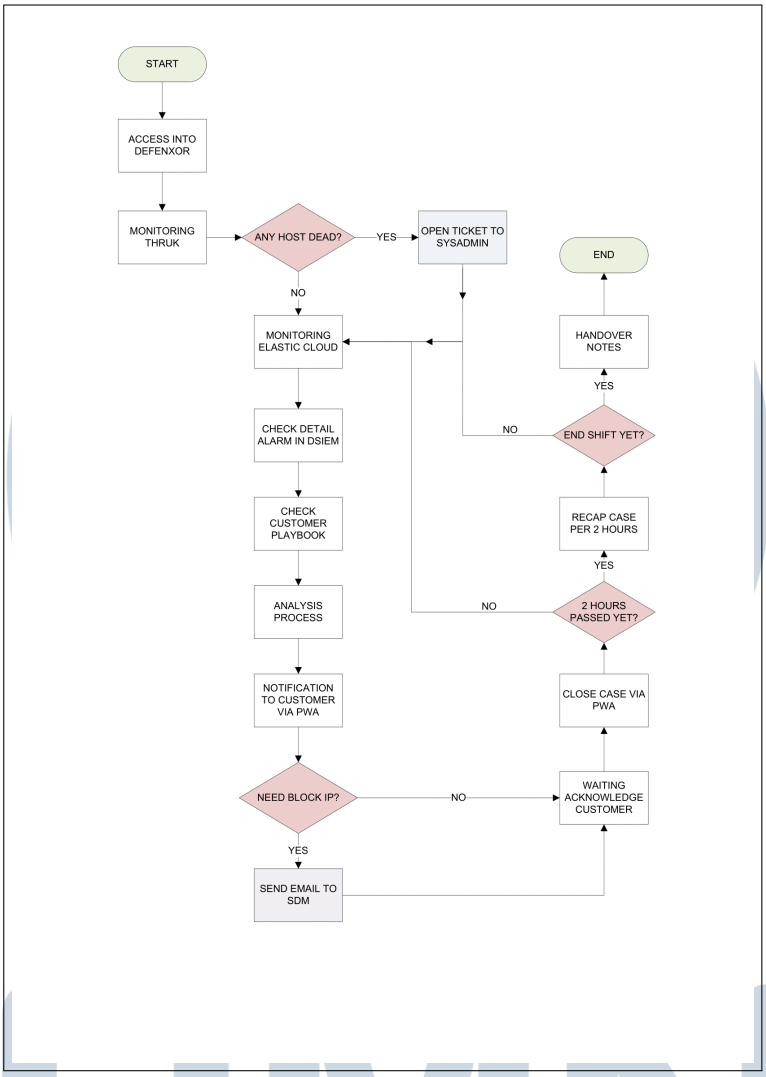
Minggu Ke -	Pekerjaan yang dilakukan
11	Melakukan monitoring dengan Trend Micro Vision One untuk kasus “Outbound RDP Connections”, mencakup pengecekan tujuan koneksi, korelasi IP reputation, identifikasi akses tidak sah, dan notifikasi ke customer sesuai prosedur.
12	Melakukan monitoring pada index SOCRadar untuk kasus “Impersonating Domain Registration Detected”, mencakup identifikasi domain tiruan, analisis potensi phishing, serta notifikasi ke customer.
13	Melakukan monitoring dengan Suricata untuk kasus “Web Attack detected”, korelasi log dengan perangkat Fortigate dan F5 ASM, identifikasi pola serangan web, pengecekan reputasi IP, dan notifikasi ke customer sesuai prosedur.
14	Melakukan monitoring dengan Trend Micro Deep Security Agent untuk kasus “Multiple Pre-authentication failed”, mencakup audit login, identifikasi akun target, dan analisis potensi brute-force.
15	Melakukan monitoring dengan F5 ASM untuk kasus “Unblocked Command Execution Detected”, mencakup korelasi dengan Palo Alto Threat, analisis log, dan pengecekan reputasi IP, dan notifikasi ke customer sesuai prosedur.
	Lanjut pada halaman berikutnya

Tabel 3.4. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
16	Melakukan monitoring pada index Akamai untuk kasus "Chrome Signature Anomaly", mencakup analisis payload malicious, notifikasi ke customer, dan blocking IP ke tim SDM.
17	Melakukan monitoring dengan Proofpoint untuk kasus "Malware Attack Detected", mencakup verifikasi hash via VirusTotal, evaluasi email health, pengecekan reputasi IP, dan notifikasi ke customer.
18	Melakukan monitoring dengan FortiMail untuk kasus "Infected email", mencakup pengecekan reputasi IP, email health, analisis attachment, dan notifikasi ke customer.

3.3 Uraian Pelaksanaan Magang

Selama enam bulan menjalani program magang di PT. Defender Nusa Semesta, mahasiswa bertugas sebagai *Security Analyst Intern* yang terlibat dalam kegiatan analisis log, pengiriman notifikasi insiden ke *customer*, penyusunan *handover notes*, serta pelaporan hasil monitoring harian. Alur kerja *Security Analyst* secara keseluruhan dapat dilihat pada gambar *flowchart* 3.2.



Gambar 3.2. SOC Monitoring Process Flow

3.3.1 Access VPN Into Defenxor

Sebelum mengakses sistem kerja internal, diperlukan proses login ke akun Linux pada perangkat kantor, diikuti dengan aktivasi VPN berbasis *OpenVPN*. VPN ini berfungsi sebagai penghubung aman ke jaringan internal perusahaan dan wajib diaktifkan terlebih dahulu untuk memastikan akses yang terenkripsi dan terlindungi dari risiko keamanan.

Sebelum digunakan secara aktif, *OpenVPN* telah melalui tahap konfigurasi mandiri pada tanggal 20 Februari, yang mencakup pengaturan profil koneksi, sertifikat, dan parameter enkripsi. Pada tanggal yang sama, dilakukan pula konfigurasi *MobaXterm* sebagai klien untuk mengakses server jarak jauh melalui protokol SSH, SFTP, dan protokol lainnya setelah jaringan VPN berhasil terhubung.

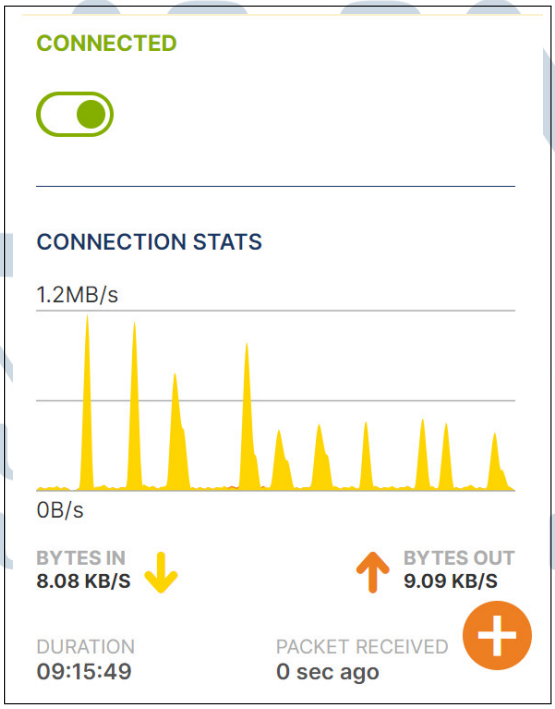
MobaXterm dipilih karena kemampuannya dalam menyediakan antarmuka terpadu untuk koneksi remote serta transfer file yang aman.

OpenVPN menggunakan enkripsi TLS (*Transport Layer Security*) untuk mengamankan koneksi antara perangkat pengguna dan server. TLS mengenkripsi lalu lintas data selama proses transmisi, sehingga mencegah potensi penyadapan dan manipulasi oleh pihak yang tidak berwenang. Selain itu, *OpenVPN* mendukung autentikasi ganda menggunakan sertifikat digital dan kredensial pengguna, guna memastikan hanya pengguna yang sah yang dapat mengakses sistem internal.

Penerapan VPN dengan skema enkripsi TLS ini merupakan bagian dari upaya penerapan standar kebijakan keamanan informasi berdasarkan ISO/IEC 27001, khususnya:

- Annex A.13.1.1 mengenai *Network Controls*, yang mengatur keamanan akses jaringan,
- Annex A.10.1 mengenai *Cryptographic Controls*, yang mencakup penggunaan teknologi enkripsi untuk melindungi data.

Selain itu, praktik ini selaras dengan pedoman teknis pada ISO/IEC 27002, yang menekankan pentingnya perlindungan data saat berada dalam jaringan, serta sejalan dengan prinsip arsitektur privasi yang diatur dalam ISO/IEC 29101. Penerapan dari proses ini dapat dilihat pada Gambar 3.3



Gambar 3.3. Aktivasi VPN Kantor Dengan OpenVpn

3.3.2 Monitoring Thruk

A. Proses Pengecekan Status Sistem Melalui Thruk

Setelah koneksi VPN kantor berhasil diaktifkan, langkah pertama yang dilakukan dalam kegiatan monitoring adalah melakukan pengecekan status sistem melalui Thruk. Thruk merupakan antarmuka web (*web interface*) yang dirancang untuk memantau dan mengelola sistem monitoring berbasis *Nagios* serta turunannya, seperti *Naemon* dan *Icinga*. Thruk menyediakan informasi status *host* dan *service* secara real-time, menampilkan notifikasi, serta menyajikan laporan ketersediaan (*availability report*) yang komprehensif. Dengan fitur-fitur tersebut, Thruk menjadi alat penting bagi *Security Analyst* dan *Sysadmin* untuk mendeteksi dan menangani gangguan sistem dengan cepat dan efisien.

Pada tampilan *dashboard Thruk*, dilakukan observasi terhadap status host dan service untuk mengidentifikasi apakah terdapat elemen yang tidak berjalan sebagaimana mestinya. Apabila suatu service mengalami gangguan, statusnya akan ditandai dengan warna merah dan label *"Critical"*, yang menandakan kondisi tidak normal. Informasi detail juga ditampilkan, meliputi:

- Waktu terakhir pengecekan (*last check*),
- Durasi terhentinya *service* hingga ke satuan hari, jam, menit, dan detik (misalnya: 20d 18h 26m 24s),
- Identitas site atau customer terkait, serta
- Status informasi tambahan, seperti apakah terdapat *record* (log) yang masuk terhadap *host/service* tersebut.

Jika ditemukan adanya host atau service yang berada dalam kondisi mati atau tidak aktif, maka langkah selanjutnya adalah melakukan proses pelaporan insiden *(ticketing)* untuk diteruskan kepada tim *System Administrator (Sysadmin)* agar dapat dilakukan penanganan lebih lanjut. Hal ini penting sebagai bagian dari alur eskalasi teknis yang telah ditetapkan.

Dalam situasi di mana Thruk tidak memberikan informasi atau mengalami malfungsi, perlu dilakukan pemeriksaan silang (*crosscheck*) terhadap sistem monitoring lainnya, seperti Kibana atau aplikasi milik customer seperti Devo. Jika hasil pemeriksaan membenarkan bahwa *service* terkait benar-benar tidak aktif,

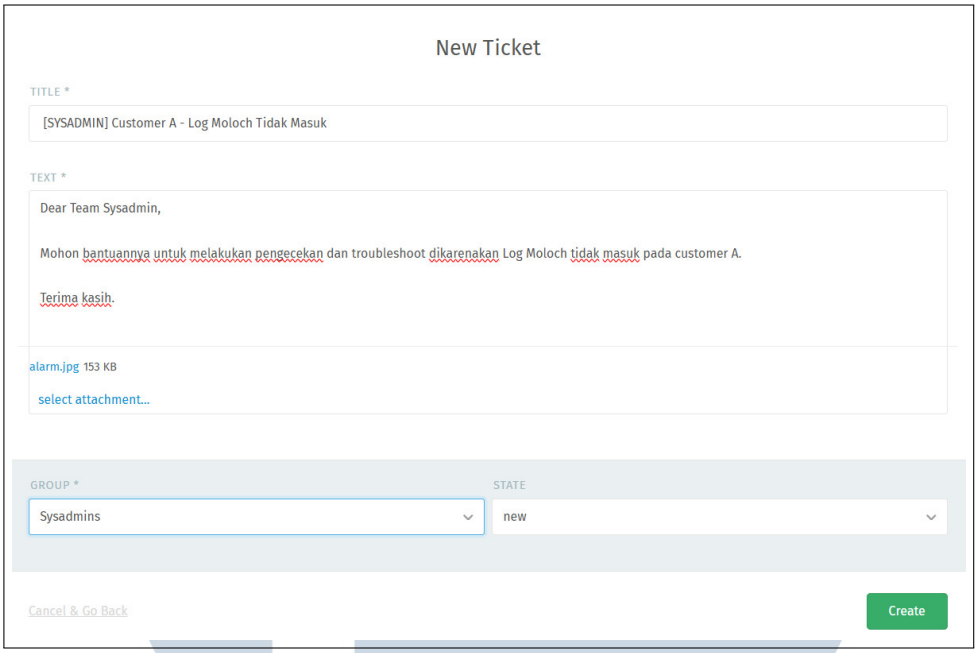
maka proses ticketing tetap perlu dilakukan dan diteruskan ke tim *Sysadmin* untuk penanganan lanjutan.

Namun, apabila tidak ditemukan indikasi masalah baik dari Thruk maupun dari hasil *crosscheck*, maka tidak perlu dilakukan proses eskalasi dan kegiatan pemantauan dapat dilanjutkan secara normal melalui *Elastic Cloud*, tanpa perlu melakukan *open ticket* atau menghubungi tim teknis.

B. Proses Ticketing Terhadap Gangguan Service atau Host

Proses *ticketing* dilakukan apabila terdeteksi adanya *service* atau *host* yang berada dalam kondisi bermasalah atau tidak aktif berdasarkan hasil pemantauan melalui sistem *Thruk*. Sistem *Ticketing* ini dikelola melalui *platform* internal *ticketing* Defenxor, yang berfungsi sebagai sistem manajemen tiket untuk menangani gangguan teknis secara terstruktur dan terdokumentasi.

Langkah awal dalam prosedur ini adalah melakukan pembuatan tiket baru pada ticketing defenxor. Pada tahap ini, perlu dicantumkan judul tiket yang secara jelas menggambarkan nama *service* atau *host* yang mengalami gangguan, disertai dengan kode identifikasi customer yang terdampak. Sebagai bukti pendukung, wajib dilampirkan tangkapan layar (*screenshot*) dari status *service* yang menunjukkan indikator "*Critical*" di *Thruk*. Kondisi ini harus terlebih dahulu diverifikasi melalui proses *crosscheck* menggunakan Kibana atau sistem pemantauan milik *customer* untuk memastikan validitas gangguan. Visualisasi contoh proses pembuatan tiket dapat dilihat pada Gambar 3.4.



Gambar 3.4. Visualisasi *Ticketing* Pada *Service* yang Mati

Setelah tiket berhasil dibuat, informasi tiket yang telah dieskalasi akan muncul secara otomatis pada dashboard "*Escalated*" di *ticketing* Defenxor. *Dashboard* ini menyajikan daftar *service* yang mengalami gangguan, nama *customer* terkait, nama *Security Analyst* yang melakukan notifikasi, status tiket terkini (misalnya: "*Open*" atau "*New*"), serta waktu dan tanggal sejak tiket tersebut pertama kali dieskalasi.

Tahap selanjutnya adalah melakukan eskalasi tiket kepada tim *System Administrator* (*Sysadmin*) untuk dilakukan investigasi teknis lebih lanjut. Selama proses penanganan berlangsung, status *service* di *Thruk* akan diperbarui menjadi "ACK" (*Acknowledged*), yang menandakan bahwa gangguan telah dikenali dan sedang dalam proses penanganan. Tim *Sysadmin* akan menganalisis lebih lanjut untuk menentukan apakah gangguan berasal dari sisi internal sistem Defenxor atau dari eksternal, yakni pada infrastruktur milik *customer*.

Apabila gangguan diketahui berasal dari sistem internal, maka tim *Sysadmin* akan segera melakukan tindakan perbaikan (*troubleshooting*) dan menginformasikan hasil pemulihan tersebut kepada *Security Analyst*, baik melalui *platform ticketing* Defenxor maupun melalui aplikasi komunikasi internal seperti *Group WhatsApp*. Sebaliknya, jika gangguan bersumber dari sisi eksternal, tim *Sysadmin* akan melakukan eskalasi lanjutan ke pihak teknis *customer* untuk ditindaklanjuti dan diperbaiki sesuai dengan prosedur di lingkungan *customer* tersebut. Seluruh rangkaian prosedur *ticketing* ini dirancang untuk memastikan

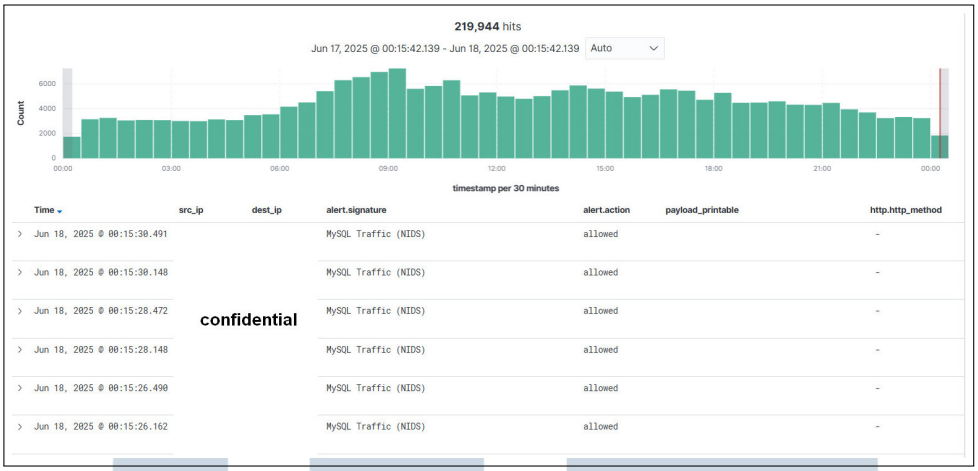
bahwa setiap masalah teknis yang terdeteksi melalui sistem pemantauan ditangani secara sistematis, dan terdokumentasi.

3.3.3 Monitoring Elastic Cloud

A. Log Collection dan Peran Sistem Keamanan

Seluruh aktivitas yang terjadi pada *endpoint*, jaringan (*network*), serta komponen infrastruktur lainnya akan terekam dan dikumpulkan oleh sistem keamanan milik *customer*. Data aktivitas ini kemudian dikirimkan dan dikonsolidasikan ke dalam *platform* SIEM (*Security Information and Event Management*) yang dikelola melalui sistem Elastic/Kibana pada lingkungan Defenxor. SIEM berfungsi sebagai pusat analisis terpusat yang memungkinkan pengelolaan log secara komprehensif untuk tujuan deteksi, korelasi, investigasi, serta respons terhadap insiden keamanan.

Sebagai contoh, ketika terjadi aktivitas web scanning dari IP asing yang mencoba mengakses file sensitif seperti .env, perangkat Fortigate mencatatnya sebagai log dengan status "*passthrough*" dan "*dropped*". Meskipun tidak diblokir sepenuhnya, informasi ini tetap dikirim ke SIEM Elastic untuk dianalisis lebih lanjut. Dalam konteks ini, log berfungsi sebagai sumber utama untuk mendeteksi pola serangan awal (*initial access*), memberikan visibilitas atas upaya akses mencurigakan, dan membantu *security analyst* menentukan apakah insiden tersebut memerlukan respons lanjutan seperti pemblokiran IP atau hardening sistem. Dengan demikian, log dalam SIEM Elastic berperan penting sebagai dasar deteksi, korelasi, dan pengambilan keputusan keamanan. Berikut visualisasi log suricata pada gambar 3.5.



Gambar 3.5. Visualisasi Log pada suricata

Selain itu, setiap organisasi umumnya memiliki arsitektur dan perangkat keamanan tersendiri yang dikonfigurasi berdasarkan kebutuhan masing-masing. Perangkat keamanan ini dapat berbeda-beda antar organisasi, namun secara umum dapat dikategorikan sebagai berikut:

Tabel 3.5. Kategori Perangkat Keamanan Pada Customer

Kategori Keamanan	Contoh Produk/Tools
EDR/XDR (Endpoint/Extended Detection and Response)	CrowdStrike, SentinelOne, Kaspersky Security Center (KSC), Cortex XDR, Trend Micro Vision One
NIDS/NDR (Network Intrusion/Detection and Response)	Suricata
HIDS/HIPS/CWPP (Host-based IDS/IPS & Cloud Workload Protection Platform)	Trend Micro Deep Security
SIEM (Security Information and Event Management)	IBM QRadar, Wazuh, Microsoft Sentinel
SOAR (Security Orchestration, Automation, and Response)	Cortex XSOAR, QRadar SOAR, Trend Micro Vision One
NGFW (Next-Generation Firewall)	FortiGate, Palo Alto, Cisco, Check Point
WAF (Web Application Firewall)	F5 ASM, Imperva, AWS WAF, Akamai
Email Security & DLP (Data Loss Prevention)	Proofpoint, FortiMail, Microsoft O365
Threat Intelligence Platform	SOCRadar
Cloud Security Tools	AWS Security Tools, Microsoft O365 Security Suite, Akamai

Perangkat-perangkat keamanan tersebut bekerja dengan pendekatan preventif dan detektif, yaitu secara aktif mencegah dan mendeteksi ancaman. Sebagian besar perangkat ini memiliki *engine* dan *database* sendiri untuk mengenali pola ancaman serta menghasilkan log yang bersifat mentah (*raw*) maupun log hasil analisis awal. Kemampuan masing-masing perangkat keamanan dapat mencakup tindakan langsung terhadap potensi ancaman, antara lain:

- ***Firewall*:** Memblokir koneksi berdasarkan *port*, alamat IP, atau protokol.
- ***EDR/XDR*:** Menghentikan proses mencurigakan (*kill process*), mengisolasi perangkat dari jaringan (*isolate host*).
- ***Antimalware*:** Melakukan karantina pada file berbahaya.
- ***IDS/IPS*:** Mendeteksi serta memblokir lalu lintas jaringan yang mencurigakan.

Namun demikian, tidak seluruh aktivitas yang terindikasi mencurigakan (*suspicious*) secara otomatis diblokir. Banyak aktivitas yang masih berada dalam *grey area*, yakni belum terbukti berbahaya (*malicious*) yang tetap dicatat sebagai log tanpa dilakukan pemblokiran langsung. Aktivitas semacam ini dikirim ke SIEM untuk dilakukan analisis lanjutan oleh *Security Analyst*, baik secara otomatis melalui *rule* korelasi maupun secara manual oleh *Security Analyst*. SIEM berfungsi sebagai lapisan deteksi strategis yang mampu melihat konteks lintas sistem, mendeteksi pola serangan lanjutan, dan menghasilkan respons yang terarah serta terdokumentasi. Pendekatan ini selaras dengan prinsip dan praktik terbaik dalam standar keamanan informasi seperti ISO/IEC 27001 – Sistem Manajemen Keamanan Informasi (ISMS), dan ISO/IEC 27002 – Pedoman Praktik Pengendalian Keamanan Informasi, khususnya dalam pengendalian akses, pencatatan peristiwa (*event logging*), dan manajemen insiden.

B. Rules dan Mekanisme Correlation Trigger dalam SIEM

Dalam sistem manajemen keamanan informasi berbasis SIEM (*Security Information and Event Management*), Elastic berperan sebagai media sentralisasi log, yang menerima dan mengelola data log dari berbagai sumber sistem keamanan dan infrastruktur TI. Setiap log yang diterima akan disimpan dalam bentuk *index* di dalam *Elastic Stack*, dan selanjutnya diproses untuk dilakukan analisis lanjutan. Proses ini menjadi bagian penting dalam mendeteksi aktivitas mencurigakan yang tidak selalu dapat diidentifikasi oleh perangkat keamanan konvensional.

Salah satu fitur utama dari SIEM adalah kemampuannya dalam melakukan korelasi antar-log (*multi-source correlation*), yaitu menggabungkan data dari berbagai sumber untuk mendeteksi pola ancaman yang lebih kompleks. Korelasi ini dilakukan dengan menyaring ulang log berdasarkan kombinasi aktivitas yang mencurigakan dari waktu dan sumber yang berbeda. Sebagai contoh, dalam kasus yang melibatkan F5 ASM, sistem dapat mencatat adanya lonjakan permintaan masif (*massive web request*) ke aplikasi web yang dilindungi, seperti akses berulang ke suatu endpoint atau payload yang tidak lazim. Meskipun tidak langsung diblokir oleh F5 dengan status ”*unblocked*”, aktivitas tersebut tetap terekam dalam log. Bila pola tersebut terjadi dari puluhan IP publik berbeda dalam waktu singkat, sistem SIEM akan menganggapnya sebagai indikasi awal dari serangan DDoS berbasis aplikasi.

Sebagai ilustrasi kasus, dapat diasumsikan bahwa terdapat permintaan

HTTP mencurigakan dari IP publik 91.232.140.197 yang mencoba mengakses file sensitif seperti /shared/.env, /local/.env, dan /docker/.env ke server internal 172.x.x.x. Permintaan ini tidak terdeteksi oleh perangkat F5 ASM kemungkinan karena lalu lintas tidak melewati jalur yang dilindungi oleh WAF tersebut, atau karena *rule signature* pada F5 belum mengantisipasi pola akses ke *path-path* tersebut. Sebaliknya, Fortigate sebagai *Next-Generation Firewall* berhasil mencatat aktivitas tersebut dengan status "*dropped*" dan "*passthrough*", menandakan bahwa koneksi berbahaya ini sempat menyentuh perimeter jaringan internal. Melalui integrasi log dari Fortigate ke dalam SIEM Elastic, rule deteksi yang telah dikonfigurasi secara khusus akan mengenali pola-pola permintaan terhadap file konfigurasi yang tidak seharusnya diakses. Rule ini akan memicu *alert* karena mendeteksi adanya distribusi serangan yang tidak wajar terhadap sumber daya sensitif. Inilah yang terjadi pada rule bernama "*NIDS Web Attack Detected*" Berikut merupakan contoh rules dari *NIDS Web Attack Detected* pada gambar 3.6.

```
event.module: "suricata" and
event.kind: "alert" and
(alert.signature: "*web attack*" or
 alert.signature: "*.env" or
 alert.signature: "*http*" or
 rule.category: "web-application-attack")
```

Gambar 3.6. Visualisasi *Logic pada Rules Elastic*

Rule ini menggunakan *query* berbasis KQL (*Kibana Query Language*) dalam Elastic yang menyaring log dari modul seperti suricata atau fortigate selama 10 menit terakhir, dengan jadwal eksekusi setiap 5 menit. Query ini memfokuskan pencarian pada *event* yang mengandung *signature* terkait *web attack*, seperti akses mencurigakan ke *file* sensitif .env atau eksploitasi berbasis HTTP. Rule akan mengekstrak data berdasarkan *alert.signature* atau *rule.category* seperti "*web-application-attack*" dan mencatat kejadian yang memiliki nilai *event.kind: alert*. Jika ditemukan aktivitas dari IP penyerang seperti 91.232.140.197 menuju host internal maka *alert* akan dibuat untuk memberi peringatan dini. Untuk mencegah *alert fatigue*, aturan ini dapat dikombinasikan dengan mekanisme *suppression* per

kombinasi IP sumber dan target selama periode tertentu.

Proses korelasi ini tidak berdiri sendiri, melainkan dikendalikan oleh *rules* yang dirancang secara khusus oleh analis keamanan tingkat lanjut, dalam hal ini biasanya oleh *Senior Security Analyst (SSA)* Level 2. *Rules* ini ditulis dalam platform SIEM seperti Elastic/Kibana, dan menjadi logika utama untuk mendeteksi potensi ancaman. Berbeda dengan perangkat keamanan seperti *firewall*, EDR, atau IDS yang mendeteksi ancaman berdasarkan *engine* dan *signature database*, rules dalam SIEM bersifat lebih logis dan kontekstual. Aturan-aturan ini disusun berdasarkan pendekatan analitis, seperti korelasi waktu, *threshold* aktivitas, pengecualian IP internal, serta pemanfaatan data *threat intelligence*. Aturan-aturan tersebut dirancang berdasarkan berbagai pendekatan, antara lain:

- Logika hubungan antar-event (*event correlation logic*),
- Frekuensi dan interval waktu (*time-based or frequency analysis*),
- Pola perilaku (*behavioral analysis*),
- Kondisi umum yang sensitif terhadap aktivitas mencurigakan (*generalized suspicious pattern detection*).

Dengan pendekatan ini, SIEM dapat mendeteksi aktivitas yang tidak terklasifikasi sebagai serangan eksplisit (*malicious*), tetapi tetap patut dicurigai (*suspicious*) karena menunjukkan pola yang tidak biasa atau berpotensi menjadi bagian dari serangan bertahap (*multi-stage attack*).

Log yang sebelumnya mungkin lolos dari deteksi awal oleh perangkat keamanan lokal akan kembali diperiksa oleh Elastic melalui *rules* yang telah ditentukan. Apabila suatu log memenuhi kondisi dari *rule* yang ada, maka *rule* tersebut akan terpicu (*triggered*) dan log akan ditandai sebagai aktivitas mencurigakan. Setiap kejadian yang terdeteksi akan dikelompokkan berdasarkan tingkat keparahan (*severity level*), yang umumnya diklasifikasikan ke dalam kategori *low*, *medium*, atau *high*, tergantung pada tingkat risiko dan dampaknya terhadap sistem.

C. Proses Terbentuknya Alert dari Hasil Trigger Rules

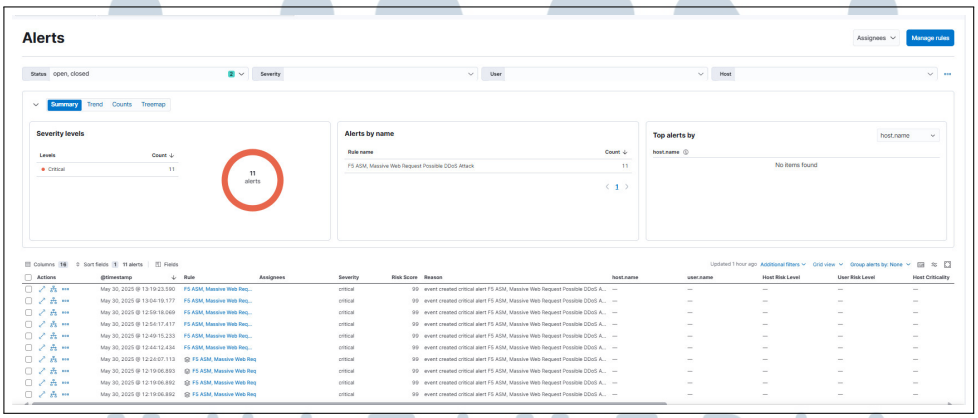
Setelah proses korelasi log dijalankan dan sebuah log memenuhi kriteria dari salah satu *rules* yang telah ditetapkan dalam sistem SIEM (seperti Elastic/Kibana),

maka sistem akan menandai log tersebut sebagai *matching event*. Log yang telah teridentifikasi ini kemudian akan naik tingkat menjadi sebuah *alert*.

Alert merupakan sinyal otomatis yang dihasilkan oleh sistem sebagai indikator bahwa telah terjadi suatu aktivitas mencurigakan atau anomali yang perlu mendapat perhatian lebih lanjut. Proses ini menandai transisi dari log biasa yang sebelumnya bersifat pasif, menjadi entitas aktif yang membutuhkan analisis lanjutan dan, jika diperlukan, tindakan respons. Proses pembentukan *alert* ini dimulai dari:

- Penerimaan log dari berbagai sumber, seperti perangkat *endpoint*, jaringan, *firewall*, *VPN gateway*, sistem *cloud*, dan lainnya,
- Penyimpanan log ke dalam *index* Elastic,
- Penerapan *rules* korelasi oleh SIEM, yang mencocokkan pola aktivitas dalam log dengan skenario yang dianggap mencurigakan atau berisiko,
- Pemicu *rules* (*trigger*) apabila ditemukan kesesuaian,
- Dan akhirnya, pembuatan *alert* berdasarkan *output rules* yang terpenuhi.

Visualisasi dari *alert* yang dihasilkan setelah pemenuhan rule dapat dilihat pada Gambar 3.7.



Gambar 3.7. Visualisasi Alerts pada Elastic

Setiap *alert* yang terbentuk biasanya dilengkapi dengan informasi penting seperti:

- Waktu kejadian (*timestamp*),
- Jenis aktivitas atau event yang terdeteksi,

- Sumber dan tujuan log (IP, *hostname, port,* dsb.),
- Tingkat keparahan (*severity level*), seperti *low*, *medium*, atau *high*,
- *Rule ID* atau nama *rule* yang menyebabkan *alert*, serta
- Rangkuman data yang dikaitkan dalam korelasi.

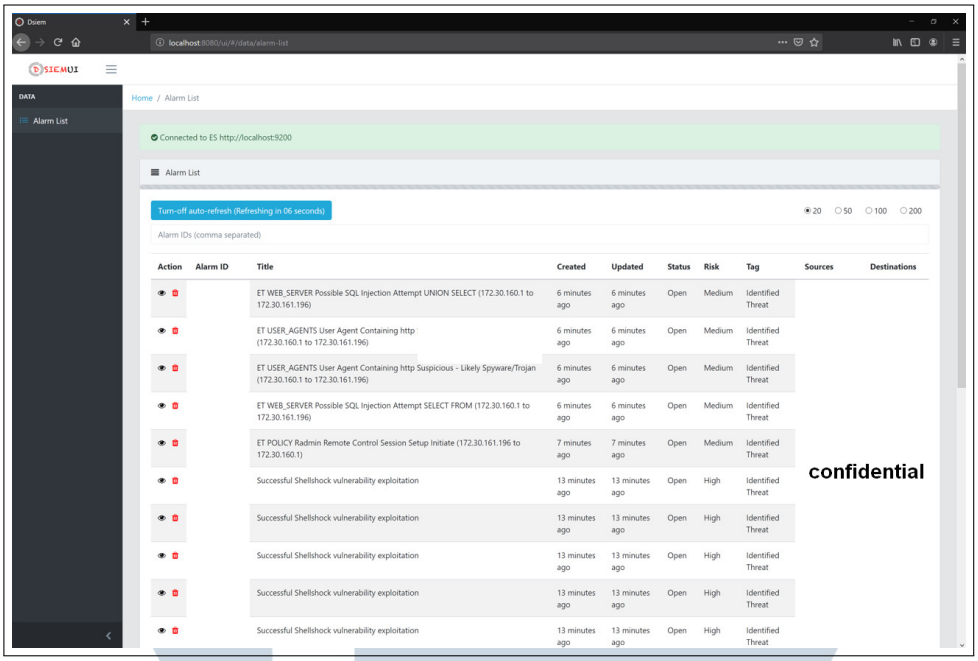
Alert ini akan ditampilkan secara real-time pada dashboard pemantauan di Kibana, dan dapat disusun dalam bentuk daftar prioritas berdasarkan tingkat keparahan dan urgensi. Alert berfungsi sebagai dasar bagi analis keamanan informasi untuk melakukan investigasi awal, validasi insiden, dan penentuan langkah selanjutnya, seperti eskalasi, isolasi sistem, atau pencatatan sebagai false positive.

Dengan sistem *alert* yang terotomatisasi, ini dapat meningkatkan efisiensi dalam mendeteksi dan merespons ancaman siber. Selain itu, proses ini mendukung prinsip deteksi dini *(early detection)* yang merupakan bagian krusial dalam manajemen insiden keamanan informasi.

D. Proses Eskalasi Alert Menjadi Alarm

Dalam sistem pemantauan keamanan berbasis SIEM (*Security Information and Event Management*), setiap log yang memenuhi kondisi pada rule tertentu akan menghasilkan sebuah *alert*. Namun, tidak semua *alert* memiliki dampak yang signifikan secara individual. Oleh karena itu, sistem dirancang untuk melakukan *escalation* atau peningkatan level peringatan apabila ditemukan konsistensi atau volume *alert* yang tinggi dalam periode waktu tertentu, khususnya yang memiliki tingkat risiko tinggi.

Apabila sejumlah *alert* yang dihasilkan memiliki korelasi yang kuat, berasal dari sumber kritikal, atau menunjukkan pola serangan berulang dalam jangka waktu yang singkat, maka SIEM akan secara otomatis atau semi-otomatis menaikkan statusnya menjadi sebuah *alarm*. *Alarm* merupakan peringatan sistem tingkat lanjut yang menunjukkan adanya potensi insiden serius atau ancaman berskala besar terhadap sistem informasi organisasi. Visualisasi *alarm* yang dikelola melalui sistem internal DSIEM dapat dilihat pada Gambar 3.8 berikut.



Gambar 3.8. Visualisasi Alarm pada Elastic

Alarm list pada DSIEM ini berfungsi sebagai antarmuka untuk memudahkan *security analyst* dalam melakukan manajemen alarm. Data alarm yang tampil di DSIEM sebenarnya berasal dari *alert* yang terdeteksi oleh SIEM utama seperti Kibana atau OpenSearch. DSIEM menyajikannya kembali dalam format yang lebih terstruktur, termasuk informasi seperti judul alarm, sumber IP, status (open/closed), tingkat risiko, dan tag identifikasi ancaman.

Alarm pada DSIEM dapat naik secara otomatis menjadi *case* ketika status-nya berubah dari *open* menjadi *closed*, dan tag dari *Identified Threat* menjadi *Valid Threat*. Namun, tidak semua alarm langsung otomatis diproses beberapa memerlukan analisis manual terlebih dahulu oleh *security analyst* melalui *dashboard* Kibana/OpenSearch. Jika setelah dianalisis alarm tersebut merupakan ancaman yang valid (*valid threat*), maka status-nya dapat diubah menjadi *closed* dan *tag* diperbarui menjadi *Valid Threat*. Setelah itu, *security analyst* dapat membuat *case* secara manual ke dalam sistem PWA sebagai langkah tindak lanjut investigasi insiden. Proses peningkatan dari *alert* menjadi *alarm* biasanya bergantung pada beberapa faktor berikut:

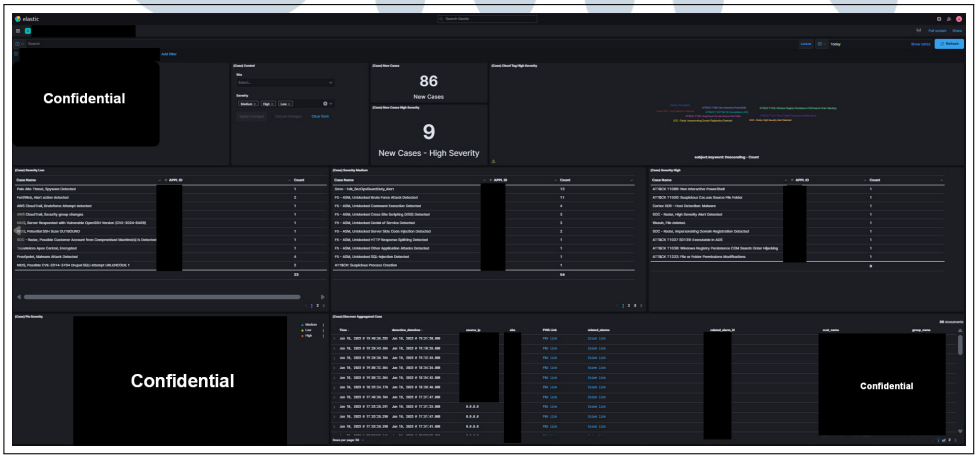
- Frekuensi *alert* dalam waktu tertentu (misalnya, lebih dari 10 *alert* dengan *rule* serupa dalam 5 menit),
- Tingkat keparahan (*severity*) dari *alert* yang muncul (terutama kategori high),

- Sumber sistem yang terdampak, misalnya berasal dari sistem kritis seperti server otentikasi, *gateway VPN*, atau database utama,
- Jenis aktivitas yang terdeteksi, seperti eksploitasi kerentanan, *brute force login, lateral movement*, *data exfiltration*, dan sebagainya.

E. Case Management pada Elastic Cloud

Dalam lingkungan keamanan siber modern, Elastic Cloud berperan sebagai pusat pengelolaan insiden melalui sistem *case management* yang terintegrasi dengan *alarm* dan *alert* yang dihasilkan oleh berbagai perangkat keamanan. *Alarm-alarm* tersebut dapat naik menjadi *case* investigasi melalui dua mekanisme utama, yaitu secara otomatis berdasarkan *rule* dan korelasi log yang telah ditentukan oleh sistem deteksi, atau secara manual atas *request* dari *Senior Security Analyst (SSA)* maupun dari pihak *customer* yang melaporkan aktivitas mencurigakan. Elastic Cloud berfungsi sebagai *platform* sentralisasi untuk seluruh kasus keamanan yang diklasifikasikan berdasarkan tingkat keparahan (*severity*), seperti *low, medium, high*, hingga *critical*. Klasifikasi ini mempermudah tim keamanan dalam memprioritaskan penanganan insiden secara efisien dan terstruktur.

Elastic Cloud sendiri merupakan layanan *cloud* terkelola dari Elastic yang menyediakan akses ke berbagai komponen utama seperti Elasticsearch, Kibana, dan tools lainnya yang dirancang untuk pencarian, analisis, serta visualisasi data dalam skala besar. Melalui *platform* ini, *security analyst* dapat memantau log, mendeteksi aktivitas mencurigakan, serta melakukan investigasi insiden tanpa harus mengelola infrastruktur secara manual. Berikut merupakan visualisasi dari *Elastic Cloud* sebagai *management case* pada gambar 3.9.



Gambar 3.9. Visualisasi Case pada Elastic

Dalam praktiknya, *Security Analyst* biasanya melakukan investigasi dalam interval waktu tertentu, seperti setiap dua jam, misalnya dari pukul 10.00 hingga 12.00, untuk memastikan bahwa seluruh potensi insiden dapat terpantau secara berkala dan menyeluruh. Selain itu, pada tampilan antarmuka Elastic Cloud, alarm yang terdeteksi akan disertai dengan informasi penting seperti *subject alarm*, alamat IP sumber aktivitas mencurigakan *(IP source*), serta tautan ke sistem internal perusahaan seperti DSIEM dan PWA. DSIEM merupakan sistem internal Defenxor yang digunakan untuk mengelola alarm, dan melihat detail teknis, sementara PWA berfungsi sebagai sistem untuk menyusun draft hasil analisis disertai dengan bukti (*evidence*) dari temuan yang diperoleh, dan digunakan dalam proses notifikasi kepada pihak *customer*. Dengan fungsi-fungsi tersebut, *Elastic Cloud* tidak hanya berperan sebagai alat deteksi dan pemantauan, tetapi juga menjadi komponen utama dalam koordinasi investigasi serta komunikasi insiden keamanan siber secara *end-to-end.*

F. Service Level Agreement

Dalam sistem manajemen insiden yang diimplementasikan pada *Elastic Cloud*, setiap *case* yang terdeteksi dan diklasifikasikan akan dibedakan berdasarkan tingkat keparahan atau *severity*, yang terdiri dari tiga kategori utama: *low, medium,* dan *high*. Masing-masing kategori tersebut memiliki *Service Level Agreement* (SLA) yang berbeda, yang menentukan batas waktu maksimal respons awal terhadap sebuah insiden. Untuk kasus dengan severity *low*, SLA yang diterapkan adalah selama 60 menit, sementara kasus dengan severity *medium* memiliki SLA 30 menit, dan untuk kasus dengan severity *high*, SLA yang ditetapkan adalah hanya 15 menit. Penetapan SLA ini bertujuan untuk memastikan bahwa setiap insiden ditangani secara cepat dan proporsional berdasarkan tingkat risiko yang ditimbulkan terhadap sistem atau data perusahaan. Sebagai bagian dari prosedur standar operasional, seluruh *security analyst* diwajibkan untuk memberikan prioritas penanganan tertinggi terhadap *case* dengan *severity high*, mengingat potensi dampak kritis yang dapat ditimbulkan apabila tidak segera direspon.

3.3.4 Check Detail Alarm in DSIEM

A. DSIEM Sebagai Case Management

Dalam proses investigasi insiden melalui Elastic Cloud, terdapat dua tautan penting yang mengarah ke sistem internal perusahaan yang berfungsi sebagai sarana manajemen kasus dan penyampaian notifikasi analisis kepada *customer*. Salah satu tahapan awal yang dilakukan oleh *security analyst* saat mengakses Elastic Cloud adalah membuka dan memeriksa detail dari sebuah alarm melalui sistem internal bernama DSIEM. DSIEM merupakan *platform* manajemen kasus yang digunakan oleh tim *security analyst* untuk menangani alarm secara terstruktur dan terdokumentasi. Di dalam sistem ini, ditampilkan daftar alarm yang telah terdeteksi, lengkap dengan berbagai informasi penting terkait masing-masing *alarm*. Berikut merupakan visualisasi dari DSIEM pada gambar 3.10.

Gambar 3.10. Visualisasi Case pada Elastic

Informasi yang disajikan dalam DSIEM mencakup rincian *alarm* secara mendalam, termasuk *timestamp* atau waktu kejadian, status penanganan saat ini, serta tingkat keparahan atau *severity* dari insiden tersebut. Selain itu, sistem juga menampilkan detail aturan (*rule details*), dan tahapan korelasi (*correlation stage*), serta sumber log atau *event detail* yang menunjukkan dari index atau sistem mana alarm tersebut berasal. Melalui tampilan informasi yang komprehensif ini, *security analyst* dapat memperoleh gambaran awal mengenai insiden dan menentukan langkah selanjutnya yang harus diambil, baik untuk eskalasi, mitigasi, maupun pembuatan notifikasi resmi kepada pihak yang terkait. Keberadaan DSIEM sebagai penghubung utama antara Elastic Cloud dan sistem internal perusahaan sangat penting untuk memastikan bahwa setiap alarm yang muncul dapat dikelola secara efisien, terdokumentasi, dan terintegrasi ke dalam proses keamanan organisasi secara keseluruhan.

B. Melakukan Pengecekan pada Index DSIEM

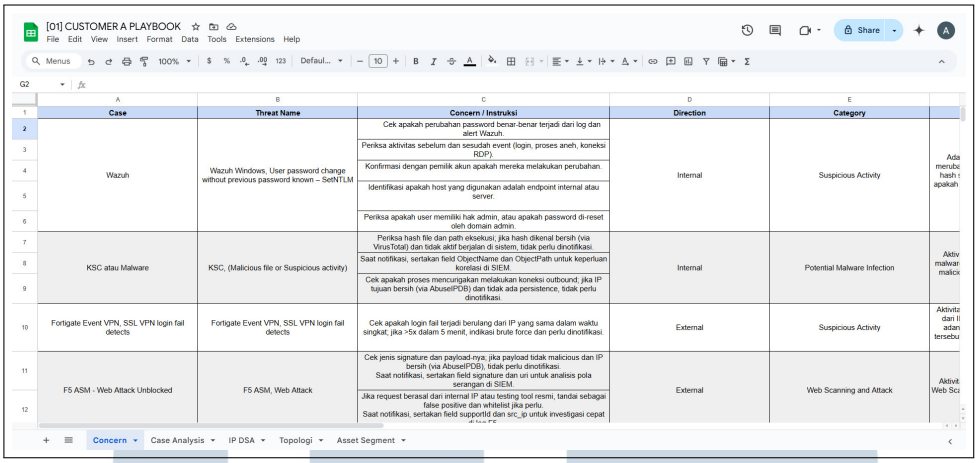
Langkah awal dalam proses investigasi keamanan adalah melakukan pengecekan terhadap *index alarm* pada sistem internal DSIEM. Setelah *security*

analyst membuka detail kasus yang muncul di SIEM, *security analyst* harus mengidentifikasi dari index mana alarm tersebut berasal. Informasi ini dapat ditemukan pada kolom *event details* yang terdapat dalam tampilan detail kasus di DSIEM. Mengetahui asal *index* ini sangat penting karena akan menjadi titik awal untuk proses pencarian dan analisis log lebih lanjut di platform seperti Kibana atau OpenSearch. Dengan mengacu pada *index* tersebut, *security analyst* dapat menelusuri log terkait yang menjadi pemicu alarm, sehingga arah investigasi menjadi lebih tepat sasaran dan efisien.

Dalam praktiknya, jenis *index* yang digunakan akan sangat bergantung pada jenis kasus yang sedang ditangani. Untuk kasus seperti *"NIDS Web Attack Detected"*, investigasi dilakukan pada *index* seperti logs-suricata* atau logs-fortigate* karena log yang dianalisis berasal dari perangkat *Network Intrusion Detection System (NIDS)* seperti Suricata dan perangkat *Next-Generation Firewall* seperti Fortigate. Suricata berperan mendeteksi pola serangan berbasis signature pada lalu lintas jaringan, sedangkan Fortigate mencatat aksi yang terjadi di level perimeter, seperti koneksi yang dibatalkan (*dropped*) atau dibiarkan lewat (*passthrough*). Sementara itu, untuk kasus-kasus yang berkaitan dengan framework MITRE ATT&CK, biasanya investigasi diarahkan pada *index* dari sistem HIDS (*Host Intrusion Detection System*), seperti *wazuh-alerts*. *Index* ini mencakup log dan peringatan dari *endpoint* atau agen keamanan di sisi host. Oleh karena itu, memilih index yang sesuai dengan konteks ancaman merupakan aspek krusial dalam keberhasilan proses investigasi.

3.3.5 Check Customer Playbook

Sebelum melakukan analisis lebih lanjut di Kibana atau Elastic, *Security Analyst* perlu terlebih dahulu melakukan pengecekan terhadap *playbook* milik customer. *Playbook* ini berfungsi sebagai panduan dan referensi teknis yang berisi ketentuan atau kebijakan yang berlaku untuk masing-masing *customer*. Playbook penting karena menjadi acuan dalam investigasi agar analyst memahami batasan, kebijakan, dan concern teknis dari customer. Visualisasi pada tampilan playbook dapat dilihat pada Gambar 3.11.



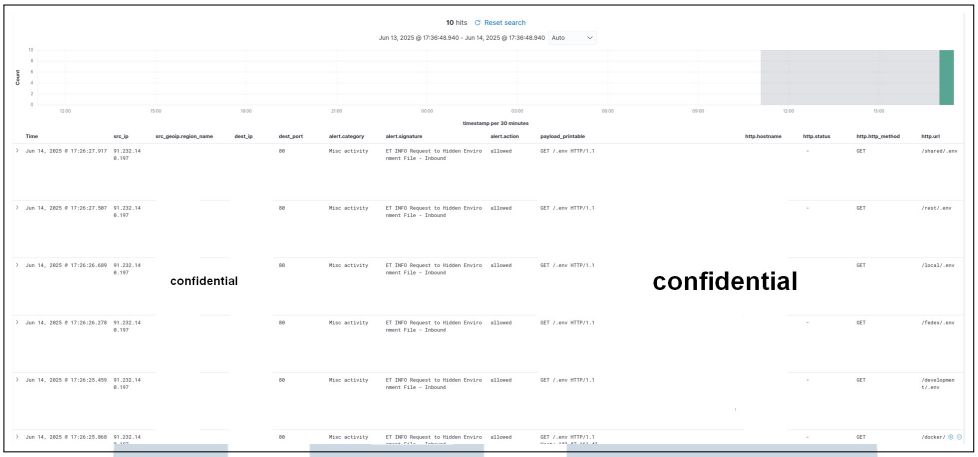
Gambar 3.11. Visualisasi Playbook Customer

Beberapa hal yang biasa diperiksa dalam *playbook* antara lain adalah adanya *concern* atau perhatian khusus dari pelanggan terhadap jenis ancaman tertentu, batas minimum atau maksimum jumlah log (*hits*) yang harus dipenuhi sebelum kasus dinotifikasi, serta daftar IP atau *domain* yang telah masuk *whitelist*. Dengan memahami isi *playbook* terlebih dahulu, *Security Analyst* dapat melakukan analisis yang tidak hanya akurat secara teknis, tetapi juga selaras dengan kebutuhan spesifik dari pelanggan. Langkah ini juga meminimalisir kemungkinan terjadinya *false positive* atau eskalasi yang tidak diperlukan.

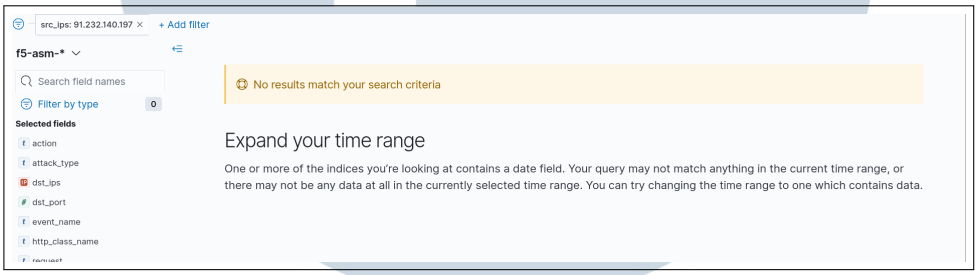
3.3.6 Analysis Process

A. Melakukan Analisis pada Discover Kibana/Elastic

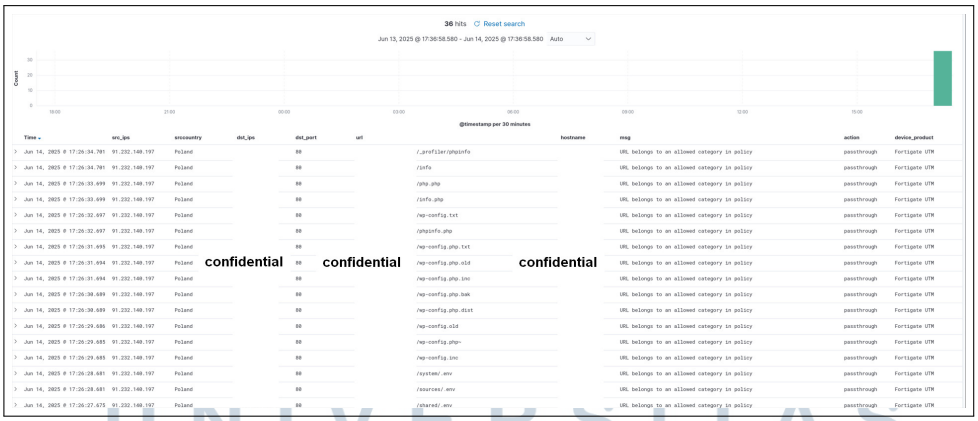
Setelah proses validasi terhadap *playbook* selesai dilakukan, langkah selanjutnya adalah melakukan analisis data melalui menu *Discover* di Kibana atau antarmuka pencarian log lainnya di Elastic. Di sini, *Security Analyst* menyusun filter berdasarkan parameter-parameter yang relevan, seperti *source.ip, destination.ip, user.name,* dan field lain yang sesuai dengan jenis kasus. Filter ini disesuaikan berdasarkan struktur data dari *index* terkait dan mengacu pada pedoman prosedur yang berlaku. Visualisasi ketika analisis pada kibana dapat dilihat pada Gambar 3.12, 3.13, dan 3.14.



Gambar 3.12. Analisis pada Discover Elastic



Gambar 3.13. Analisis pada Discover Elastic



Gambar 3.14. Analisis pada Discover Elastic

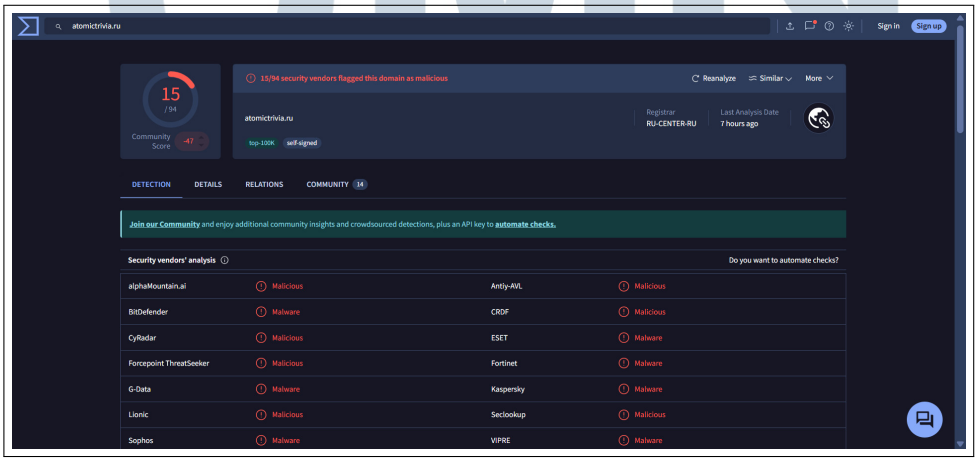
Dalam berbagai jenis kasus, seperti "*NIDS Web Attack Detected*", "*F5-ASM Web Unblock Detected*", *"Login from Outside Indonesia"*, *"IAM Credential Exfiltration Detected from EC2*", atau *"Command and Control (C2) Connection Attempt", filtering* umumnya diawali dengan penelusuran berdasarkan IP sumber *(source IP*). Selanjutnya, *Security Analyst* dapat menggali informasi tambahan yang mendukung konteks, seperti *malicious payload*, lokasi geografis IP, *user agent*

yang digunakan, serta pola koneksi yang mencurigakan. Pendekatan ini membantu memperkuat hipotesis awal dan mempercepat proses identifikasi terhadap potensi ancaman.

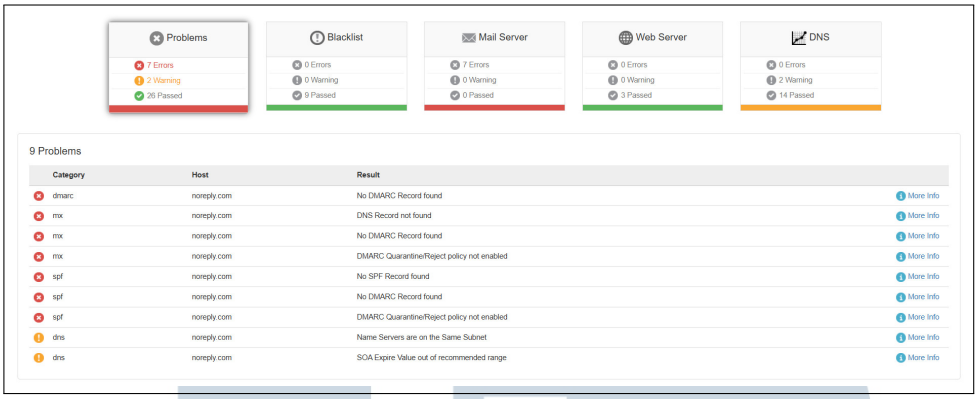
B. Melakukan Analisis dengan Tools Tambahan

Selain mengandalkan Kibana atau Elastic sebagai alat utama dalam menganalisis log, *security analyst* juga menggunakan berbagai *tools* eksternal untuk mendukung validasi data dan memperkuat hasil investigasi. Salah satu *tools* yang umum digunakan adalah VirusTotal, yang memungkinkan pengecekan reputasi *domain* atau URL untuk menentukan apakah suatu entitas terindikasi *malicious*. *Tools* ini sangat efektif dalam menangani kasus-kasus yang melibatkan komunikasi keluar *(outbound communication)* ke *domain* yang mencurigakan.

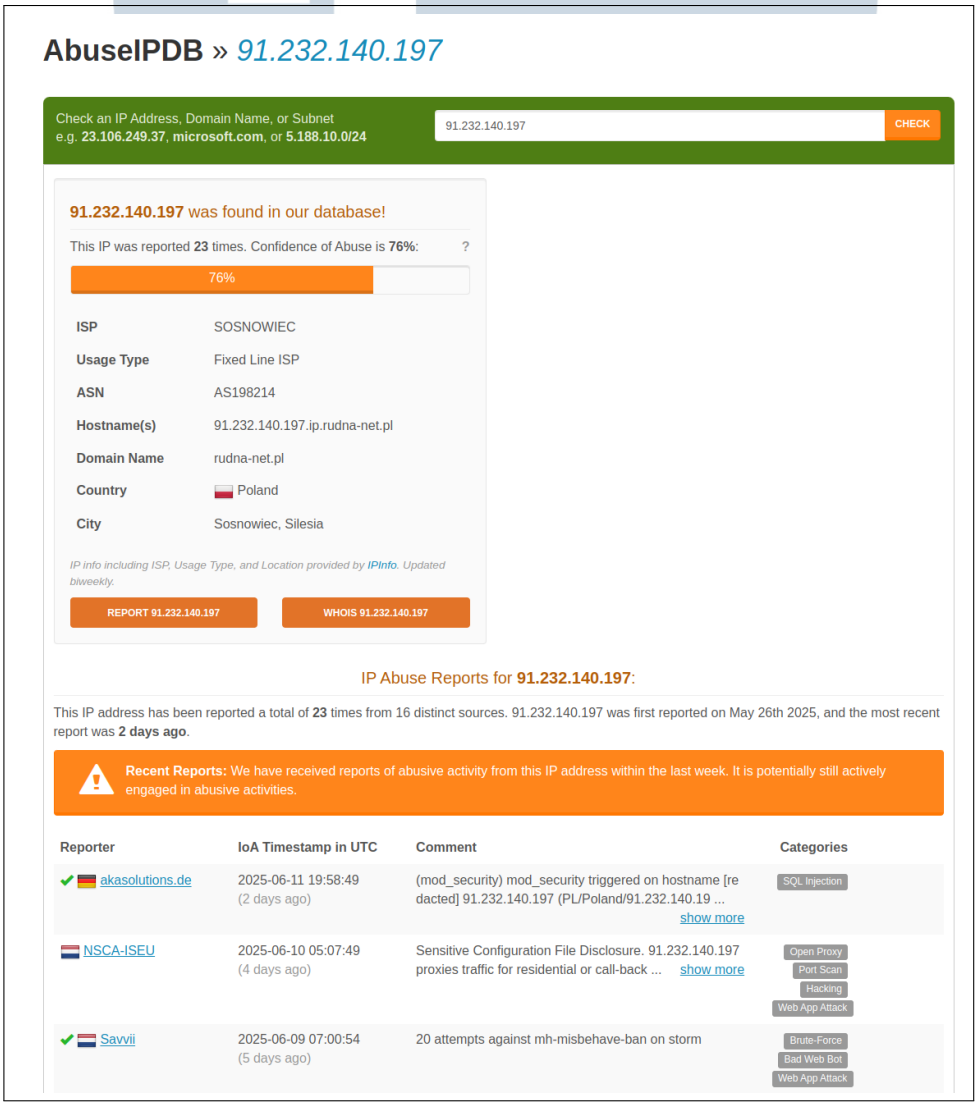
Selain itu, MXToolbox sering digunakan untuk memeriksa kesehatan (*email health*) dan reputasi layanan email, terutama dalam kasus yang melibatkan FortiMail atau Proofpoint, seperti spam atau serangan phishing. Untuk validasi terhadap alamat IP, digunakan tools seperti AbuseIPDB, *Threatbook, Live IP Map, IP Void*, dan *tools* lainnya yang menyajikan informasi apakah suatu IP memiliki riwayat sebagai sumber serangan atau aktivitas berbahaya lainnya. Dengan memadukan hasil analisis internal dari log dan informasi *intelligent* dari *tools* pihak ketiga ini, *security analyst* dapat memperoleh gambaran yang lebih utuh mengenai ancaman yang sedang dianalisis. Berikut adalah visualisasi dari masing-masing *tools* yang digunakan untuk mendukung analisis, seperti VirusTotal, MXToolbox, dan AbuseIPDB yang ditampilkan pada Gambar 3.15, 3.16, dan 3.17.



Gambar 3.15. Visualisasi Virustotal



Gambar 3.16. Visualisasi Mxtoolbox



Gambar 3.17. Visualisasi Abuseipdb

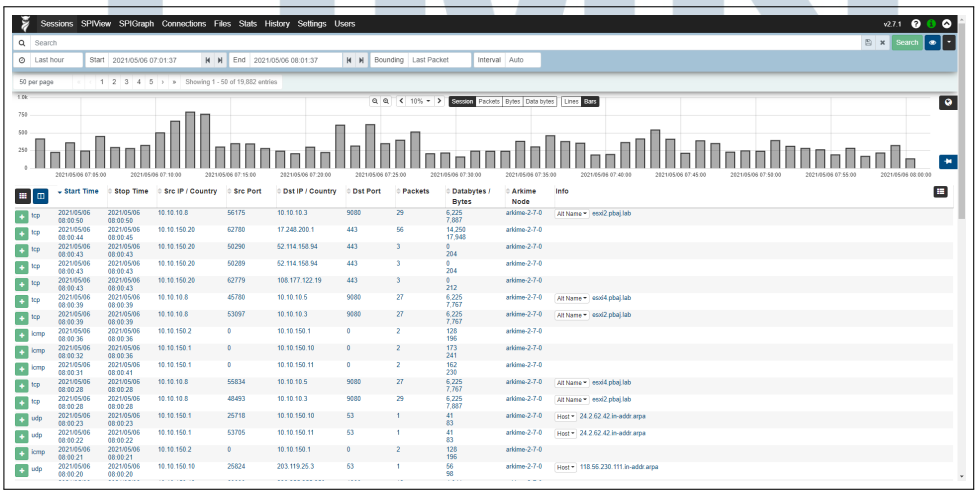
Penggunaan berbagai *tools* sebagai penunjang analisis sangat membantu

dalam pengambilan keputusan yang cepat, akurat, dan berbasis data nyata, sehingga memungkinkan *Security Analyst* untuk merespons ancaman dengan lebih efektif dan mengurangi potensi dampak terhadap sistem dan data organisasi.

C. Melakukan Analisis dengan Moloch

Dalam investigasi kasus yang berbasis NIDS (*Network Intrusion Detection System*), proses analisis lanjutan sering kali memerlukan pemeriksaan yang lebih mendalam terhadap lalu lintas jaringan. Untuk keperluan tersebut, *Moloch* sebuah *platform open-source* yang berfungsi sebagai sistem perekaman dan analisis *packet* jaringan secara mendetail menjadi alat yang sangat penting dan sering digunakan oleh *Security Analyst*. Meskipun sebagian besar informasi awal dapat ditemukan melalui *platform* log seperti Kibana atau Elastic, terdapat beberapa kondisi di mana informasi yang tersedia di sana tidak cukup granular untuk mendukung analisis teknis secara menyeluruh.

Moloch memungkinkan *Security Analyst* untuk menelusuri paket data mentah (*raw packet capture*) secara langsung, sehingga memberikan konteks yang lebih kaya terhadap komunikasi jaringan yang mencurigakan. Misalnya, dalam kasus yang melibatkan komunikasi ke *port* tidak standar, aktivitas command and control (C2), atau upaya eksploitasi yang kompleks, Moloch dapat mengungkap *payload, header,* dan urutan sesi komunikasi secara lebih detail daripada sekadar metadata yang terekam dalam log. Visualisasi Moloch dalam proses analisis *packet* data dapat dilihat pada Gambar 3.18.



Gambar 3.18. Visualisasi Moloch

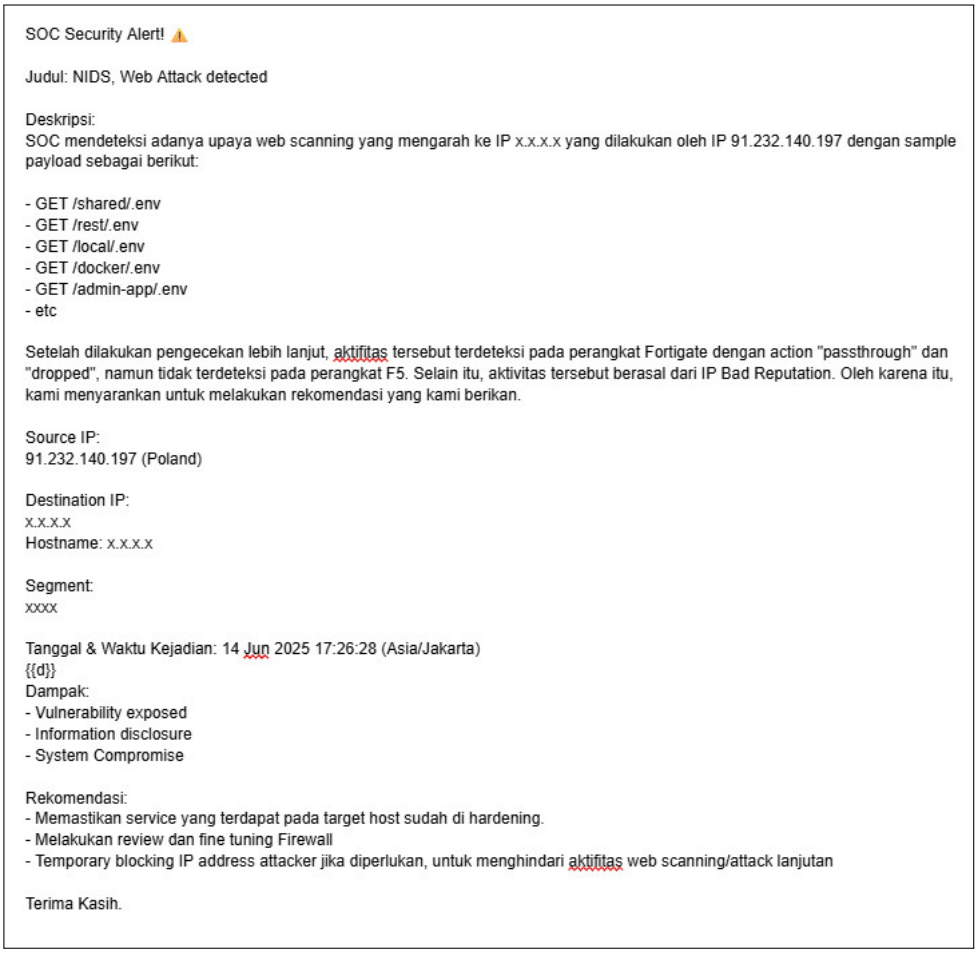
Oleh karena itu, pada investigasi kasus NIDS seperti kasus yang terdeteksi

melalui *index* suricata. *Security Analyst* biasanya akan menggunakan Moloch sebagai alat pelengkap untuk memvalidasi dan memperkuat temuan teknis yang sebelumnya diperoleh melalui Kibana. Data hasil tangkapan dari Moloch juga sering digunakan sebagai *evidence* tambahan dalam proses penyusunan laporan insiden atau notifikasi kepada *customer*.

3.3.7 Notification to Customer via Portal Warning Alert (PWA)

Setelah proses analisis terhadap sebuah kasus selesai dilakukan, langkah selanjutnya yang dilakukan oleh *Security Analyst* adalah menyusun laporan hasil analisis dan menyampaikannya kepada pihak *customer* melalui sistem internal bernama PWA. PWA (*Portal Warning Alert*) merupakan platform internal perusahaan yang dirancang khusus sebagai sistem pengelolaan notifikasi insiden keamanan. Sistem ini digunakan untuk menginformasikan kepada *customer* terkait kasus-kasus yang telah dianalisis secara menyeluruh oleh *Security Analyst*.

Notifikasi melalui PWA umumnya dilakukan dalam dua kondisi. Pertama, ketika hasil analisis menunjukkan bahwa kasus tersebut merupakan ancaman valid (*valid threat*) yang dapat berdampak pada sistem atau data *customer*. Kedua, ketika *Security Analyst* perlu melakukan konfirmasi atau klarifikasi kepada pihak *customer* terkait suatu aktivitas yang terdeteksi apakah aktivitas tersebut merupakan tindakan yang sah (*legitimate*) dan memiliki otorisasi (*authorized*), atau justru merupakan indikasi awal dari aktivitas mencurigakan. Hal ini penting untuk menghindari kesalahan eskalasi serta memastikan bahwa analisis dilakukan secara tepat sasaran. Setelah proses drafting selesai, *security analyst* kemudian dapat melakukan *preview drafting* untuk memastikan semua informasi telah benar dan lengkap sebelum notifikasi dikirim. Visualisasi *preview draft* sebelum notifikasi dapat dilihat pada Gambar 3.19.


SOC Security Alert! ⚠

Judul: NIDS, Web Attack detected

Deskripsi:
SOC mendeteksi adanya upaya web scanning yang mengarah ke IP x.x.x.x yang dilakukan oleh IP 91.232.140.197 dengan sample payload sebagai berikut:

- GET /shared/.env
- GET /rest/.env
- GET /local/.env
- GET /docker/.env
- GET /admin-app/.env
- etc

Setelah dilakukan pengecekan lebih lanjut, aktifitas tersebut terdeteksi pada perangkat Fortigate dengan action "passthrough" dan "dropped", namun tidak terdeteksi pada perangkat F5. Selain itu, aktivitas tersebut berasal dari IP Bad Reputation. Oleh karena itu, kami menyarankan untuk melakukan rekomendasi yang kami berikan.

Source IP:
91.232.140.197 (Poland)

Destination IP:
x.x.x.x
Hostname: x.x.x.x

Segment:
xxxx

Tanggal & Waktu Kejadian: 14 Jun 2025 17:26:28 (Asia/Jakarta)
{{d}}
Dampak:
- Vulnerability exposed
- Information disclosure
- System Compromise

Rekomendasi:
- Memastikan service yang terdapat pada target host sudah di hardening.
- Melakukan review dan fine tuning Firewall
- Temporary blocking IP address attacker jika diperlukan, untuk menghindari aktifitas web scanning/attack lanjutan

Terima Kasih.

Gambar 3.19. Visualisasi Drafting Notifikasi

Dalam situasi di mana sebuah kasus dikonfirmasi sebagai *valid threat*, *security analyst* akan melakukan *drafting* notifikasi di dalam sistem PWA. *Draft* ini berisi penjelasan terperinci mengenai hasil analisis yang telah dilakukan, termasuk di dalamnya rincian teknis kasus, alamat IP atau *hostname* yang terdampak, nama pengguna (*user*) terkait, potensi dampak, serta rekomendasi penanganan yang disarankan. Selain itu, *Security Analyst* juga menyisipkan *evidence* pendukung berupa tangkapan layar (*screenshot*) yang menunjukkan data log atau temuan spesifik dari hasil investigasi. *Evidence* ini bertujuan untuk memperkuat keabsahan laporan dan memberikan gambaran konkret kepada pihak pelanggan mengenai konteks ancaman yang dihadapi.

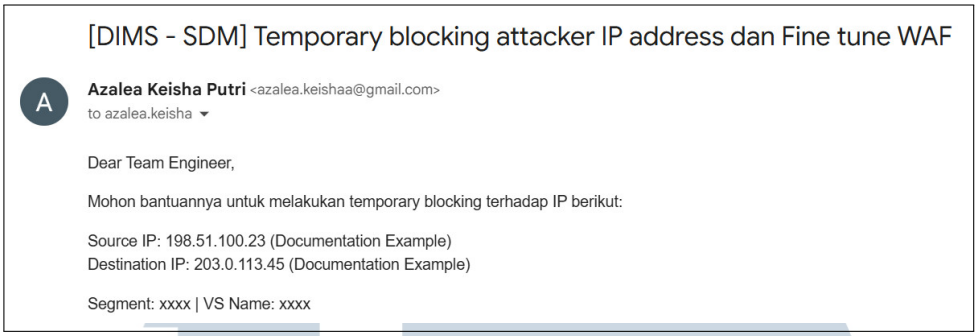
Setelah proses *drafting* selesai dan notifikasi dikirim melalui sistem PWA, sistem akan secara otomatis melakukan penyampaian notifikasi kepada *customer*, baik melalui email resmi maupun melalui grup aplikasi Signal yang telah disepakati bersama dengan *customer*. Proses ini memastikan bahwa pihak pelanggan

menerima informasi secara cepat dan dapat segera melakukan langkah-langkah mitigasi yang diperlukan sesuai dengan rekomendasi yang telah diberikan oleh tim *security analyst*. Dengan demikian, PWA berperan sebagai penghubung utama dalam komunikasi insiden antara tim keamanan internal dan pihak *customer* secara sistematis dan terdokumentasi.

3.3.8 Blocking IP Via Email to Security Device Management (SDM)

Dalam penanganan kasus tertentu, khususnya yang berkaitan dengan sistem proteksi berbasis jaringan seperti F5 dan Akamai, diperlukan tindakan tambahan berupa pemblokiran alamat IP yang terindikasi sebagai sumber ancaman. Pemblokiran ini dilakukan sebagai langkah mitigasi proaktif untuk mencegah potensi serangan lanjutan dari alamat IP yang sama, serta untuk menghindari kerentanan lebih lanjut pada sistem yang menjadi target. IP yang diblokir umumnya merupakan sumber dari aktivitas mencurigakan seperti *brute force login, scanning port* tidak wajar, atau upaya *bypass* proteksi melalui trafik abnormal, yang dapat mengancam ketersediaan dan integritas layanan yang dilindungi oleh F5 atau Akamai. Oleh karena itu, pemblokiran IP menjadi tindakan penting guna mempersempit ruang gerak pelaku dan memperkuat lapisan keamanan perimeter jaringan.

Setelah proses notifikasi insiden dilakukan melalui sistem PWA, dan apabila dalam proses analisis ditemukan bahwa alamat IP tertentu memang perlu diblokir, maka security analyst akan mengajukan permintaan pemblokiran IP kepada divisi terkait, yaitu SDM (Security Device Management). Permintaan ini dilakukan melalui pengiriman email resmi yang memuat informasi lengkap mengenai IP yang ingin diblokir, disertai dengan penjelasan alasan teknis, bukti log, serta referensi *case* ID dari PWA sebagai dasar permintaan. SDM merupakan divisi yang memiliki otoritas dan akses untuk melakukan konfigurasi serta perubahan pada sistem perangkat keamanan, termasuk *firewall* dan *load balancer.* Visualisasi proses pengiriman *email* untuk permintaan pemblokiran IP dapat dilihat pada Gambar 3.20.



Gambar 3.20. Visualisasi Blocking IP Melalui Email

Dengan melibatkan SDM sebagai pihak eksekutor dalam proses *blocking*, seluruh tindakan dapat dilakukan secara terkontrol dan sesuai prosedur, sekaligus memastikan bahwa konfigurasi jaringan tetap terdokumentasi dengan baik.

3.3.9 Waiting for Acknowledge Customer

Setelah notifikasi dikirimkan melalui sistem PWA, *customer* biasanya akan memberikan tanggapan atau *acknowledgment* sebagai bentuk konfirmasi terhadap informasi yang telah disampaikan. Proses ini dilakukan melalui media komunikasi seperti grup *Signal, WhatsApp* atau *email*. Tanggapan dari customer merupakan langkah yang sangat penting dalam menentukan arah tindak lanjut terhadap kasus yang telah dinotifikasi, baik dalam hal validasi apakah aktivitas tersebut merupakan tindakan sah, maupun dalam konteks penanganan insiden apabila aktivitas tersebut merupakan potensi ancaman.

Dalam hal ini, respons *customer* dapat berupa konfirmasi bahwa aktivitas yang dianalisis adalah aktivitas sah (*legitimate*) dan telah memiliki otorisasi yang sesuai (*authorized activity*), sehingga tidak diperlukan eskalasi lebih lanjut. Di sisi lain, jika aktivitas tersebut dinilai memang mengandung unsur ancaman namun telah berhasil diblokir atau dimitigasi oleh sistem internal *customer*, maka insiden dianggap telah tertangani secara efektif. Dalam kedua situasi tersebut, *customer* telah menyelesaikan tanggung jawab validasi mereka dan selanjutnya proses dapat dilanjutkan oleh tim *Security Analyst.*

3.3.10 Close Case Via Portal Warning Alert (PWA)

Setelah *customer* memberikan *acknowledgment* atas notifikasi yang telah dikirim, *security analyst* akan melanjutkan dengan menutup kasus (*close case*) di dalam sistem PWA. Proses ini dilakukan dengan merujuk pada *Case ID*

yang tercantum pada notifikasi sebelumnya, agar pencatatan tetap konsisten dan terdokumentasi dengan baik. Penutupan kasus ini menandakan bahwa seluruh tahapan penanganan mulai dari analisis hingga konfirmasi dari *customer* telah selesai, dan tidak diperlukan tindakan lanjutan terhadap kasus tersebut.

3.3.11 Recap Case Per-2 Hours

Catatan *Monitoring* yang dilakukan secara berkala setiap dua jam merupakan salah satu prosedur penting dalam operasional tim *Security Operations Center (SOC)* yang bertujuan untuk mengurangi jumlah case yang belum diproses (*unprocessed case*) dan memastikan seluruh kasus yang masuk dapat ditangani secara tepat waktu. Kegiatan *monitoring* ini mencakup proses pencatatan dan pembuatan ringkasan (*recap*) terhadap seluruh *case* yang terdeteksi dalam kurun waktu dua jam terakhir, baik untuk kasus dengan tingkat *severity high, medium,* maupun *low*.

Untuk mendukung proses ini, data log terkait *case-case* yang muncul dikumpulkan dari sistem Elastic dan diunduh dalam format CSV. File CSV tersebut kemudian diproses menggunakan *Google Colab*, yaitu *platform* komputasi berbasis *cloud* yang mendukung eksekusi kode *Python*. Di dalam *Google Colab*, data CSV tersebut dijalankan melalui *script Python* yang telah disiapkan sebelumnya untuk melakukan proses rekap otomatis. *Script* ini secara otomatis akan mengelompokkan case berdasarkan *severity*, *subject*, serta menyiapkan format draft recap yang siap untuk dilaporkan.

Setelah rekap disusun, *drafting recap* hasil *monitoring* tersebut akan dibagikan ke dalam grup komunikasi tim SOC sebagai bentuk pelaporan berkala dan transparansi status investigasi. *Draft recap* ini juga dilengkapi dengan *evidence* pendukung berupa tangkapan layar (*screenshot*) dari sistem Thruk (yang memantau layanan dan sistem infrastruktur) serta dari *dashboard* Elastic untuk menunjukkan bukti visual dari data yang dilaporkan. Dengan pendekatan ini, tim SOC dapat memastikan seluruh aktivitas pemantauan dilakukan secara konsisten, terdokumentasi, dan dapat ditindaklanjuti dengan cepat apabila ditemukan kasus yang perlu diprioritaskan untuk investigasi lebih lanjut.

3.3.12 Handover Notes

Handover Notes disusun setiap pergantian *shift* untuk memastikan transisi tugas berjalan lancar dan tidak ada informasi penting yang terlewat. Catatan ini mencakup daftar personel yang bertugas di *shift* sebelumnya dan *shift* saat ini, jumlah *case* yang sudah dinotifikasi, serta total *unhandled case* yang dibagi berdasarkan tingkat *severity: high, medium*, dan *low*.

Selain itu, *handover* juga memuat hasil monitoring dari berbagai entitas *customer*, termasuk status *log* dan *agent*, seperti "log sepi", "sudah eskalasi", atau "konfirmasi customer". Informasi ini dikumpulkan dari sistem seperti Thruk, Wazuh, CrowdStrike, F5, dan lainnya. *Link* akses ke *dashboard* Grafana juga disertakan agar tim berikutnya dapat langsung melakukan pemantauan lanjutan.

Handover Notes juga mencatat *customer concern*, seperti instruksi khusus untuk format notifikasi atau *masking* data sensitif. Dengan adanya *handover* yang jelas memudahkan tim SOC *shift* selanjutnya untuk langsung melanjutkan tugas dan menangani kasus tanpa ada yang terlewatkan.

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala

Selama proses kerja magang berlangsung, terdapat beberapa kenda;a yang dihadapi baik itu besar maupun kecil. Berikut adalah beberapa kendala yang dihadapi selama praktik kerja magang di PT Defender Nusa Semesta.

1. Log Sistem Tidak Stabil, terdapat insiden di mana log dari sistem monitoring tidak masuk sebagaimana mestinya atau tidak terekam karena server log mengalami *downtime*.

2. Koneksi WiFi Tidak Konsisten, beberapa kali koneksi WiFi terputus secara tiba-tiba, menyebabkan gangguan terhadap proses kerja dan akses sistem internal.

3. Akses ke Kibana Terganggu, *platform* Kibana tidak dapat diakses karena terdapat masalah pada konektivitas antara DSA dan *Checkpoint customer.* Keduanya tidak tersinkronisasi (*sync*), sehingga menyebabkan VPN harus dilakukan *restart* secara manual.

4. VPN Tidak Dapat Digunakan, dalam beberapa kesempatan, VPN tidak dapat digunakan untuk mengakses jaringan internal perusahaan selama periode tertentu.

5. Akses ke Checkpoint Harmony Terblokir, sistem keamanan Checkpoint Harmony tidak dapat diakses karena perangkat belum direstart, serta entri IP dan *hostname* belum ditambahkan ke dalam file *hosts* pada sistem operasi Ubuntu.

3.4.2 Solusi

Dari berbagai kendala yang dihadapi, berikut adalah solusi yang ditemukan untuk mengatasi kendala selama praktik kerja magang di PT Defender Nusa Semesta

1. Dilakukan pengecekan melalui *platform trhuk* dan *Grafana* untuk memverifikasi status log. Jika ditemukan log yang tidak aktif atau tidak tercatat, masalah tersebut segera dieskalasikan ke tim *System Administrator* melalui tiket resmi.

2. Masalah koneksi WiFi dilaporkan dan dieskalasikan kepada tim *General Affairs (GA)* untuk penanganan infrastruktur jaringan secara internal.

3. *Security Analyst* melakukan *ping test* terhadap alamat IP DSA untuk memastikan konektivitas. Hasil tangkapan layar (*screenshot*) dari proses ini dijadikan bukti untuk mendukung eskalasi ke tim *System Administrator*.

4. Saat terjadi gangguan VPN, tim segera mengadakan pertemuan dadakan (*impromptu meeting*) untuk melakukan *troubleshooting*. Langkah utama yang diambil meliputi pembuatan konfigurasi VPN baru, validasi ulang konfigurasi tersebut, serta penyesuaian pada *file* konfigurasi dan direktori *root* melalui terminal Linux guna memastikan koneksi kembali stabil dan dapat digunakan.

5. Pemulihan akses ke Checkpoint Harmony dilakukan melalui proses rekonfigurasi sistem. Langkah yang diambil adalah menambahkan entri IP dan *hostname* terkait ke dalam file /etc/hosts pada sistem operasi Ubuntu. Setelah perubahan tersebut disimpan, sistem kemudian direstart untuk memuat ulang konfigurasi jaringan. Proses ini memastikan bahwa koneksi ke Checkpoint Harmony dapat berjalan dengan normal kembali.