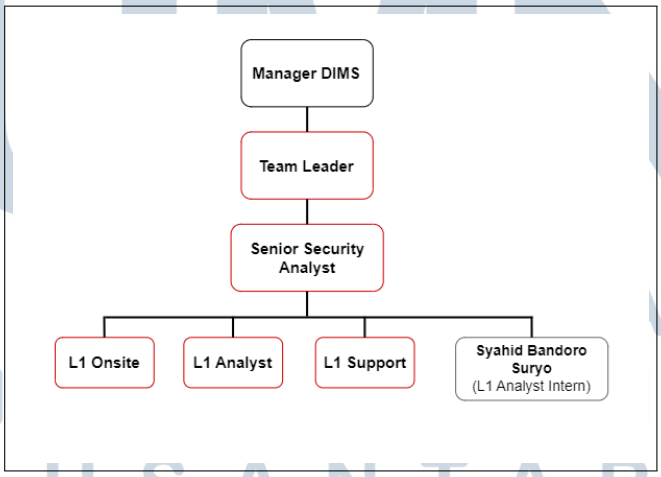
BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama mengikuti program magang di PT Defender Nusa Semesta, peran yang dijalankan berada pada level *L1 Security Analyst* sebagai bagian dari operasional *Security Operations Center* (SOC). Tanggung jawab utama meliputi kegiatan *security monitoring*, yaitu melakukan analisis awal terhadap log dan notifikasi keamanan melalui sistem pemantauan yang digunakan di lingkungan perusahaan. Dalam pelaksanaannya, kegiatan magang berada di bawah bimbingan seorang *Senior Security Analyst* (SSA) yang berfungsi sebagai *buddy*, serta berkoordinasi dengan anggota tim pada jadwal *shift* yang sama.

Unit kerja yang menangani operasional SOC berada di bawah Divisi *Defenxor Intelligence Managed Security* (DIMS). Divisi ini mencakup beberapa fungsi, yaitu tim analis keamanan (*security analyst*), tim pengelola perangkat keamanan (*security device management* atau SDM), dan tim *system administrator*. Seluruh aktivitas divisi dipimpin oleh *Team Leader SOC Operation* di bawah koordinasi *Manager DIMS*. Struktur organisasi *DIMS* yang menggambarkan posisi dalam tim dapat dilihat pada Gambar 3.1, yang menunjukkan hierarki dan jalur koordinasi yang berlaku selama masa magang berlangsung.



Gambar 3.1. Struktur *DIMS (Defenxor Intelligence Managed Security)*

Alur koordinasi kerja dalam SOC mengikuti *escalation matrix* yang telah ditetapkan oleh Divisi DIMS. Ketika sebuah insiden atau kendala teknis muncul, penanganan diawali melalui diskusi internal dengan rekan satu *shift* sebagai

tingkat respons pertama. Jika permasalahan tidak dapat diselesaikan pada level tersebut, proses dilanjutkan dengan eskalasi kepada *Senior Security Analyst* yang bertanggung jawab terhadap klien terkait. Apabila penanganan memerlukan keputusan atau tindakan lebih lanjut, insiden diteruskan kepada *Team Leader*, dan pada tahap akhir kepada *Manager DIMS*. Struktur eskalasi berjenjang ini memastikan setiap insiden mendapatkan penanganan sesuai tingkat urgensi dan kompleksitasnya.

Dalam struktur operasional SOC di Defenxor, analis tingkat pertama terbagi ke dalam beberapa kategori, yaitu *L1 Onsite Analyst* yang ditempatkan langsung di lokasi klien, *L1 Analyst* yang melakukan pemantauan dari kantor pusat, serta *L1 Support* yang berperan sebagai penghubung koordinasi antara SOC dan klien.

3.2 Tugas yang Dilakukan

Selama pelaksanaan program magang tahap kedua, berbagai tugas dilaksanakan yang berkaitan dengan operasional *Security Operations Center* (SOC) serta analisis sistem deteksi intrusi berbasis *host*. Adapun tugas-tugas yang dilakukan meliputi:

1. Security Monitoring melalui SIEM
   Pemantauan aktivitas keamanan sistem klien dilakukan secara *real-time* menggunakan platform *Security Information and Event Management* (SIEM). Kegiatan ini mencakup pengamatan *event log*, identifikasi anomali, serta penilaian awal terhadap *security alert* yang berpotensi mengarah pada insiden keamanan.

2. Analisis Alert dan Penyusunan Notifikasi Insiden
   Analisis lanjutan terhadap *alert* dilakukan untuk menentukan tingkat risiko dan relevansinya. Hasil analisis kemudian disusun dalam bentuk notifikasi insiden yang berisi deskripsi teknis, indikator ancaman, serta rekomendasi tindakan mitigasi yang disampaikan kepada klien.

3. Penyusunan Laporan Bulanan Keamanan
   Penyusunan *monthly report* dilakukan sebagai rangkuman aktivitas keamanan dalam satu periode tertentu, yang mencakup statistik *alert*, klasifikasi ancaman, tren aktivitas mencurigakan, serta evaluasi efektivitas sistem monitoring sebagai bahan asesmen keamanan berkelanjutan.

4. Analisis Integrasi Wazuh HIDS
   Analisis terhadap integrasi Wazuh *Host Intrusion Detection System* (HIDS) dilakukan pada lingkungan simulasi SOC, meliputi proses instalasi *agent*, konfigurasi modul deteksi, serta integrasi dengan platform SIEM.

5. Analisis File Integrity Monitoring (FIM)
   Analisis mekanisme kerja modul *File Integrity Monitoring* (FIM) pada Wazuh HIDS dilakukan untuk mendeteksi perubahan berkas seperti *added* dan *modified*. Analisis mencakup pengamatan alur deteksi mulai dari perubahan berkas pada *endpoint*, pengiriman log ke *Wazuh Manager*, hingga visualisasi *alert* pada dashboard analitik.

Seluruh tugas tersebut dilaksanakan dengan mengikuti prosedur operasional dan standar keamanan informasi yang berlaku di lingkungan SOC, sehingga mendukung proses monitoring dan analisis keamanan secara aman dan terstruktur.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan magang pada tahap kedua difokuskan pada kegiatan pengembangan sistem keamanan informasi yang berkaitan langsung dengan deteksi intrusi berbasis *host* di lingkungan simulasi SOC Defenxor. Seluruh kegiatan diarahkan untuk memahami proses kerja *Host Intrusion Detection System* (HIDS), khususnya modul *File Integrity Monitoring* (FIM), serta integrasinya dengan platform *Security Information and Event Management* (SIEM).

Dalam kegiatan ini, Wazuh digunakan sebagai platform utama karena menyediakan kemampuan deteksi intrusi, pemantauan integritas berkas, dan analisis log secara terpadu. Proses diawali dengan instalasi dan konfigurasi *Wazuh Agent* pada sistem target yang digunakan sebagai *endpoint simulasi*. Agent ini bertugas memantau perubahan berkas, aktivitas sistem, serta kejadian yang berpotensi menjadi indikator insiden keamanan. Data yang dikumpulkan kemudian dikirimkan ke *Wazuh Manager* untuk diproses lebih lanjut.

Integrasi Wazuh dengan Elastic Stack dilakukan untuk mendukung proses korelasi dan visualisasi log. Dalam hal ini, Elasticsearch berfungsi sebagai tempat penyimpanan data terpusat dan Kibana digunakan untuk menampilkan hasil deteksi dalam bentuk visualisasi yang dapat dianalisis secara langsung. Integrasi ini memungkinkan proses pemantauan terhadap kejadian pada *host* dilakukan

secara *real-time*, termasuk pendeteksian perubahan berkas seperti *added*, *modified*, maupun *deleted*.

Selain proses instalasi dan konfigurasi, kegiatan magang juga meliputi pengujian modul *File Integrity Monitoring* melalui simulasi perubahan berkas pada *endpoint*. Pengujian ini dilakukan untuk memahami bagaimana sistem mendeteksi anomali, bagaimana *alert* dikirimkan ke *Wazuh Manager*, dan bagaimana data tersebut divisualisasikan pada dashboard SIEM. Melalui proses ini, diperoleh pemahaman menyeluruh mengenai alur deteksi FIM mulai dari sumber kejadian hingga tahap analisis.

Berikut uraian kegiatan yang dilakukan dalam pelaksanaan Magang di PT Defender Nusa Semesta dalam Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1-3	Melakukan monitoring keamanan sistem menggunakan platform SIEM dalam skema kerja *shift* dengan menganalisis log keamanan untuk mengidentifikasi potensi ancaman dan anomali pada sistem klien. Selain itu, menangani permintaan (*request*) dari klien terkait kebutuhan monitoring dan menyusun notifikasi insiden. Pada awal bulan, dilakukan pula penyusunan *monthly report* sebagai rekapitulasi aktivitas keamanan sistem.
4-7	Melanjutkan kegiatan *shifting*, serta mulai mempelajari konsep dasar dan arsitektur *Host-based Intrusion Detection System* (HIDS) menggunakan Wazuh dan ELK Stack, termasuk alur komunikasi antar komponen seperti *Wazuh Agent*, *Wazuh Manager*, dan Elasticsearch melalui dokumentasi teknis.
8-11	Melakukan penyiapan dan pengujian lingkungan HIDS berbasis Wazuh pada sistem virtual, termasuk konfigurasi komponen Wazuh dan Elastic Stack. Kegiatan difokuskan pada verifikasi konektivitas, pengumpulan log, dan pengujian awal fungsi deteksi.
	Lanjutan pada halaman berikutnya

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

Minggu Ke -	Pekerjaan yang dilakukan
12-15	Mempelajari mekanisme *File Integrity Monitoring* (FIM) pada Wazuh HIDS. Dilakukan konfigurasi *syscheck rules* dan integrasi VirusTotal untuk memahami alur deteksi perubahan file serta proses *threat intelligence enrichment*. Selain itu, dilakukan peninjauan terhadap hasil *security event* dan *alert* yang ditampilkan melalui platform SIEM.
16	Melakukan simulasi perubahan file untuk menguji efektivitas *File Integrity Monitoring* dan integrasi VirusTotal dalam mendeteksi aktivitas mencurigakan. Selanjutnya dilakukan evaluasi hasil deteksi, dokumentasi temuan, serta penyusunan laporan pengujian sebagai bagian dari laporan akhir. Kegiatan monitoring SOC tetap dilaksanakan sebagai tugas rutin harian.

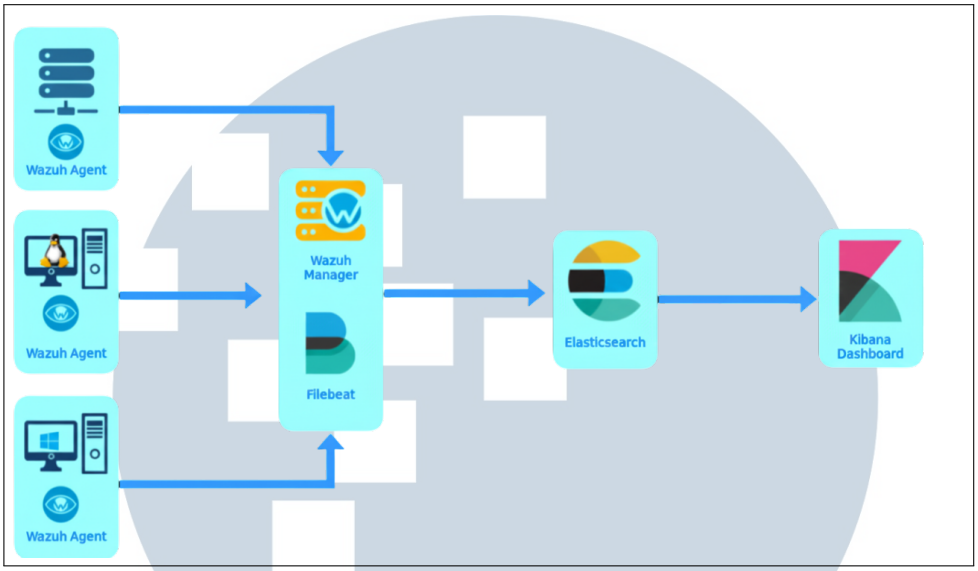
3.3.1 Komponen Utama dan Alur Sistem

Integrasi sistem *File Integrity Monitoring* (FIM) pada kegiatan magang ini dilakukan dengan memanfaatkan Wazuh sebagai platform *Host Intrusion Detection System* (HIDS) yang mampu mendeteksi perubahan pada berkas sistem secara *real-time*, seperti aktivitas *added*, *modified*, dan *deleted*. Sistem ini diintegrasikan dengan komponen *Elastic Stack* (ELK) yang berfungsi sebagai media penyimpanan, pemrosesan, serta visualisasi data, sehingga mampu mensimulasikan alur kerja operasional SOC dalam memantau ancaman berbasis perubahan file.

Alur sistem terdiri atas beberapa komponen utama, yaitu *Wazuh Agent*, *Wazuh Manager*, Filebeat, Elasticsearch, dan Kibana. *Wazuh Agent* dipasang pada *host* yang dipantau dan bertugas melakukan pemindaian integritas file berdasarkan kebijakan FIM yang telah ditentukan. Setiap perubahan file yang terdeteksi baik-penambahan, modifikasi, maupun penghapusan—akan dikirimkan ke *Wazuh Manager* untuk dilakukan proses analisis awal.

Selanjutnya, Filebeat digunakan untuk meneruskan data log hasil analisis dari *Wazuh Manager* menuju Elasticsearch sebagai pusat penyimpanan dan pemrosesan data. Data yang telah terindeks ini kemudian divisualisasikan melalui Kibana dalam bentuk dashboard sehingga memudahkan proses analisis, korelasi,

dan pemantauan oleh analis SOC. Gambar 3.2 adalah alur kerja dari sistem HIDS.



Gambar 3.2. Alur Kerja Wazuh HIDS

A Wazuh Agent

Wazuh Agent merupakan komponen yang dipasang pada *host* atau endpoint yang akan dipantau. Pada konteks *File Integrity Monitoring* (FIM), agent bertanggung jawab untuk melakukan pemindaian terhadap direktori atau berkas yang telah ditentukan dalam konfigurasi. Agent akan mencatat setiap perubahan yang terjadi, seperti pembuatan berkas baru, modifikasi konten, perubahan hak akses, maupun penghapusan berkas. Informasi perubahan tersebut kemudian dikemas menjadi *event* dan dikirimkan ke Wazuh Manager untuk dianalisis lebih lanjut.

B Wazuh Manager

Wazuh Manager berfungsi sebagai pusat pemrosesan dan analisis data yang dikirimkan oleh *Wazuh Agent*. Pada tahap ini, setiap *event* hasil deteksi FIM akan diproses menggunakan aturan (*ruleset*) yang dimiliki Wazuh untuk menentukan tingkat keparahan serta klasifikasi ancaman. Manager juga menyediakan mekanisme korelasi, deteksi anomali, dan penentuan level *alert*. Setelah proses analisis selesai, data akan diteruskan ke komponen selanjutnya untuk disimpan dan divisualisasikan.

C Filebeat

Filebeat merupakan *lightweight shipper* yang digunakan untuk mengirimkan log dari *Wazuh Manager* ke Elasticsearch. Dalam proses ini, Filebeat mengambil log yang dihasilkan oleh *Wazuh Manager*, mengemasnya dalam format yang sesuai, dan mengirimkannya ke Elasticsearch untuk diindeks. Filebeat memastikan bahwa log dikirim secara efisien, konsisten, dan dapat dipantau alur pengirimannya.

D Elasticsearch

Elasticsearch adalah komponen utama dalam Elastic Stack yang digunakan sebagai pusat penyimpanan dan pengindeksan data. Semua log FIM yang dikirim oleh Filebeat akan disimpan dalam indeks tertentu sehingga dapat diakses dengan cepat berdasarkan pencarian kata kunci, filter, atau kueri tertentu. Kemampuan Elasticsearch dalam mengelola dan memproses data dalam jumlah besar menjadikannya elemen penting dalam analisis keamanan berbasis log.

E Kibana

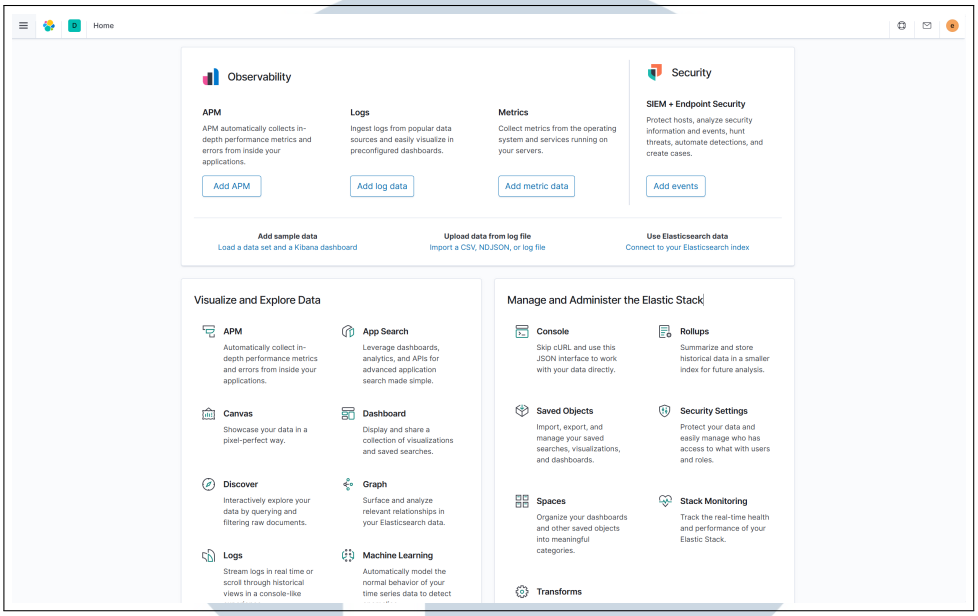
Kibana berfungsi sebagai antarmuka visual untuk menampilkan data yang tersimpan di Elasticsearch. Dalam konteks kegiatan magang ini, Kibana digunakan untuk menampilkan *dashboard* FIM, seperti daftar perubahan file, frekuensi aktivitas tertentu, atau tren modifikasi yang terjadi pada *host*. Melalui visualisasi tersebut, analis SOC dapat melakukan pengamatan, identifikasi anomali, serta memahami konteks insiden dengan lebih cepat dan mudah.

3.3.2 Tampilan Dashboard Kibana dan Navigasi Utama

Kibana menyediakan antarmuka utama yang digunakan untuk berinteraksi dengan data hasil pemantauan yang dikirimkan oleh Wazuh dan komponen Elastic Stack lainnya. Saat pertama kali diakses, pengguna diarahkan menuju halaman awal yang menampilkan rangkuman fitur serta akses menuju modul-modul analisis. Tampilan ini berperan sebagai orientasi awal sebelum memasuki proses pemantauan yang lebih mendalam.

Sebagaimana ditunjukkan pada Gambar 3.3, tampilan awal Kibana menyediakan beberapa area penting seperti daftar fitur, objek tersimpan, dan akses menuju fungsi yang sering digunakan. Susunan elemen ini membantu

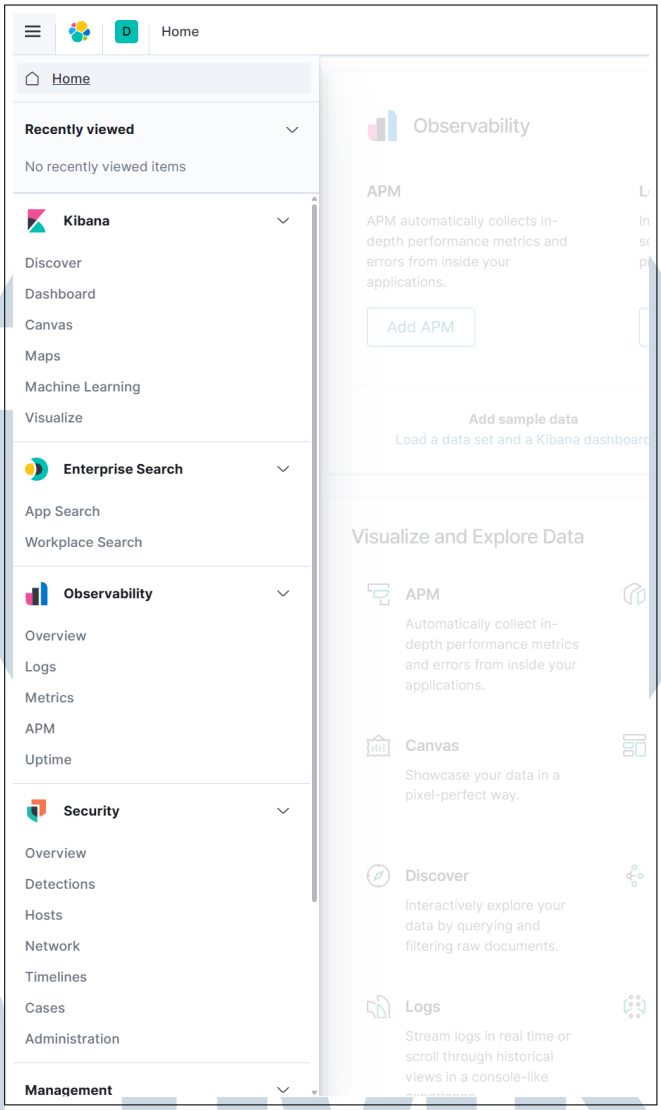
pengguna menentukan langkah kerja selanjutnya tanpa harus melakukan navigasi yang kompleks.



Gambar 3.3. *Halaman Utama Kibana*

Integrasi modul Wazuh juga tersedia di dalam navigasi Kibana, memungkinkan akses langsung menuju data *host monitoring* seperti *security events*, status agen, serta informasi *file integrity*. Seluruh proses analisis dapat dilakukan dalam satu platform tanpa memerlukan aplikasi tambahan, sehingga alur investigasi menjadi lebih efisien. Secara keseluruhan, struktur navigasi Kibana membentuk alur kerja yang runtut—dimulai dari halaman utama, dilanjutkan melalui panel navigasi, hingga menuju modul analisis yang lebih mendalam—yang secara langsung mendukung kebutuhan operasional SOC dalam proses observasi, investigasi, dan korelasi data.

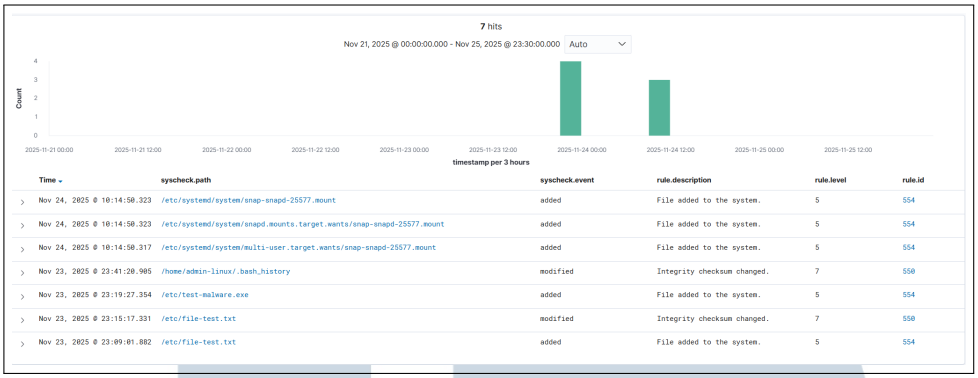
Struktur navigasi dikendalikan melalui sidebar sebagaimana ditunjukkan pada Gambar 3.4. Menu pada sidebar mengarahkan pengguna menuju berbagai fungsi inti seperti halaman Discover untuk pencarian log, Dashboard untuk tampilan visual, hingga menu Management untuk mengatur indeks serta konfigurasi sistem. Penempatan seluruh menu pada satu panel membuat proses perpindahan antarfitur menjadi lebih terarah dan konsisten.



Gambar 3.4. *Navigasi Sidebar Kibana*

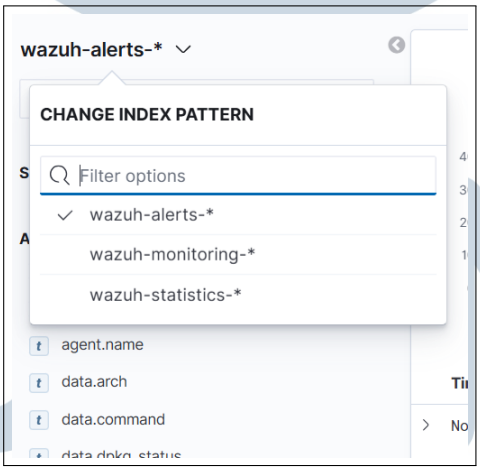
A Discover

Halaman Discover merupakan salah satu fitur utama pada Kibana yang digunakan untuk melakukan eksplorasi data log secara menyeluruh. Fitur ini menampilkan log dalam bentuk tabel yang tersusun berdasarkan *timestamp*, sehingga mempermudah pengguna dalam meninjau urutan peristiwa secara kronologis. Tampilan Discover menyediakan kolom pencarian, daftar *fields*, serta area tampilan dokumen yang dapat diperluas untuk melihat detail setiap log.



Gambar 3.5. *Tampilan Discover pada Kibana*

Seperti terlihat pada Gambar 3.5, halaman Discover menampilkan data log dalam format yang mudah dibaca dan dilengkapi dengan opsi untuk memperluas informasi setiap entri. Bagian kiri halaman menunjukkan daftar *fields* yang tersedia, memungkinkan pengguna untuk menambahkan atau menghapus kolom sesuai kebutuhan analisis. Sementara itu, bagian atas dilengkapi dengan fitur pencarian berbasis *Kibana Query Language* (KQL), yang mempermudah proses penyaringan log berdasarkan parameter tertentu.



Gambar 3.6. *Index Pattern*

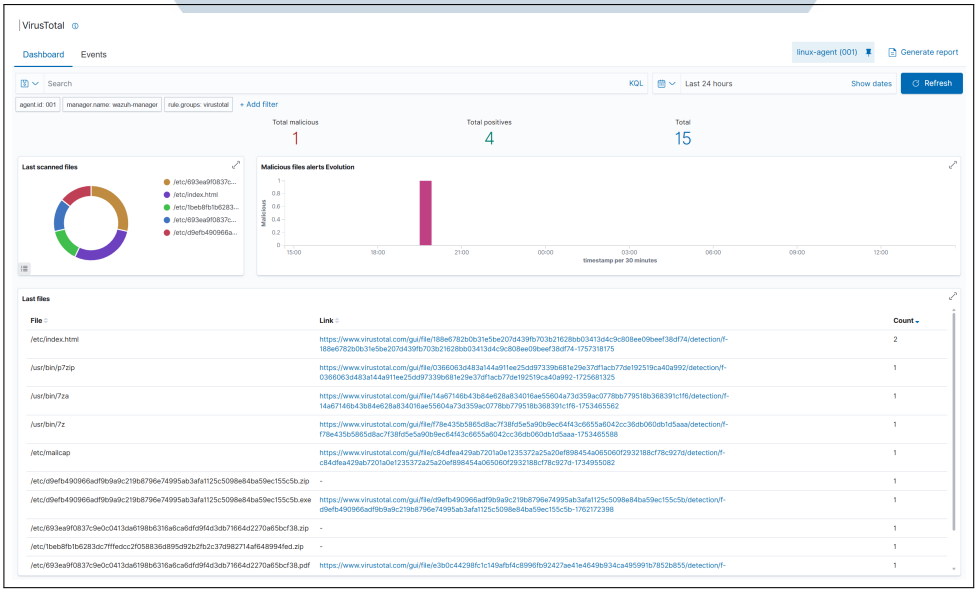
Salah satu elemen penting pada halaman Discover adalah kemampuan untuk memilih *index pattern*, seperti yang ditunjukkan pada Gambar 3.6. Pemilihan *index pattern* ini menentukan dari kumpulan data mana log akan ditampilkan. Dengan memilih indeks yang sesuai, seperti indeks khusus Wazuh, analis dapat membatasi ruang pencarian agar lebih fokus terhadap konteks yang sedang ditinjau, misalnya data *file integrity monitoring*, *security alerts*, atau log sistem lainnya.

Kehadiran Discover sangat penting dalam proses analisis karena memungkinkan tim SOC melakukan pencarian cepat terhadap aktivitas tertentu,

mengidentifikasi pola, hingga memverifikasi apakah suatu kejadian berhubungan dengan anomali keamanan. Melalui tampilan yang interaktif dan fleksibel, Discover mendukung proses investigasi awal sebelum data dianalisis lebih lanjut di dashboard atau visualisasi lainnya.

B Dashboard

Dashboard pada Kibana berfungsi sebagai halaman pemantauan yang menyajikan data log dalam bentuk grafik, diagram, tabel, maupun *metric view*. Berbeda dengan halaman Discover yang menampilkan log dalam bentuk data mentah, Dashboard menyajikan ringkasan informasi sehingga membantu analis dalam memahami tren, pola, serta anomali dari kumpulan data dalam jumlah besar. Penyajian data dalam bentuk panel ini memudahkan proses monitoring kondisi keamanan secara menyeluruh.



Gambar 3.7. *Contoh Tampilan Dashboard Integrasi VirusTotal pada Kibana*

Sebagaimana terlihat pada Gambar 3.7, halaman Dashboard terdiri dari sejumlah panel yang masing-masing menampilkan ringkasan data berdasarkan kriteria tertentu. Panel-panel tersebut dapat disusun, diatur ulang, maupun disesuaikan sesuai dengan kebutuhan pemantauan. Setiap panel merepresentasikan hasil pengolahan data log dari indeks Elasticsearch, termasuk data yang dihasilkan oleh integrasi Wazuh.

Dalam penggunaannya, Dashboard memungkinkan analis SOC untuk memantau kondisi sistem secara *real-time*. Informasi seperti jumlah *security alerts*,

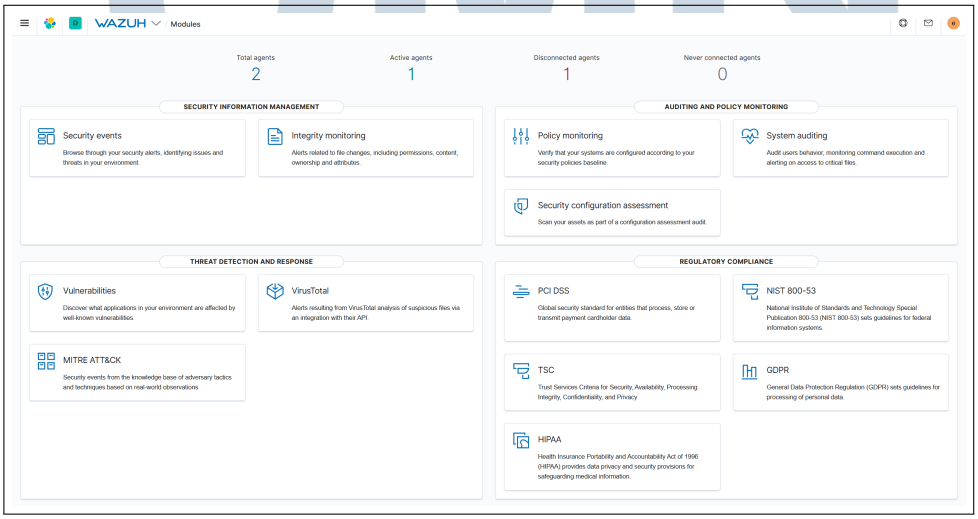
aktivitas *File Integrity Monitoring*, status agent, serta tren kejadian keamanan dapat diamati secara cepat tanpa perlu menelusuri log satu per satu. Hal ini sangat membantu dalam mendeteksi indikasi awal anomali atau lonjakan aktivitas yang berpotensi mencurigakan.

Hubungan antara Dashboard dan Discover bersifat saling melengkapi. Data yang ditampilkan pada Dashboard berasal dari sumber log yang sama dengan yang ditampilkan pada Discover. Ketika Dashboard menunjukkan adanya pola tidak normal atau peningkatan aktivitas tertentu, analis dapat langsung berpindah ke halaman Discover untuk meninjau detail log yang mendasarinya secara lebih teknis. Dengan demikian, Dashboard berperan sebagai sarana pemantauan awal, sedangkan Discover digunakan untuk analisis lanjutan dan investigasi mendalam.

Secara keseluruhan, Dashboard menjadi komponen penting dalam operasional SOC karena menyederhanakan proses monitoring dan mempercepat pengambilan keputusan. Penyajian data yang terstruktur dan ringkas memungkinkan analis untuk memahami kondisi keamanan sistem dengan lebih cepat serta menentukan langkah tindak lanjut secara lebih efektif.

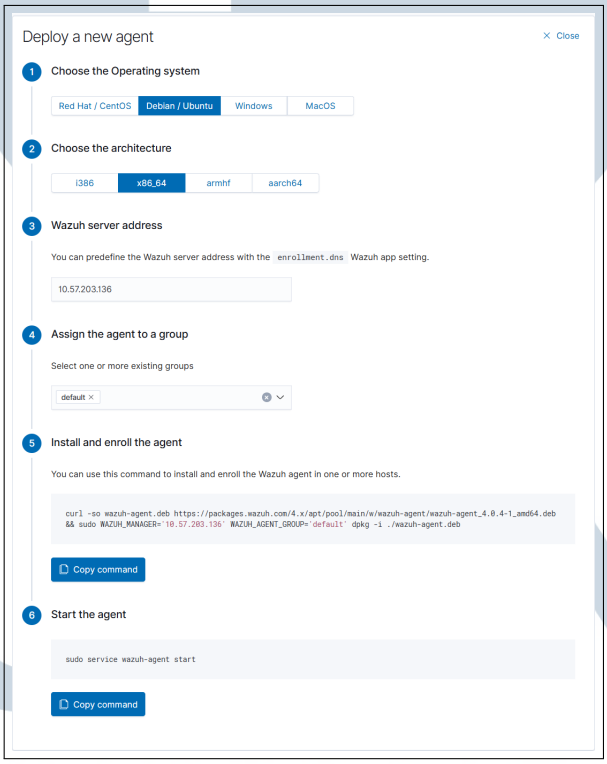
C Wazuh App

Wazuh App merupakan modul khusus pada Kibana yang berfungsi untuk menampilkan seluruh data hasil deteksi dan aktivitas keamanan yang dikumpulkan oleh Wazuh. Melalui aplikasi ini, analis dapat melakukan pemantauan menyeluruh terhadap kondisi agent, hasil *File Integrity Monitoring*, *security events*, serta berbagai komponen lain yang berkaitan dengan operasional HIDS.



Gambar 3.8. *Tampilan Wazuh App pada Kibana*

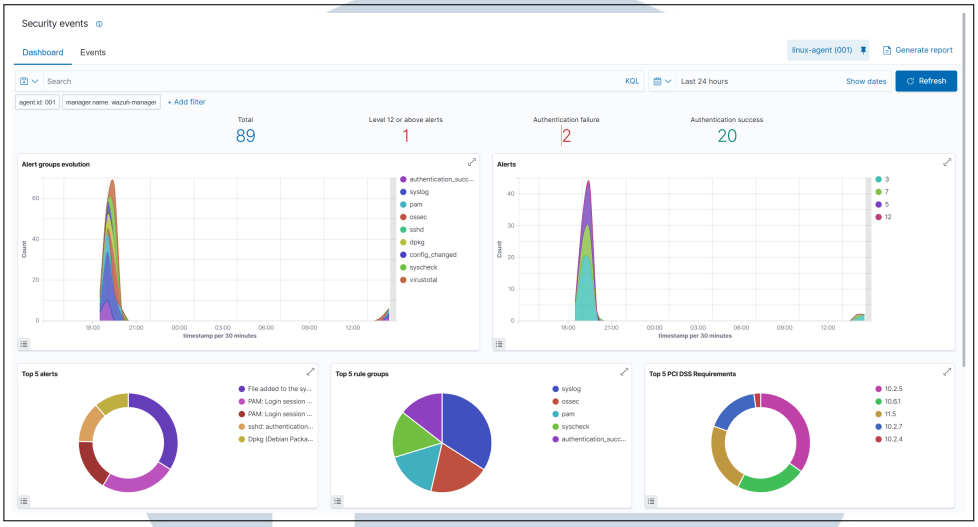
Pada tampilan awal *Wazuh App* (Gambar 3.8), pengguna dapat melihat ringkasan kondisi agent dan berbagai kategori event yang diklasifikasikan oleh sistem. Salah satu fungsi utama modul ini adalah memantau status *Wazuh Agent* yang terpasang pada berbagai *host*. Status agent ditampilkan dalam dua kondisi utama, yaitu *active* dan *disconnected*. Menjaga agent tetap aktif sangat penting karena agent merupakan komponen yang mengirimkan data pemantauan ke server Wazuh. Jika agent tidak aktif, maka sistem tidak dapat menerima data apa pun dari *host* tersebut, sehingga aktivitas perubahan file, potensi ancaman, dan anomali sistem tidak akan terpantau. Kondisi ini dapat menyebabkan *host* menjadi area tanpa visibilitas (*blind spot*) dan berpotensi menghambat proses analisis jika terjadi insiden keamanan.



Gambar 3.9. Penambahan agent

Seperti yang ditunjukkan pada Gambar 3.9, selain pemantauan status agent, *Wazuh App* juga menyediakan fitur untuk menambahkan agent baru melalui menu *Agent Management*. Fitur ini memungkinkan pengguna untuk memilih sistem operasi *host* yang akan dipantau, kemudian Wazuh akan memberikan skrip instalasi serta konfigurasi *registration key* dan alamat *Wazuh Manager*. Dengan demikian, proses penambahan *host* baru dapat dilakukan secara cepat dan terstandar, serta memastikan seluruh aset yang relevan dapat masuk ke dalam pemantauan HIDS.

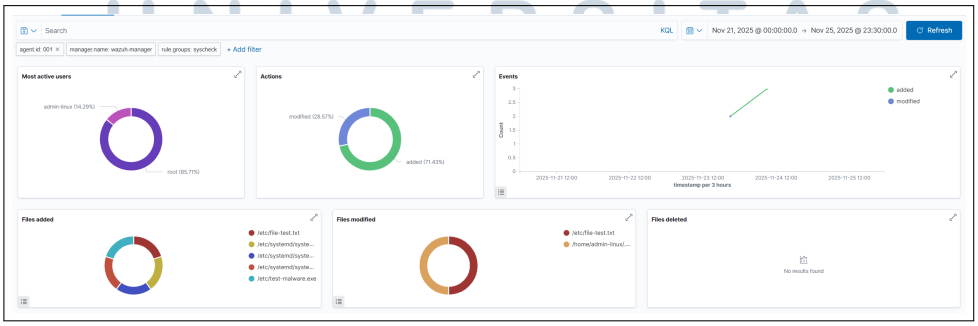
C.1 Security Events



Gambar 3.10. *Tampilan Menu Security Events pada Wazuh App*

Menu *Security Events* pada *Wazuh App* digunakan untuk menampilkan seluruh peristiwa keamanan yang terdeteksi oleh *Wazuh Agent* dan dianalisis oleh *Wazuh Manager*. Seperti ditunjukkan pada Gambar 3.10, halaman ini menyediakan informasi mengenai jenis ancaman, tingkat keparahan, sumber event, serta waktu kejadian. Data pada menu ini digunakan oleh analis SOC untuk melakukan penilaian awal terhadap aktivitas yang mencurigakan, serta menjadi dasar untuk menentukan apakah suatu event perlu ditindaklanjuti sebagai insiden. *Security Events* berperan penting karena menjadi indikator utama apakah terjadi aktivitas abnormal pada *host*.

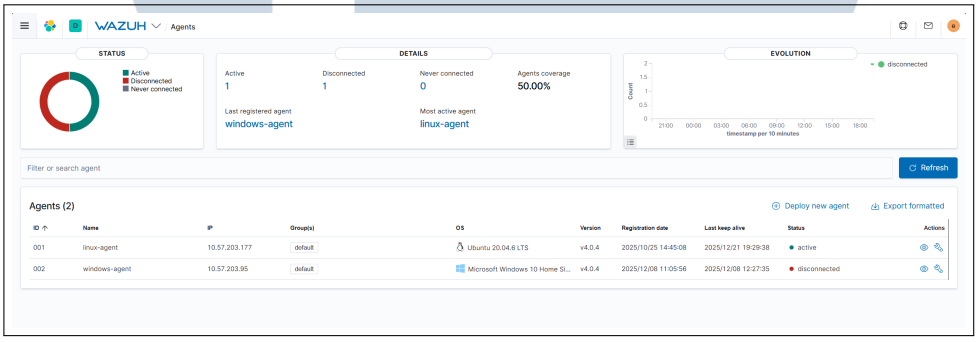
C.2 Integrity Monitoring



Gambar 3.11. *Dashboard File Integrity Monitoring pada Wazuh*

Menu Integrity Monitoring pada Gambar 3.11 menampilkan seluruh aktivitas perubahan file yang terdeteksi melalui mekanisme *File Integrity Monitoring* (FIM). Pada menu ini, pengguna dapat melihat daftar file yang mengalami penambahan, modifikasi, atau penghapusan, lengkap dengan detail seperti path file, hash sebelum dan sesudah perubahan, serta user atau proses yang memicu perubahan tersebut. Informasi ini sangat penting untuk mendeteksi penyimpangan pada sistem, terutama perubahan file yang tidak sah atau yang berpotensi berhubungan dengan aktivitas berbahaya seperti *malware injection* atau modifikasi konfigurasi kritikal.

C.3 Agent Management



Gambar 3.12. *Tampilan Agent Management pada Wazuh App*

Agent Management berfungsi sebagai pusat pengelolaan seluruh agent yang terhubung dengan *Wazuh Manager*. Pada halaman ini, pengguna dapat melihat daftar agent beserta statusnya, melakukan proses pendaftaran agent baru, serta menghapus agent yang tidak lagi digunakan. Seperti terlihat pada Gambar 3.12, fitur ini memberikan visibilitas penuh terhadap semua *host* yang dipantau dan memungkinkan administrator memastikan bahwa setiap agent berfungsi sebagaimana mestinya. Agent Management juga menyediakan informasi detail seperti sistem operasi *host*, alamat IP, versi agent, dan waktu terakhir agent terhubung, yang membantu analis dalam memastikan keandalan pemantauan.

3.3.3 File Integrity Monitoring sebagai Fitur Utama HIDS

File Integrity Monitoring (FIM) merupakan fitur inti dalam sistem *Host Intrusion Detection System* (HIDS) yang berfungsi untuk mendeteksi perubahan pada berkas dan direktori yang memiliki peran penting terhadap keamanan

sistem. Melalui pemantauan berbasis hash, metadata, dan aturan referensi, FIM mampu mengidentifikasi aktivitas penambahan dan modifikasi file secara *real-time*. Perubahan yang terdeteksi dapat menjadi indikasi awal terjadinya kompromi, seperti manipulasi konfigurasi sistem atau penyisipan berkas berbahaya. Oleh karena itu, FIM menjadi komponen penting dalam mendukung proses deteksi dini dan analisis insiden dalam lingkungan *Security Operations Center* (SOC).

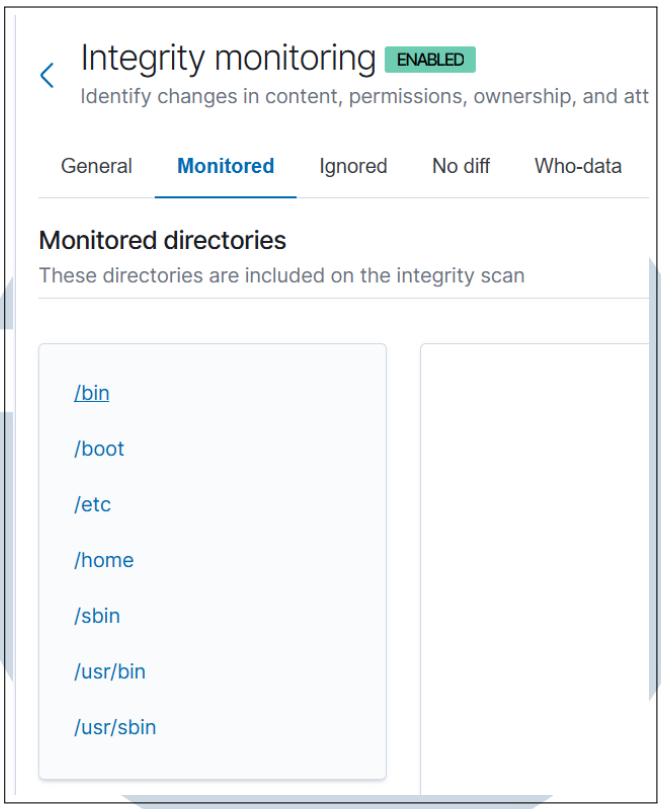
A Konsep dan Peran File Integrity Monitoring

File Integrity Monitoring berperan sebagai mekanisme pemantauan pada tingkat *host* untuk mengidentifikasi perubahan yang berpotensi mengancam keamanan sistem. Setiap perubahan pada file atau direktori yang dipantau dicatat dan dianalisis sebagai *security event*. Informasi ini tidak hanya digunakan untuk mendeteksi indikasi serangan, tetapi juga untuk memverifikasi integritas sistem setelah proses pembaruan atau aktivitas administratif yang sah. Dalam operasional SOC, data FIM dimanfaatkan oleh analis untuk membedakan antara aktivitas normal dan perubahan yang mencurigakan, sehingga membantu proses monitoring harian, investigasi insiden, serta pengambilan keputusan keamanan secara lebih cepat dan terarah.

B Konfigurasi Monitoring File dan Direktori

Konfigurasi *File Integrity Monitoring* (FIM) dilakukan untuk menentukan file dan direktori sistem Linux yang perlu dipantau serta mekanisme pendeteksian perubahan yang diterapkan oleh agen Wazuh. Pada tahap ini, fokus utama adalah memastikan bahwa perubahan terhadap file kritikal dapat terdeteksi secara konsisten dan tercatat sebagai event keamanan.

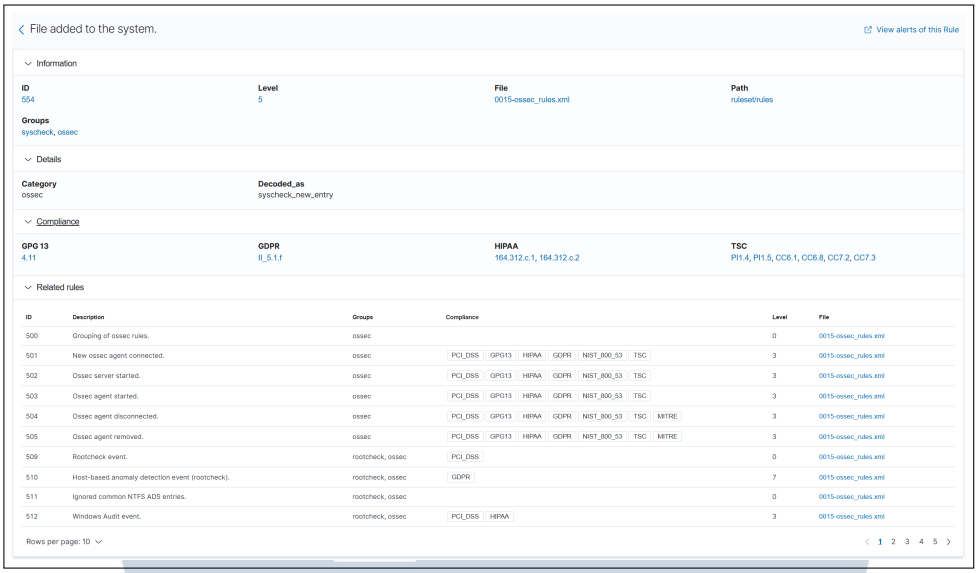
Pada sistem Linux yang dipantau, direktori yang dimasukkan ke dalam cakupan FIM mencakup direktori sistem dan direktori pengguna. Direktori-direktori tersebut meliputi `/bin`, `/boot`, `/etc`, `/home`, `/sbin`, `/usr/bin`, dan `/usr/sbin`, sebagaimana ditunjukkan pada Gambar 3.13. Pemilihan direktori ini didasarkan pada tingkat kritikalitasnya terhadap operasional sistem, di mana perubahan yang tidak sah pada direktori tersebut berpotensi mengindikasikan aktivitas berbahaya atau kesalahan konfigurasi.



Gambar 3.13. Daftar direktori yang dipantau oleh File Integrity Monitoring pada sistem Linux

Selain penentuan direktori, konfigurasi FIM juga melibatkan aturan (*rules*) yang digunakan untuk mengklasifikasikan jenis perubahan file, seperti yang ditunjukkan pada Gambar 3.14. Aturan ini memungkinkan sistem membedakan antara file yang ditambahkan (*added*), diubah (*modified*), atau dihapus (*deleted*). Setiap jenis perubahan memiliki tingkat keparahan (*alert level*) yang berbeda, sehingga analis SOC dapat melakukan prioritisasi terhadap event yang dihasilkan.

Pada Wazuh, perubahan file yang terdeteksi oleh modul *syscheck* akan dipetakan ke dalam *rule* tertentu, seperti `syscheck_new_entry` untuk file baru, `syscheck_integrity_changed` untuk perubahan isi file, serta `syscheck_deleted` untuk file yang dihapus. Informasi ini ditampilkan dalam dashboard Wazuh dalam bentuk detail *rule*, termasuk level, kategori, serta keterkaitannya dengan standar kepatuhan.



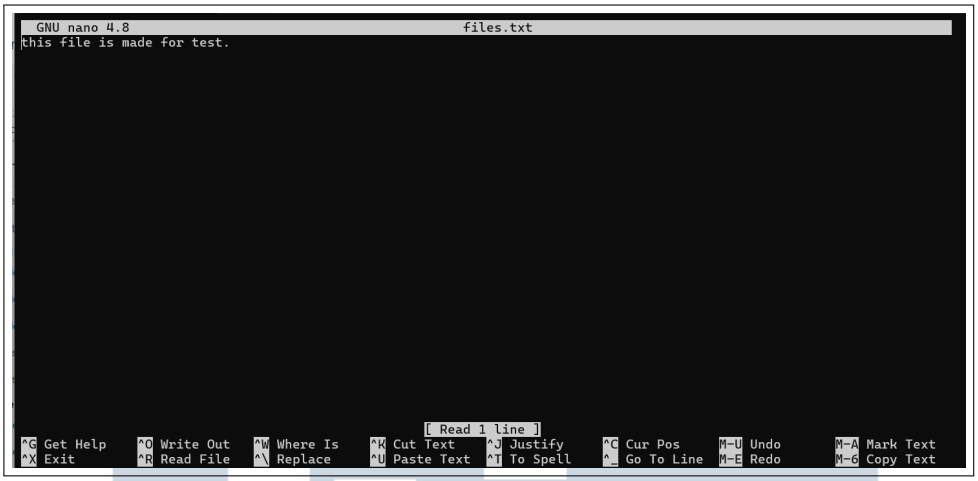
Gambar 3.14. Contoh rule File Integrity Monitoring untuk deteksi file baru pada Wazuh

Konfigurasi monitoring file dan direktori ini menjadi dasar bagi proses deteksi perubahan file pada sistem. Dengan cakupan direktori yang jelas dan aturan deteksi yang terdefinisi, setiap perubahan yang terjadi dapat diproses lebih lanjut dalam alur kerja deteksi dan analisis keamanan pada tahap berikutnya.

C Alur Kerja Deteksi Perubahan File

Alur kerja deteksi perubahan file pada *File Integrity Monitoring* (FIM) merupakan bagian dari modul *syscheck* pada Wazuh HIDS yang berfungsi untuk memantau integritas file pada sistem *host*. Mekanisme ini bekerja dengan membangun kondisi awal (*baseline*) file dan melakukan perbandingan nilai hash untuk mendeteksi adanya perubahan yang terjadi pada direktori yang telah dikonfigurasi.

Untuk memverifikasi mekanisme deteksi perubahan file, dilakukan simulasi perubahan integritas file pada sistem *host* berbasis Linux dengan membuat sebuah file uji pada direktori yang termasuk dalam cakupan pemantauan FIM. Pengujian ini bertujuan untuk memverifikasi bahwa *Wazuh Agent* mampu mendeteksi aktivitas perubahan file sesuai dengan skenario yang dirancang.



Gambar 3.15. *Pembuatan File Uji pada Sistem Linux*

Gambar 3.15 menunjukkan proses pembuatan file uji pada sistem Linux. Karena file tersebut berada pada direktori yang dipantau oleh FIM, aktivitas ini berhasil terdeteksi oleh *Wazuh Agent* dan dicatat sebagai sebuah *security event* dengan tipe *file added*. Hasil ini menunjukkan bahwa mekanisme pemantauan direktori telah berjalan sesuai dengan konfigurasi yang ditetapkan.

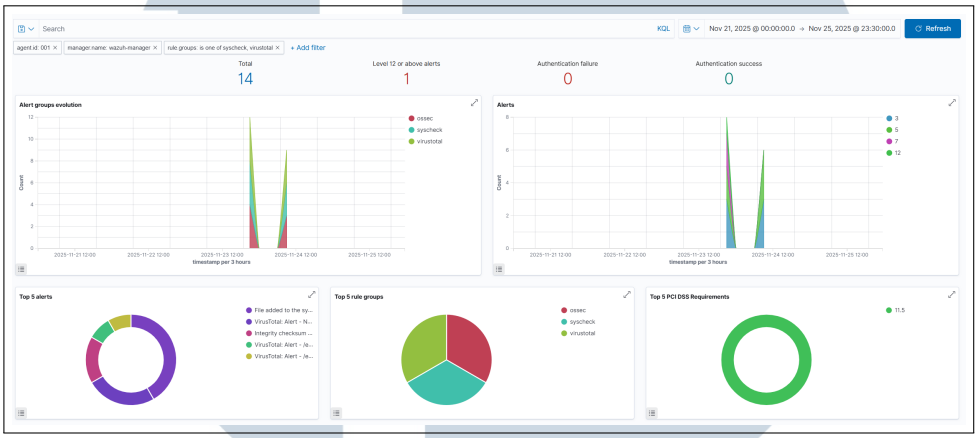
Setelah file berhasil dibuat, dilakukan perubahan terhadap isi file untuk memicu event perubahan integritas.

Gambar 3.16. *Perubahan Konten File Uji*

Pada Gambar 3.16 terlihat perubahan konten file yang mengakibatkan perubahan nilai hash. *Wazuh Agent* mendeteksi kondisi ini melalui mekanisme perbandingan checksum antara kondisi awal dan kondisi terkini file, kemudian menghasilkan event dengan tipe *integrity checksum changed*. Event yang dihasilkan

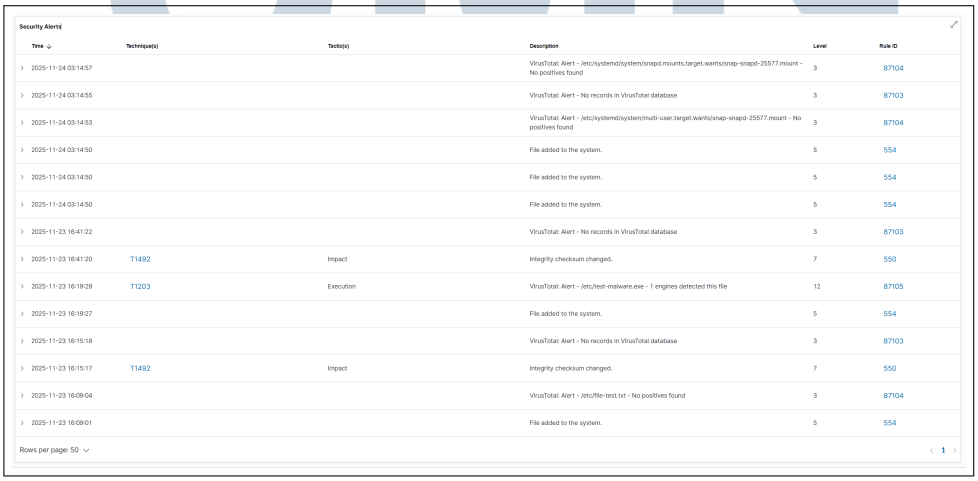
sesuai dengan aktivitas yang dilakukan pada file, sehingga dapat disimpulkan bahwa proses deteksi perubahan integritas berjalan secara akurat.

Event hasil pemantauan FIM selanjutnya dikirimkan ke *Wazuh Manager* untuk diproses berdasarkan aturan (*rule*) yang berlaku. Event tersebut kemudian diteruskan ke Elastic Stack dan disimpan dalam indeks Elasticsearch untuk keperluan visualisasi dan analisis lebih lanjut.



Gambar 3.17. *Dashboard Tren Security Events File Integrity Monitoring*

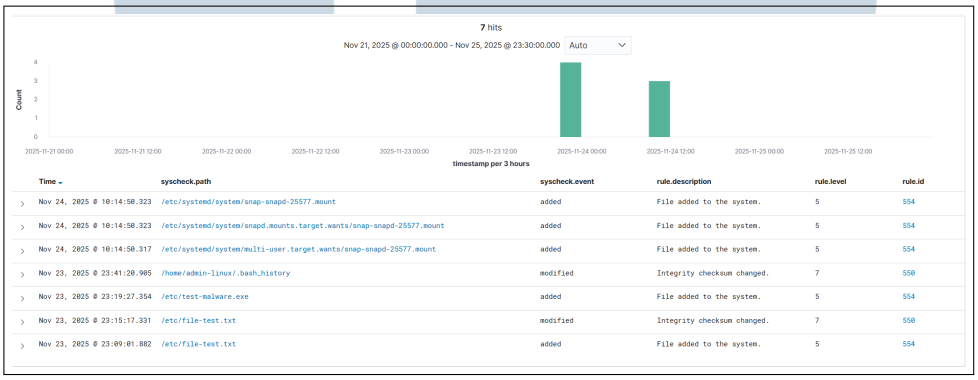
Dashboard pada Gambar 3.17 menampilkan tren *security events* yang dihasilkan dari aktivitas perubahan file selama proses pengujian. Terlihat bahwa event hanya muncul setelah dilakukan simulasi pembuatan dan modifikasi file, tanpa adanya event tambahan di luar skenario pengujian. Hal ini mengindikasikan bahwa konfigurasi *File Integrity Monitoring* telah berjalan dengan baik dan tidak menghasilkan *false alert* pada kondisi normal.



Gambar 3.18. *Daftar Alert File Integrity Monitoring*

Selain visualisasi tren, setiap event perubahan file juga menghasilkan *alert* dengan tingkat keparahan tertentu sesuai dengan *rule* yang terpicu. *Alert* tersebut ditampilkan pada halaman *Security Alerts* di Kibana sebagai indikator awal bagi analis SOC, seperti yang ditunjukan pada Gambar 3.18. Beberapa *alert* yang dihasilkan dari aktivitas pembuatan dan perubahan file, seperti *file added* dan *integrity checksum changed*. Informasi yang ditampilkan mencakup waktu kejadian, deskripsi *rule*, serta level *alert* yang merepresentasikan tingkat urgensi kejadian.

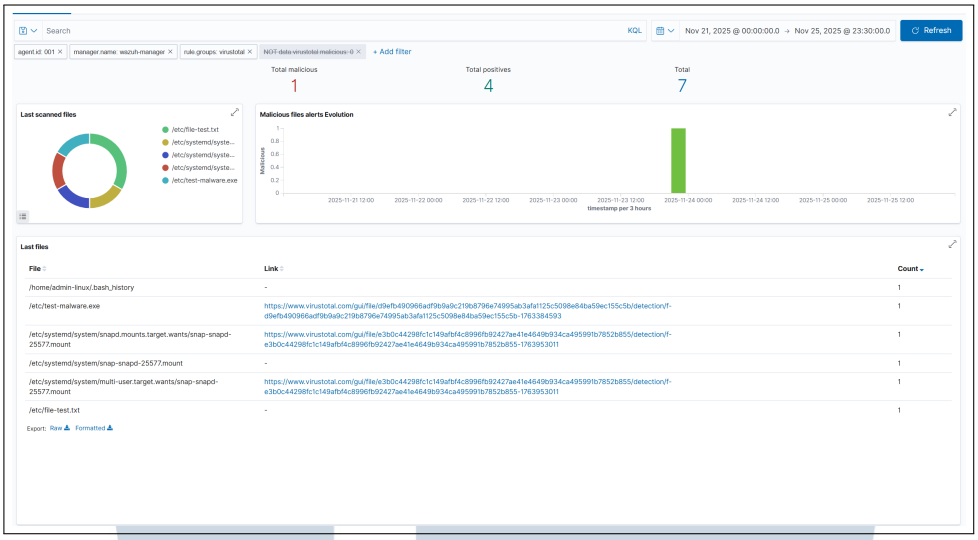
Untuk keperluan analisis yang lebih mendalam, fitur Discover pada Kibana digunakan dengan memfilter event yang berkaitan dengan *File Integrity Monitoring*.



Gambar 3.19. *Tampilan Discover dengan Filter File Integrity Monitoring*

Berdasarkan Gambar 3.19, analis dapat mengamati informasi rinci seperti jalur file (*syscheck.path*), jenis event (*syscheck.event*), serta tingkat keparahan *rule* yang terpicu. Seluruh log yang ditampilkan sesuai dengan aktivitas pengujian yang dilakukan, sehingga dapat digunakan sebagai bukti bahwa mekanisme FIM bekerja secara konsisten.

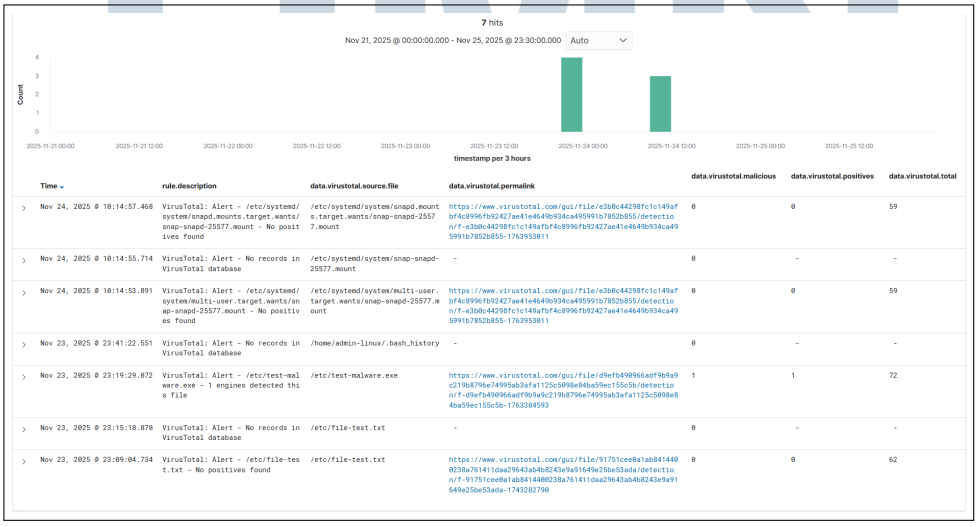
Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa *File Integrity Monitoring* pada Wazuh mampu mendeteksi perubahan file secara akurat dan sesuai dengan skenario simulasi yang dirancang. Seluruh aktivitas pembuatan dan modifikasi file berhasil teridentifikasi dan divisualisasikan secara terintegrasi melalui Elastic Stack. Dengan demikian, pengujian ini tidak hanya menggambarkan alur kerja FIM, tetapi juga memvalidasi efektivitas mekanisme deteksi perubahan file sebagai bagian dari sistem *Host-based Intrusion Detection System* (HIDS).



Gambar 3.20. *Dashboard Tren Alert Integrasi VirusTotal*

Gambar 3.20 menunjukkan dashboard tren *alert* hasil integrasi VirusTotal yang menampilkan jumlah file yang diperiksa serta indikasi file berbahaya dalam rentang waktu tertentu. Visualisasi ini membantu analis SOC dalam mengidentifikasi pola kemunculan file mencurigakan serta mengevaluasi tingkat risiko secara keseluruhan pada sistem yang dipantau.

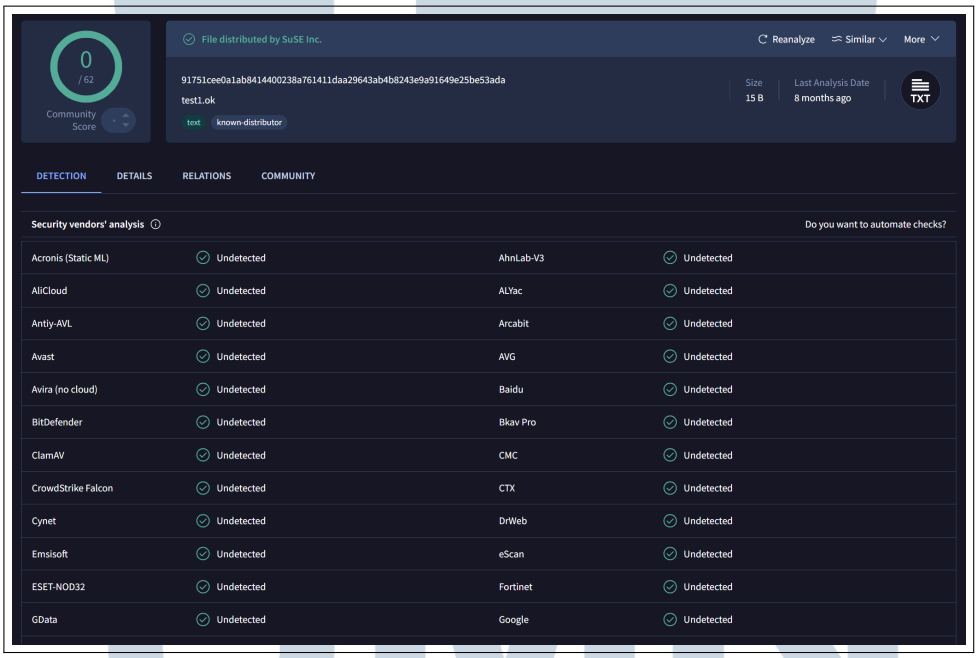
Selain visualisasi tren, hasil integrasi VirusTotal juga dapat dianalisis secara rinci melalui fitur Discover di Kibana. Dengan memfilter event berdasarkan *rule group* VirusTotal, analis dapat menelusuri log mentah yang berisi detail pemeriksaan setiap file.



Gambar 3.21. *Tampilan Discover Event Integrasi VirusTotal*

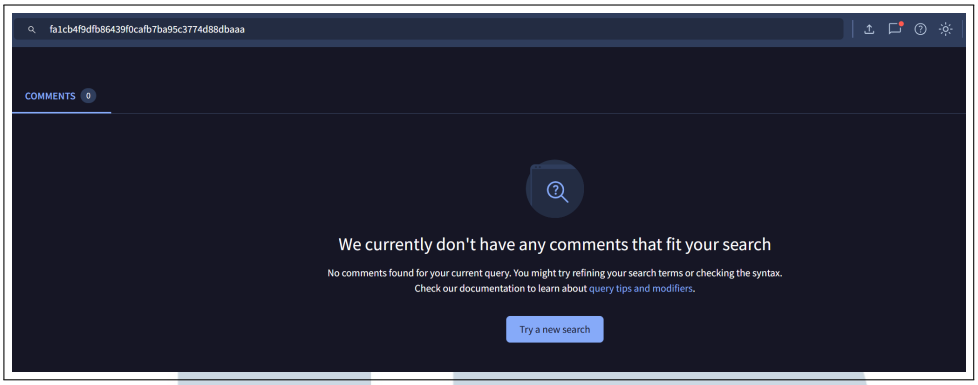
Pada Gambar 3.21, terlihat informasi detail seperti jalur file, nilai hash, jumlah mesin antivirus yang mendeteksi file sebagai berbahaya, serta tautan langsung ke halaman analisis VirusTotal. Informasi ini memungkinkan analis untuk melakukan validasi lanjutan secara cepat tanpa harus berpindah konteks dari sistem SIEM.

Sebagai bagian dari pengujian, dilakukan pula pengecekan hash file secara langsung melalui platform VirusTotal untuk membandingkan hasil yang diperoleh dari integrasi Wazuh. Pada pengujian file uji yang tidak berbahaya, hasil analisis menunjukkan bahwa tidak ada mesin antivirus yang menandai file tersebut sebagai ancaman, seperti yang ditunjukkan pada Gambar 3.22.



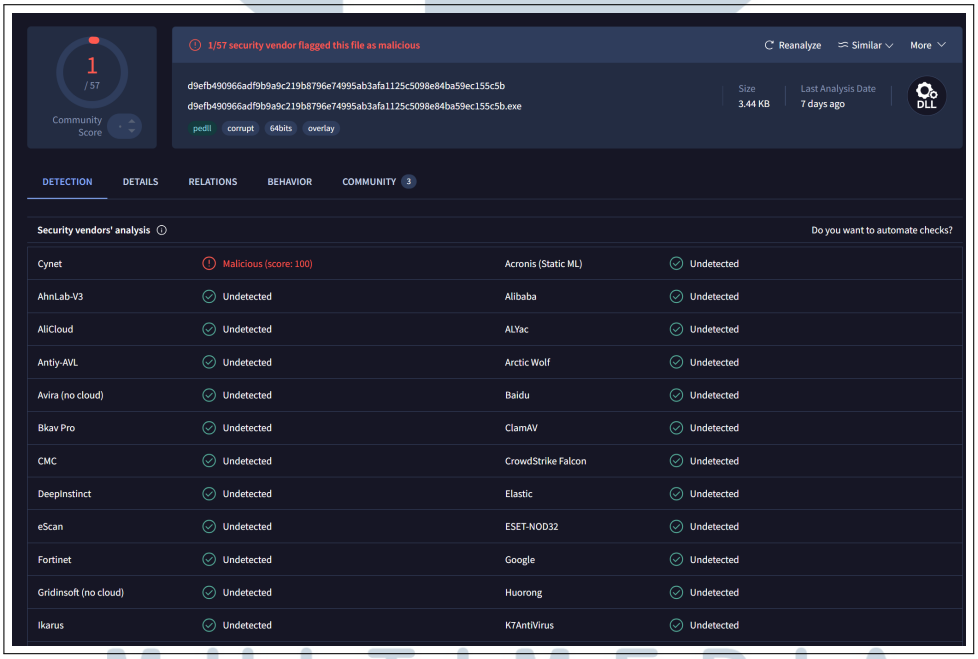
Gambar 3.22. *Hasil Pengecekan Hash File Aman di VirusTotal*

Selain itu, terdapat pula kondisi di mana hash file belum terdaftar dalam basis data VirusTotal. Pada kasus ini, VirusTotal tidak memberikan informasi reputasi karena file tersebut belum pernah dianalisis sebelumnya, seperti yang ditunjukkan pada Gambar 3.23.



Gambar 3.23. *Hasil Pengecekan Hash File yang Tidak Terdaftar di VirusTotal*

Untuk mensimulasikan skenario ancaman, dilakukan pengujian menggunakan sampel file yang terdeteksi berbahaya oleh mesin antivirus pada VirusTotal. Hasil analisis VirusTotal menunjukkan bahwa file tersebut terdeteksi oleh salah satu mesin antivirus sebagai *malicious*, yang kemudian tercermin pada *alert* yang dihasilkan oleh Wazuh, sebagaimana yang ditunjukkan 3.24.



Gambar 3.24. *Hasil Pengecekan Hash File Berbahaya di VirusTotal*

Berdasarkan hasil pengujian yang dilakukan, dapat disimpulkan bahwa integrasi VirusTotal pada Wazuh berhasil memberikan konteks tambahan terhadap event perubahan file yang terdeteksi oleh *File Integrity Monitoring*. Setiap file yang terpantau FIM dapat divalidasi reputasinya melalui mekanisme *hash lookup*,

sehingga analis SOC memperoleh informasi tambahan mengenai status keamanan file tersebut.

Hasil pengujian menunjukkan bahwa sistem mampu membedakan beberapa kondisi, yaitu file yang teridentifikasi aman, file yang belum memiliki rekam jejak pada basis data VirusTotal, serta file yang terindikasi berbahaya oleh mesin antivirus tertentu. Informasi ini ditampilkan secara terintegrasi melalui dashboard dan fitur Discover pada Kibana, sehingga mendukung proses analisis dan pengambilan keputusan secara efisien.

Dengan demikian, integrasi VirusTotal berperan sebagai lapisan *threat intelligence enrichment* yang melengkapi mekanisme deteksi perubahan file pada Wazuh HIDS. Kombinasi antara *File Integrity Monitoring* dan validasi reputasi file melalui VirusTotal memungkinkan sistem tidak hanya mendeteksi perubahan pada level teknis, tetapi juga memberikan penilaian awal terhadap potensi ancaman, sehingga mendukung peningkatan efektivitas proses monitoring keamanan serta respons awal terhadap insiden.

3.4 Kendala dan Solusi yang Ditemukan

Selama proses kegiatan magang yang berfokus pada integrasi *File Integrity Monitoring* (FIM) menggunakan Wazuh HIDS dan Elastic Stack, terdapat beberapa kendala yang ditemui dalam proses pengujian, pemantauan, dan analisis data keamanan. Kendala-kendala tersebut antara lain sebagai berikut:

1. Cakupan direktori dan file yang dipantau oleh *File Integrity Monitoring* tidak dapat dibuat terlalu luas karena berpotensi menimbulkan beban kinerja pada sistem *host* serta menghasilkan volume log yang berlebihan.
2. *File Integrity Monitoring* mendeteksi seluruh perubahan file tanpa membedakan apakah perubahan tersebut merupakan aktivitas yang sah atau indikasi ancaman, sehingga berpotensi memunculkan *alert* terhadap perubahan normal sistem.
3. Integrasi VirusTotal memiliki keterbatasan dalam menyediakan informasi reputasi file, terutama ketika hash file belum terdaftar dalam basis data VirusTotal atau ketika layanan berada pada batas kuota permintaan.

Berikut merupakan solusi yang diterapkan dan direkomendasikan untuk mengatasi kendala-kendala tersebut selama proses kegiatan magang:

1. Pemantauan *File Integrity Monitoring* difokuskan pada direktori-direktori kritis yang berpengaruh langsung terhadap keamanan sistem, seperti direktori konfigurasi dan file layanan penting, guna menjaga keseimbangan antara efektivitas deteksi dan performa sistem.
2. Penyesuaian konfigurasi dan analisis lanjutan dilakukan dengan memperhatikan konteks perubahan file, waktu kejadian, serta korelasi dengan event keamanan lainnya, sehingga analis dapat membedakan antara aktivitas normal dan potensi ancaman.
3. Hasil integrasi VirusTotal digunakan sebagai informasi pendukung dalam proses analisis, bukan sebagai satu-satunya acuan pengambilan keputusan. Ketika data reputasi tidak tersedia, analisis tetap dilakukan berdasarkan indikator teknis dari *File Integrity Monitoring*.