Vol. 2 No. 2 – Tahun 2023
Indonesian Journal Computer Science
ISSN: 2829-3819

Perancangan Fitur Audit *Security Configuration Compliance* Pada Aplikasi Helium *Security*

Erik Andri Budiman[1], Girindro Pringgo Digdo[2]
STMIK AMIKBANDUNG [1,2]
erik.andri10@gmail.com[1], pringgo@stmik-amikbandung.ac.id[2]

Diterima	Direvisi	Disetujui
(24-09-2023)	(29-09-2023)	(10-10-2023)

Abstrak - Hasil penelitian yang dilakukan di PT Global Inovasi Siber Indonesia menunjukkan bahwa layanan penyedia jasa penetrasi untuk teknologi komputer di internet masih terbatas. Layanan ini umumnya berfokus pada pengujian umum untuk mendeteksi celah atau kerentanan dalam layanan *website* dengan melakukan penetrasi ke sisi server atau *back-end* melalui API (*Application Programming Interface*). Seiring dengan perkembangan teknologi, sering kali terjadi masalah di bagian jaringan karena kesalahan konfigurasi layanan di sisi server, yang dapat membuka peluang serangan pengaksesan tanpa izin ke dalam server tersebut. Dalam konteks ini, aplikasi Helium *Security* merancang fitur baru yang dapat menjadi solusi untuk masalah tersebut. Fitur ini dirancang untuk melakukan audit kepatuhan standar di sisi server dengan cara melakukan penetrasi ke sisi server melalui jaringannya untuk mengidentifikasi standar keamanan yang tidak dipatuhi oleh organisasi. Aplikasi ini memberikan gambaran yang lebih akurat tentang paparan kerentanan, lokasi dan penyebab potensial pengaksesan yang tidak sah, serta memberikan saran untuk menutup celah tersebut. Penelitian ini menggunakan metodologi *Agile* dengan model *Scrum*. Fitur audit *security configuration* pada aplikasi Helium *Security* dirancang menggunakan pemodelan UML dan diimplementasikan menggunakan bahasa pemrograman *Python*. Setelah uji coba, fitur audit *security configuration compliance* telah terbukti efektif, membantu mengidentifikasi kerentanan terhadap ketidakpatuhan standar keamanan teknologi. Berdasarkan hasil survei, sebanyak 91,3% responden menyatakan sangat setuju, sementara 8,7% setuju dengan keberhasilan fitur ini. Ini memberikan gambaran lengkap tentang informasi dan celah yang ada pada layanan dalam jaringan komputer, memungkinkan teknisi di balik server untuk segera mengatasi ancaman pengaksesan tanpa izin yang dapat terjadi kapan saja.

Kata Kunci : Teknologi, Audit *Security Configuration Compliance*, Helium *Security*, Server, Jaringan

Abstract *- The research conducted at PT Global Inovasi Siber Indonesia highlights the limited availability of penetration testing services for computer technology on the internet. These services primarily focus on general testing to detect vulnerabilities in website services by penetrating the server or back-end through the Application Programming Interface (API). With technology advancements, network problems often arise due to misconfigurations on the server side, creating opportunities for unauthorized access to the server. In response, the Helium Security application designed a new feature to address these challenges. This feature was crafted to conduct compliance standard audits on the server side by penetrating the server's network, identifying security standards non-compliance within organizations. The application provides a more precise overview of vulnerability exposures, potential unauthorized access points, their locations, causes, and suggestions to rectify these gaps. The study utilized Agile methodology with the Scrum model. The security configuration audit feature in the Helium Security application was designed using UML modeling and implemented using the Python programming language. Post-testing, the audit security configuration compliance feature proved highly effective, aiding in identifying vulnerabilities related to non-compliance with technology security standards. Survey results indicated strong agreement, with 91.3% of respondents strongly agreeing and 8.7% agreeing with the feature's success. This comprehensive insight into information and vulnerabilities within computer network services enables technicians behind the servers to promptly address potential unauthorized access threats that can occur at any time.*

Keywords: Technology, Security Configuration Compliance Audit, Helium Security, Server, Network

I. PENDAHULUAN

Helium *Security* merupakan platform yang digunakan untuk melakukan penetrasi testing guna mengetahui kerentanan suatu sistem sehingga dapat mengurangi risiko ancaman sedini mungkin. Helium *Security* mempunyai banyak perangkat lunak yang mendukung penilaian kerentanan suatu aplikasi hingga

mendapatkan hasil rekap yang disusun dengan baik. Saat ini Helium *Security* akan menambahkan fitur baru untuk melakukan Audit *Security Configuration Compliance* guna melakukan pemindaian terhadap suatu sistem tentang kepatuhan terhadap standar keamanan teknologi dunia.

Lingkungan bisnis yang berkembang saat ini semakin bergantung pada teknologi demi kemudahan akses dan pengelolaan data sehingga banyak menggunakan sistem yang disediakan namun tidak sedikit yang tahu banyak soal keamanan terhadap sistem yang mereka bangun. Bisnis yang memanfaatkan teknologi menyimpan banyak informasi sensitif, hal tersebut menjadi ancaman terhadap kerahasiaan data dan integritas data. Bisnis di era digital perlu menaruh perhatian tinggi untuk dapat melindungi aset bisnis mereka dari target serangan oleh penjahat siber. Salah satu cara untuk memastikan keamanan sistem adalah dengan mematuhi standar keamanan yang ditetapkan untuk industri, sebuah studi pada tahun 2014 melaporkan bahwa 90 persen lembaga keuangan menggunakan satu atau lebih standar keamanan seperti NIST, ISACA, ISO 27001, dan ISO 27002 (Carter & Zheng, 2015). Standar keamanan ini memberikan panduan tentang praktik keamanan yang harus diikuti oleh organisasi untuk melindungi aset bisnis mereka.

Namun, mengikuti standar keamanan ini bukanlah tugas yang mudah. Banyak organisasi yang kesulitan dalam memenuhi persyaratan keamanan yang ditetapkan oleh standar tersebut. Hal ini dapat terjadi karena berbagai faktor, seperti perubahan teknologi, ketidakmampuan untuk mengelola dan memperbarui sistem keamanan, atau kurangnya sumber daya manusia yang terlatih dalam bidang keamanan siber.

Melalui perancangan fitur untuk melakukan pemindaian atau audit *security configuration compliance* pada aplikasi Helium *Security*, organisasi dapat memperoleh alat yang efektif dan efisien untuk memeriksa kepatuhan sistem mereka dengan standar keamanan. Hal ini dapat membantu organisasi untuk meningkatkan keamanan jaringan dan sistem mereka tanpa harus memiliki kemampuan dan pemahaman dalam audit infrastruktur sistem agar dapat terus memantau dan memeriksa layanan sesuai standar teknologi informasi yang terbaru, sehingga dapat melindungi aset bisnis mereka dari serangan siber yang merugikan.

1. Tinjauan Pustaka

1. Audit *Security Configuration Compliance*

   Audit *Security Configuration Compliance* adalah suatu proses pemindaian terhadap suatu sistem yang berfokus pada konfigurasi yang diterapkan pada sistem untuk memenuhi aturan standar kepatuhan dari suatu kerangka kerja. Singkatnya audit *security configuration compliance* akan menilai kepatuhan standar keamanan yang diterapkan pada sistem. Audit *security configuration compliance* memiliki cara kerja hampir sama dengan *vulnerability scan*, namun audit *security configuration compliance* hanya berfokus untuk meningkatkan konfigurasi sistem berdasarkan standar keamanan yang dibutuhkan saat pengecekan dilakukan. Peningkatan ini terdiri dari penerapan panduan keamanan dari berbagai *framework compliance*. Misalnya suatu perusahaan harus menerapkan standar keamanan dari HIPAA dan/atau HITRUST dikarenakan sistem dirancang untuk kebutuhan sistem pelayanan kesehatan, maka sistem harus dikonfigurasi untuk memenuhi kebutuhan kepatuhan tersebut (Moran - Director Of Services, 2017).
2. IT Audit

   IT Audit secara umum diartikan sebagai pemeriksaan dan evaluasi terhadap infrastruktur teknologi informasi dari suatu organisasi. Audit pada infrastruktur IT dapat menentukan apakah kontrol TI pada suatu organisasi dapat melindungi aset perusahaan, memastikan integritas data, dan selaras dengan keseluruhan bisnis. Operasi di perusahaan modern saat ini semakin terkomputerisasi, sehingga IT Audit digunakan untuk memastikan kontrol dan proses terkait informasi berjalan dengan baik (Cole, 2014).
3. Standar Keamanan Siber

   Standar keamanan siber adalah seperangkat pedoman atau praktik terbaik yang dapat digunakan organisasi untuk meningkatkan postur keamanan siber mereka. Organisasi dapat menggunakan standar keamanan siber untuk membantu mereka mengidentifikasi dan menerapkan tindakan yang tepat untuk melindungi sistem dan data mereka dari ancaman siber. Standar juga dapat memberikan panduan tentang cara menanggapi dan memulihkan dari insiden keamanan siber (*Cybersecurity Standards and Frameworks | IT Governance USA*, 2017).

   Beberapa standar keamanan siber antara lain:

   a. ISO 27001. Standar internasional untuk keamanan informasi yang menyediakan kerangka kerja untuk mengelola informasi sensitif perusahaan. Standar tersebut mencakup persyaratan untuk mengembangkan ISMS (sistem manajemen keamanan informasi), menerapkan kontrol keamanan, dan

melakukan penilaian risiko (*ISO 27001, the Information Security Standard | IT Governance USA*, 2022).

b. ISO 27002 adalah kode praktik untuk manajemen keamanan informasi. Ini memberikan panduan dan rekomendasi tentang cara menerapkan kontrol keamanan dalam suatu organisasi. ISO 27002 mendukung standar ISO 27001, yang menyediakan persyaratan untuk SMKI (*ISO 27002: Security Controls*, 2022).

c. ISACA merujuk pada kerangka kerja yang dikembangkan oleh Information Systems Audit and Control Association (ISACA) untuk memastikan keamanan informasi dan manajemen risiko dalam organisasi. Standar ini mencakup pedoman dan praktik terbaik dalam mengelola keamanan informasi, termasuk pengelolaan akses, pengendalian keamanan fisik, pengelolaan risiko, dan pengelolaan kebijakan keamanan. Standar ini dirancang untuk membantu organisasi melindungi informasi sensitif, menjaga keberlanjutan operasi, dan mematuhi regulasi yang berlaku. ISACA menyediakan beberapa panduan kerangka kerja, standar keamanan, dan model (*Frameworks, Standards and Models | ISACA*, 2008) seperti yang dilansir dalam laman resminya diantara-Nya:
   1) COBIT sebagai sumber daya, dan panduan untuk tata kelola dan manajemen TI perusahaan.
   2) *Risk* IT *Framework* yang memberikan pandangan komprehensif dan *end-to-end* mengenai risiko yang terkait dengan penggunaan TI dan perlakuan menyeluruh terhadap manajemen risiko.
   3) IT Audit *Framework* (ITAF) yang memberikan panduan dan teknik yang akan memberikan konsistensi dan efektivitas pada audit. ITAF edisi ke-4 yang baru menguraikan standar dan praktik terbaik yang selaras dengan urutan proses audit (penilaian risiko, perencanaan, dan kerja lapangan) untuk memandu dalam menilai efektivitas operasional suatu perusahaan dan dalam memastikan kepatuhan.
   4) BMIS (*Business Model for Information Systems*) sebagai model Bisnis Keamanan Informasi, memberikan penjelasan mendalam tentang model bisnis holistik yang mengkaji masalah keamanan dari perspektif sistem.

d. NIST CSF (*National Institute of Standards and Technology Cybersecurity Framework*) adalah kerangka kerja sukarela yang menyediakan serangkaian standar, panduan, dan praktik terbaik untuk mengelola risiko keamanan siber. Kerangka kerja ini membantu organisasi untuk mengidentifikasi, menilai, dan mengelola risiko keamanan siber mereka dengan cara yang terstruktur dan dapat diulang. Kerangka ini tidak wajib, tetapi semakin banyak diadopsi oleh organisasi sebagai tindakan sukarela untuk meningkatkan postur keamanan siber mereka (*What is the NIST Cybersecurity Framework (CSF)? | IT Governance USA*, 2022).

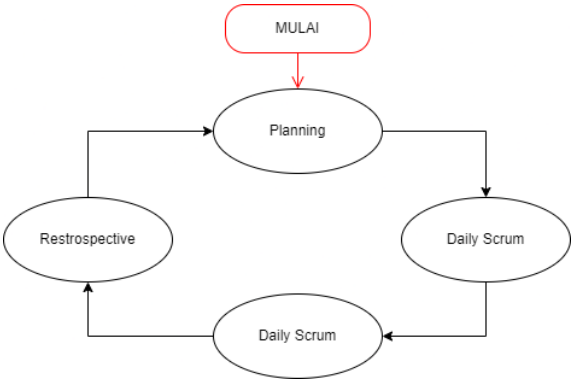
4. Greenbone OpenVAS
   OpenVAS adalah pemindai kerentanan yang kemampuannya mencakup pengujian yang tidak diautentikasi dan diautentikasi, berbagai protokol internet dan industri tingkat tinggi dan rendah, penyesuaian kinerja untuk pemindaian skala besar dan bahasa pemrograman internal yang kuat untuk mengimplementasikan semua jenis uji kerentanan. OpenVAS telah dikembangkan dan didorong maju oleh perusahaan Greenbone sejak tahun 2006. Sebagai bagian dari keluarga produk manajemen kerentanan komersial Greenbone *Enterprise Appliance*, pemindai membentuk Edisi Komunitas Greenbone bersama dengan modul sumber terbuka lainnya (*OpenVAS - Open Vulnerability Assessment Scanner*, 2021).
5. API (*Application Programming Interface*)
   API yang merupakan singkatan dari *application programming interface* adalah sekumpulan protokol yang memungkinkan berbagai komponen perangkat lunak untuk berkomunikasi dan mentransfer data. Pengembang menggunakan API untuk menjembatani kesenjangan antara potongan kode yang kecil dan terpisah untuk menciptakan aplikasi yang kuat, tangguh, aman, dan mampu memenuhi kebutuhan pengguna (*What is an API? A Beginner's Guide to APIs | Postman*, 2023).
6. Helium *Security*
   Helium *Security* adalah sebuah platform *vulnerability assessment* dan *penetration testing* untuk mengurangi risiko di semua jenis aplikasi. Helium dibangun dengan visi untuk membangun platform penilaian kerentanan yang paling fleksibel dan mudah

digunakan dengan memberdayakan Anda untuk mengurangi risiko di semua jenis aplikasi. Helium mempunyai banyak perangkat lunak yang mendukung penilaian kerentanan suatu aplikasi hingga mendapatkan hasil rekap yang disusun dengan baik dengan menggabungkan sejumlah jenis alat yang berbeda, Helium akan memberikan gambaran yang lebih akurat tentang paparan kerentanan suatu jenis aplikasi (*About Us | Helium Security*, 2022).

7. Python
   Python adalah bahasa pemrograman komputer tingkat tinggi, python dikatakan sebagai bahasa tingkat tinggi dikarenakan bahasanya yang mendekati bahasa manusia (bahasa inggris), dilansir dalam laman resmi Python bahwa bahasa pemrograman Python berorientasi objek, dengan semantik dinamis. Struktur data bawaan tingkat tinggi, dikombinasikan dengan pengetikan dinamis dan pengikatan dinamis, membuatnya sangat menarik untuk *Rapid Application Development* (Pengembangan Aplikasi Cepat), serta untuk digunakan sebagai bahasa skrip atau lem untuk menghubungkan komponen yang ada bersama-sama. Sintak Python yang sederhana dan mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program (*What is Python? Executive Summary | Python.org*, 2022).
8. CLI (*Command Line Interface*)
   CLI (*Command Line Interface*) adalah antarmuka yang digunakan untuk berinteraksi dengan sistem operasi atau perangkat lunak melalui perintah teks. Bedanya dengan antarmuka grafis adalah CLI hanya menggunakan teks yang diketikkan oleh pengguna. Dengan CLI, pengguna dapat mengetikkan perintah untuk menjalankan berbagai operasi dalam sistem operasi atau perangkat lunak, seperti menyalin, mengubah nama, memotong, menghapus, dan sebagainya (Sari, 2023).
9. Python Flask
   Flask adalah *framework* web ringan yang ditulis dengan Python. Berdasarkan Werkzeug WSGI (Python *Web Server Gateway Interface* (WSGI) adalah antarmuka antara aplikasi atau kerangka kerja Python dan server web. Telah diterima secara luas, dan pada dasarnya telah mencapai portabilitasnya. Flask menggunakan otorisasi BSD. Flask disebut juga "*micro framework*" karena menggunakan inti sederhana dan menggunakan ekstensi untuk menambahkan fungsi lainnya. Flask tidak memiliki *database default* atau alat verifikasi formulir. Namun, Flask tetap mempertahankan fleksibilitas perluasan, dan ekstensi Flask dapat digunakan untuk menambahkan fungsi berikut: ORM, verifikasi *form*, pengunggahan *file*, dan berbagai teknologi autentikasi terbuka (*Introduction to the Python Flask framework - Alibaba Cloud Community*, 2022).

II. METODOLOGI PENELITIAN

Sebelum perancangan sistem audit *compliace*, Penulis melakukan penelitian terkait kebutuhan perangkat lunak untuk dapat merancang sistem dengan baik. Penulis menggunakan metode pengembangan *Agile* dalam menentukan kebutuhan pengembangan, dan Penulis menggunakan model metodologi *Scrum* yang berasal dari *Agile*. *Scrum* membagi pengembangan perangkat lunak menjadi iterasi singkat (sprint) yang biasanya berlangsung selama 1-4 minggu (Panatagama, 2023) dengan menggunakan aplikasi Trello untuk membantu pembuatan dan pemantauan pengembangan sebagai *Scrum* Master, dan setiap sprint memiliki tujuan dan rencana kerja yang jelas.



Sumber: Penelitian (2023)

Gambar 1. *Flowchart* metodologi *Scrum*

Adapun tahapan-tahapan metode perancangan *agile scrum* adalah sebagai berikut:

1. *Planning*: Tim *Scrum* melakukan pertemuan perencanaan sprint (*Sprint Planning*) untuk menentukan tujuan sprint, mengidentifikasi pekerjaan yang harus dilakukan dalam sprint tersebut, dan merencanakan bagaimana pekerjaan tersebut akan diselesaikan. Pada tahapan ini, Penulis membuat *card* baru di Trello pada bagian *TO-DO List* dan membuatkan *checklist* di dalamnya untuk setiap arsitektur sistem seperti Helium API, OpenVAS API dan lain sebagainya, di masing-masing arsitektur sistem tersebut Penulis menuliskan proses yang harus dikerjakan secara bertahap hingga selesai sesuai dengan kebutuhan.
2. *Daily Scrum*: Setiap hari, selama sprint berlangsung, tim *Scrum* melakukan

pertemuan harian singkat (*Daily Scrum*) untuk memperbarui progres pekerjaan dan mengidentifikasi hambatan yang muncul. Pada tahapan ini, Penulis akan melakukan pertemuan singkat membahas pengerjaan dari sprint, dalam tahap ini juga pengembangan fitur akan semakin terarah sesuai dengan kebutuhan karena bisa semakin memahami proses sprint dan kebutuhan aplikasi lebih dalam dan akurat.

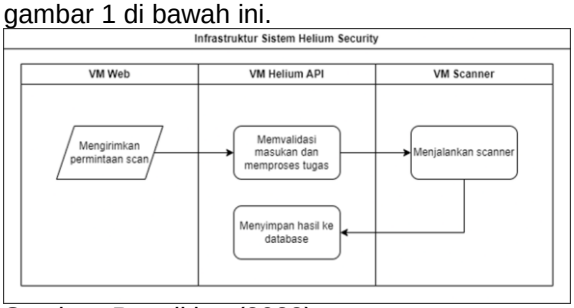
3. *Sprint Review*: Setelah sprint selesai, tim *Scrum* melakukan pertemuan *review sprint* (*Sprint Review*) untuk meninjau hasil pekerjaan dan memperoleh umpan balik dari *stakeholders*. Pada tahapan ini, Penulis akan melakukan pertemuan dengan tim *back-end engineer* dan *chief operational officer* membahas tahapan pengembangan yang sudah dikerjakan, dan dalam tahapan ini juga terjadi uji coba secara langsung di *server staging* oleh tim *back-end engineer* dan *chief operational officer* untuk memastikan fitur berjalan sesuai kebutuhan dan siap untuk dioperasikan.
4. *Sprint Retrospective*: Setelah pertemuan *Sprint Review*, tim *Scrum* melakukan pertemuan retrospeksi *sprint* (*Sprint Retrospective*) untuk mengevaluasi proses kerja dan menemukan cara untuk meningkatkan kinerja tim pada sprint berikutnya. Pada tahapan ini, Penulis akan melakukan pertemuan terakhir membahas tentang fitur yang sudah selesai dikembangkan untuk memastikan apakah ada pengembangan lebih lanjut, dalam tahap ini fitur yang dibuat sudah dapat dikonfirmasi untuk siap digunakan pengguna dan melakukan *deployment* ke server *production*.

III. HASIL DAN PEMBAHASAN

1. Analisis Sistem Berjalan

Berdasarkan hasil wawancara yang telah dilakukan, teridentifikasi bahwa platform Helium *Security* membutuhkan peralatan baru untuk mendukung proses audit tentang konfigurasi sistem yang diterapkan, peralatan audit ini bekerja dengan memanfaatkan kebijakan standar keamanan teknologi informasi yang sudah ada, sistem akan melakukan *vulnerability assessment* dengan melakukan simulasi pengecekan suatu sistem dengan berbagai metode seperti memeriksa *file* konfigurasi, melakukan autentikasi dengan kredensial *default*, memeriksa versi aplikasi dan lain sebagainya.

Pengguna platform Helium *Security* dapat memanfaatkan fitur audit *security configuration compliance* dengan cara mengirimkan tugas audit terhadap target yang sudah ditambahkan. Helium *Security* akan menjalankan tugas fungsional yang diminta oleh pengguna di latar belakang dengan proses komunikasi antar infrastruktur sistem seperti yang terlihat di gambar 1 di bawah ini.



Sumber: Penelitian (2023)

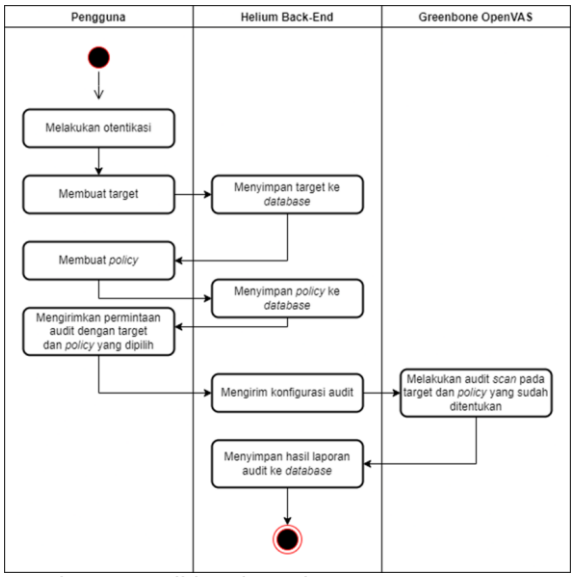
Gambar 2. *Flowchart* infrastruktur Helium *Security*

2. Analisis Sistem Rekomendasi

Sistem aplikasi yang berjalan di platform Helium *Security* melibatkan banyak infrastruktur untuk melakukan komunikasi data dengan berbagai perangkat lunak, terutama untuk menjalankan aplikasi *scanner* yang dinilai berat. Adapun usulan infrastruktur sistem untuk fitur audit *security configuration compliance* menggunakan server terpisah dengan spesifikasi yang sudah disetujui bersama.

Dengan sistem ini, pengguna dapat memanfaatkan proses audit *security configuration compliance* yang disediakan platform Helium *Security* yang dapat berjalan secara otomatis, mudah dan cepat sehingga dapat mempersingkat waktu dalam melakukan audit dan tanpa perlu pengetahuan mendalam tentang proses audit. Sistem platform Helium *Security* akan menjalankan audit di latar belakang di server terpisah dengan spesifikasi yang cukup tinggi untuk menghemat waktu.

Berikut adalah bisnis proses yang dirancang untuk infrastruktur Helium *Security* dalam menjalankan fitur audit *security configuration compliance*, rancangan proses bisnis tersebut dituangkan dalam bentuk diagram pada gambar 2 di bawah ini:



Sumber: Penelitian (2023)

Gambar 3. Proses bisnis audit *security configuration compliance*

3. Analisis Kebutuhan Sistem

Dalam analisis kebutuhan sistem, Penulis menitikberatkan pada perangkat lunak yang sedang dikembangkan, mulai dari merancang sistem desain yang sesuai dengan kebutuhan pengguna.

a. Kebutuhan Fungsional

Kebutuhan fungsional perangkat lunak dalam perancangan fitur audit security configuration compliance pada aplikasi Helium Security mendeskripsikan layanan, fitur atau fungsi yang disediakan atau diberikan oleh sistem bagi penggunanya. Kebutuhan fungsional adalah kriteria atau fitur yang harus ada dalam sebuah perangkat lunak agar dapat memenuhi kebutuhan pengguna atau pemilik sistem. Berikut tabel 1 menjelaskan kebutuhan fungsional dari fitur audit security configuration compliance.

Tabel 1. Kebutuhan fungsional

No. KF	Deskripsi
KF – 01	*Input* target URL
KF – 02	Lihat target URL
KF – 03	Hapus target URL
KF – 04	*Input* standar kepatuhan keamanan
KF – 05	Lihat standar kepatuhan keamanan
KF – 06	Hapus standar kepatuhan keamanan
KF – 07	*Scan* audit kepatuhan
KF – 08	Lihat hasil audit kepatuhan

Sumber: Penelitian (2023)

b. Kebutuhan Non-fungsional

Kebutuhan non-fungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan non fungsional juga sering disebut sebagai batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses, standarisasi dan lain-lain. Pada tabel 2 di bawah ini, Penulis akan menjabarkan kebutuhan non fungsional dari fitur audit *security configuration compliance*.

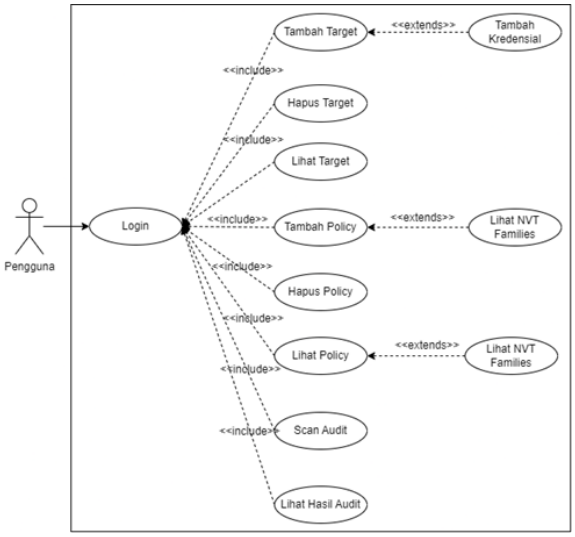
Tabel 2 Kebutuhan non-fungsional

No. KNF	Deskripsi
KNF – 01	Fitur dirancang dalam bahasa pemrograman Python sehingga dapat diakses lintas platform
KNF – 02	Sistem dapat dijalankan lintas platform yang mendukung perangkat lunak Python seperti Windows, Mac, dan Linux (Ubuntu, Debian)
KNF – 03	Penyimpanan data hasil audit *security configuration compliance* tersimpan dalam komputer di mana aplikasi dijalankan
KNF – 04	Fitur dijalankan dalam antarmuka CLI menggunakan terminal komputer

Sumber: Penelitian (2023)

4. *Use Case* Diagram

Pada gambar 3 di bawah ini akan dijelaskan mengenai *use case* secara keseluruhan, di mana *use case* bertujuan mengidentifikasi bagaimana aktor dapat berinteraksi dengan sistem dan apa yang dapat dilakukan oleh sistem.

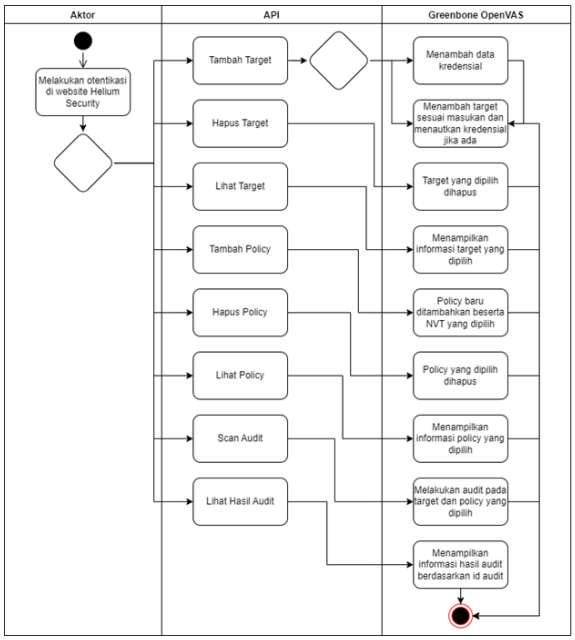


Sumber: Penelitian (2023)

Gambar 4. *Use case* diagram

5. *Activity* Diagram

Activity Diagram adalah representasi visual dari alur aktivitas atau alur kerja yang terjadi dalam sistem yang sedang berjalan. Diagram ini menggambarkan berbagai proses yang terjadi dalam sistem tersebut. Urutan proses dalam sistem tersebut diilustrasikan secara vertikal. Pada gambar 4 di bawah ini, akan ditampilkan bagaimana alur kerja sistem tersebut direpresentasikan.

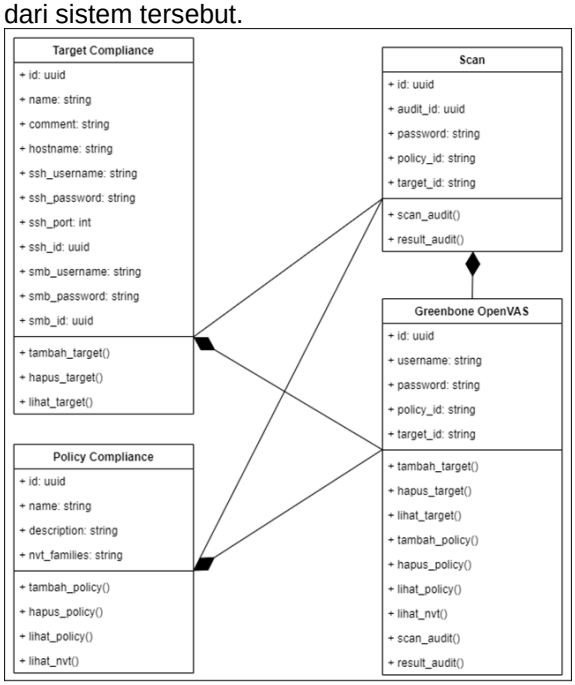


Sumber: Penelitian (2023)

Gambar 5. *Activity* diagram

6. *Class* Diagram

Class diagram digunakan untuk menjelaskan relasi antar tabel dalam *database* yang digunakan oleh aplikasi Helium *Security*. Diagram ini mengilustrasikan struktur, atribut, kelas, metode, dan hubungan dari setiap objek. Dengan *class* diagram, informasi tentang hubungan yang terjadi antara kelas-kelas dapat diperoleh. Pembuatan diagram kelas membantu dalam menciptakan representasi visual yang jelas dan terperinci. Pada gambar 5 di bawah ini, akan ditampilkan *class* diagram dari sistem tersebut.

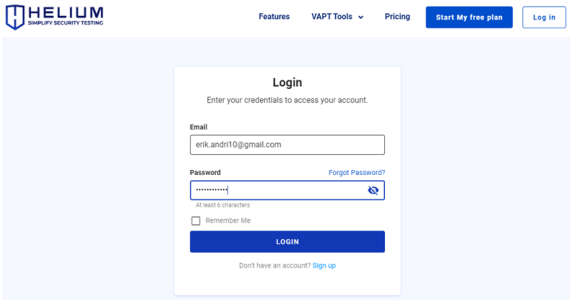


Sumber: Penelitian (2023)

Gambar 6. *Class* Diagram

7. *User Interface*

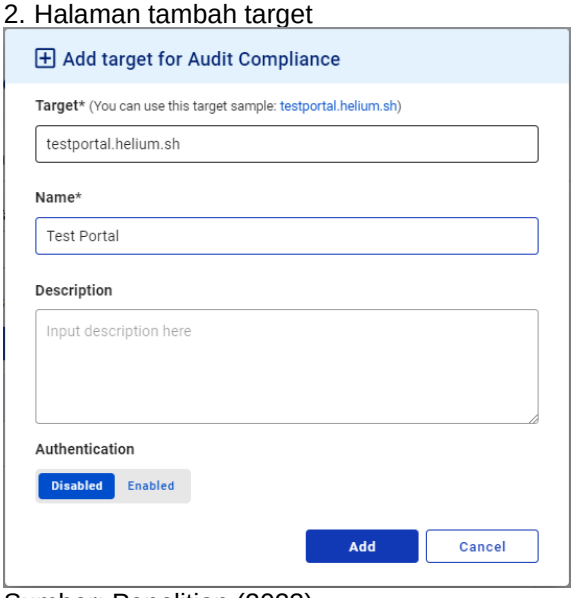
1. Halaman *login*



Sumber: Penelitian (2023)

Gambar 7. Halaman *login*

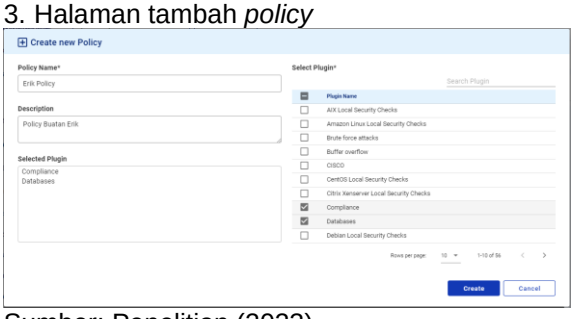
2. Halaman tambah target



Sumber: Penelitian (2023)

Gambar 8. Halaman tambah target

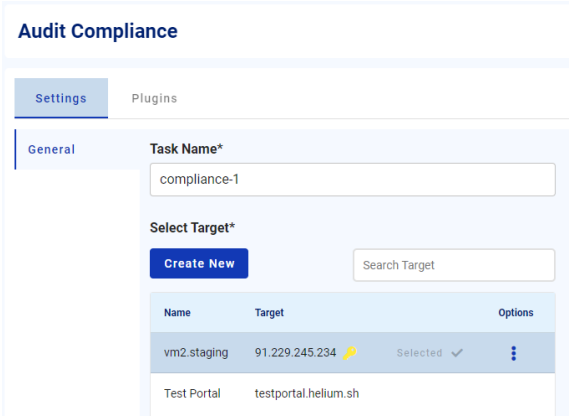
3. Halaman tambah *policy*



Sumber: Penelitian (2023)

Gambar 8 Halaman tambah *policy*

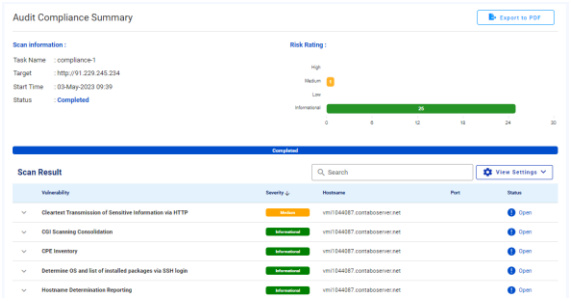
4. Halaman *scan* audit



Sumber: Penelitian (2023)

Gambar 10. Halaman *scan* audit

5. Halaman lihat hasil *scan*



Gambar 11. Halaman lihat hasil *scan*

8. Pengujian

1. *Black box testing*

Dalam tahapan pengujian sistem ini, Penulis menggunakan metode *black box testing* seperti yang dituangkan dalam tabel 3 di bawah ini di mana pengujian dilakukan dengan tujuan untuk memastikan bahwa sistem sudah teruji secara menyeluruh dan dapat berjalan dengan baik dalam segi fungsionalitas-Nya seperti yang diharapkan oleh pengguna.

Tabel 3 *Black box testing*

No	*Test case*	Skenario pengujian	Hasil yang didapatkan	Valid / Invalid
1	*Login*	Memasukkan *email* dan *password* akun Helium yang terdaftar	Berhasil masuk ke dalam *dashboard*	Valid
2	Tambah Target	Memasukkan target *hostname* dan nama target	Tampil informasi target berhasil ditambahkan	Valid
3	Hapus Target	Memilih target pada daftar target, lalu menekan tombol *Delete*	Tampil informasi target berhasil terhapus	Valid
4	Lihat Target	Masuk ke dalam halaman *Targets*, lalu menekan tab *Compliance*	Tampil informasi target milik pengguna dalam tabel	Valid
5	Tambah *Policy*	Memasukkan nama *policy* dan memilih *plugins*	Tampil informasi berhasil membuat *policy*	Valid
6	Hapus *Policy*	Memilih *policy* milik pengguna di daftar *policy* pada bagian *Customed* dan menekan tombol *Delete*	Tampil informasi *policy* berhasil dihapus	Valid
7	Lihat *Policy*	Masuk ke halaman Audit *Compliance*, lalu masuk ke tab *Plugins*	Tampil informasi sekumpulan *policy* dalam tabel	Valid
8	*Scan* Audit	Memasukkan nama *task*, target dan *policy* untuk melakukan audit	Audit berjalan dan tampil informasi detail *scan* yang diminta	Valid
9	Lihat Hasil Audit	Masuk ke halaman *Scans*, lalu menekan nama *task* pada kolom *Task Name*	Tampil informasi *scan* audit	Valid

Sumber: Penelitian (2023)

2. *Usability testing*

Usability Testing, seperti yang diuraikan oleh Jacob Nielsen dalam artikelnya "*Usability 101:*

Introduction to Usability", mengukur kualitas desain berdasarkan lima komponen utama: Kemudahan Pembelajaran (seberapa mudah pengguna dapat melakukan tugas awal), Efisiensi (seberapa cepat mereka dapat menyelesaikan tugas setelah mempelajarinya), Kemampuan Mengingat (seberapa mudah mereka dapat kembali menggunakan desain setelah absen), Tingkat Kesalahan (jumlah, tingkat keparahan, dan pemulihan dari kesalahan pengguna), dan Kepuasan (seberapa menyenangkan desain tersebut digunakan). Ini membantu dalam mengevaluasi sejauh mana desain memfasilitasi interaksi pengguna dan pengalaman pengguna secara keseluruhan (Neilsen, 2012). Pengguna diminta melakukan penilian dengan *range* angka antara 1 sampai dengan 5. Berikut adalah daftar pernyataan untuk *usability testing*:

Tabel 4. *Usability testing*

No	Pertanyaan kuesioner
1	Apakah Anda merasa mudah untuk mempelajari cara menggunakan fitur Audit *Compliance* pada aplikasi Helium *Security*?
2	Apakah merasa nyaman dalam belajar menggunakan fitur Audit *Compliance* dalam aplikasi Helium *Security*?
3	Apakah Anda merasa mampu mengingat dan menggunakan kembali fitur Audit *Compliance* pada aplikasi Helium *Security* setelah pertama kali menggunakannya?
4	Apakah Anda merasa mudah untuk mengingat langkah-langkah penggunaan fitur Audit *Compliance* pada aplikasi Helium *Security*?
5	Apakah anda merasa bahwa fitur Audit *Compliance* pada aplikasi Helium *Security* membantu Anda dalam melakukan validasi temuan kerentanan pada konfigurasi sistem?
6	Apakah sistem ini membantu Anda menghemat waktu dalam melakukan audit konfigurasi standar pada sistem yang berjalan?
7	Apakah Anda merasa bahwa fitur Audit *Compliance* pada aplikasi Helium *Security* tidak mengalami kesalahan atau masalah yang mengganggu dalam penggunaan?
8	Sejauh mana Anda percaya bahwa kesalahan yang terjadi diakibatkan oleh pengguna, bukan pada fitur itu sendiri?
9	Apakah Anda merasa puas dengan pengalaman menggunakan fitur Audit *Compliance* pada aplikasi Helium *Security* secara keseluruhan?
10	Bagaimana tingkat kepuasan Anda dengan visualisasi dari fitur Audit *Compliance* yang disediakan oleh aplikasi Helium *Security* ini?

Sumber: Penelitian (2023)

3. Rincian perhitungan keseluruhan

Tabel 5 Rincian perhitungan

P	SS	S	TS	STS	Total	Rata-rata
1	12	10	1	0	23	3,4782609
2	15	7	1	0	23	3,6086957
3	16	5	2	0	23	3,6086957
4	15	6	2	0	23	3,5652174
5	15	7	1	0	23	3,6086957
6	15	7	1	0	23	3,6086957
7	15	3	5	0	23	3,4347826
8	17	2	4	0	23	3,5652174
9	17	5	1	0	23	3,6956522
10	21	2	0	0	23	3,9130435

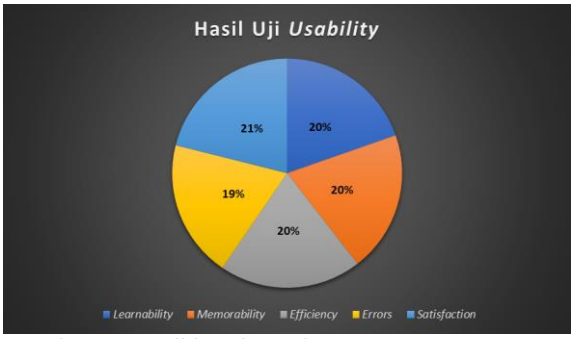
Sumber: Penelitian (2023)

Berdasarkan data yang terdokumentasi pada tabel 7, terlihat bahwa sebanyak 23 responden telah memberikan penilaian terhadap pertanyaan yang diajukan. Frekuensi mencerminkan berapa banyak responden yang memberikan penilaian yang serupa terhadap pertanyaan tersebut, sementara persentase adalah rata-rata hasil frekuensi dibagi dengan jumlah total responden. Total persentase di dalam tabel di atas mencerminkan akumulasi persentase dari masing-masing pertanyaan.

Tabel 6 Hasil uji *usability* keseluruhan

Indikator	Jumlah Responden	Rata-rata indikator (%)
Learnability	23	3,543
Memorability	23	3,587
Efficiency	23	3,609
Errors	23	3,5
Satisfaction	23	3,804

Sumber: Penelitian (2023)



Sumber: Penelitian (2023)

Gambar 12. *Pie* chart hasil uji *usability*

IV. KESIMPULAN

Berdasarkan pembahasan di atas, dapat disimpulkan sebagai berikut:

Perancangan fitur audit *security configuration compliance* sudah dapat digunakan sebagai sistem yang membantu mengidentifikasi temuan kerentanan terhadap ketidakpatuhan standar keamanan teknologi dengan baik berdasarkan hasil survei yang telah dilakukan mendapatkan hasil hingga 91,3% sangat setuju dan 8,7% setuju dengan total rata-rata kepuasan sebesar 3,8% dari skala 4%.

Fitur audit *security configuration compliance* di aplikasi Helium *Security* memudahkan pengguna untuk mengotomatisasi proses audit *security configuration compliance* dan mendapatkan hasil temuan atas ketidakpatuhan standar keamanan teknologi dengan baik berdasarkan hasil survei yang telah dilakukan mendapatkan hasil kemudahan penggunaan hingga 65,2% sangat setuju, 30,4% setuju dan 4,3% tidak setuju dengan total rata-rata kepuasan sebesar 3,5% dari skala 4%.

V. REFERENSI

About Us | Helium Security. (2022, Maret 25). https://www.helium.sh/about-us

Carter, W. A., & Zheng, D. E. (2015). *The Evolution of Cybersecurity Requirements for the U.S. Financial Industry*. 12. http://csis-website-prod.s3.amazonaws.com/s3fs-public/legacy_files/files/publication/150717_Carter_CybersecurityRequirements_Web.pdf

Cole, B. (2014, Juni 23). *What is IT audit (information technology audit)? | Definition from TechTarget*. https://www.techtarget.com/searchcio/definition/IT-audit-information-technology-audit

Cybersecurity Standards and Frameworks | IT Governance USA. (2017, Januari 5). IT Governance USA Inc. https://www.itgovernanceusa.com/cybersecurity-standards

Frameworks, Standards and Models | ISACA. (2008, Desember 19). https://www.isaca.org/resources/frameworks-standards-and-models

Introduction to the Python Flask framework - Alibaba Cloud Community. (2022, Januari 10). https://www.alibabacloud.com/blog/introduction-to-the-python-flask-framework_598443

ISO 27001, the Information Security Standard | IT Governance USA. (2022, Oktober 27). IT Governance USA Inc. https://www.itgovernanceusa.com/iso27001

ISO 27002: Security Controls. (2022, Oktober 27). IT Governance USA Inc. https://www.itgovernanceusa.com/iso27002

Moran - Director Of Services, D. (2017). Compliance Scanning THE MISSING PIECE OF VULNERABILITY MANAGEMENT. *Fortified Health Security*, 3. https://fortifiedhealthsecurity.com/wp-content/uploads/2017/06/Fortified-Compliance-Scanning-White-Paper.pdf

Neilsen, J. (2012, Januari 3). *Introduction to Usability*. Nielsen Norman Group. http://www.nngroup.com/articles/usability-101-introduction-to-usability

OpenVAS - Open Vulnerability Assessment Scanner. (2021, Juli 2). Greenbone. https://www.openvas.org/

Panatagama, A. (2023, Maret 3). *Apa Itu Agile Workflow? Pengertian dan Metodenya*. https://terralogiq.com/agile-workflow/

Sari, A. M. (2023, Juni 22). *CLI (Command Line Interface), Cara Kerja dan Manfaat - FIKTI*. https://fikti.umsu.ac.id/cli-command-line-interface-cara-kerja-dan-manfaat/

What is an API? A Beginner's Guide to APIs | Postman. (2023, April 27). https://www.postman.com/what-is-an-api/

What is Python? Executive Summary | Python.org. (2022, Maret 25). https://www.python.org/doc/essays/blurb/

What is the NIST Cybersecurity Framework (CSF)? | IT Governance USA. (2022, Oktober 27). IT Governance USA Inc. https://www.itgovernanceusa.com/nist-cybersecurity-framework