KATA PENGANTAR

Segala puji dan syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Buku Tugas Akhir Capstone Design yang berjudul ”Pengujian Keamanan Server Berbasis *Website*” dengan baik dan tepat waktu. Buku tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (S.T) pada Program Studi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom Univeristy.

Penulis menyadari bahwa dalam penyusunan pengembangan Tugas Akhir ini masih terdapat kekurangan baik dari sisi materi, teknik penulisan, dan keterbatasan ilmu yang penulis miliki. Oleh karena itu, penulis sangat terbuka dan mengharapkan kritik dan saran yang membangun demi memperbaiki pengembangan Tugas Akhir ini di masa yang akan datang. Penulis berharap bahwa pengembangan Tugas Akhir ini dapat memberikan manfaat bagi pembaca dan menjadi sumber refenrensi bagi perkembangan ilmu pengetahuan.

UCAPAN TERIMAKASIH

Dalam pengembangan Tugas Akhir ini, penulis menyadari bahwa pengembangan Tugas Akhir Capstone Design "Pengujian Keamanan Server Berbasis *Website*" ini tidak dapat terwujud tanpa dukungan, bimbingan, bantuan dan do'a yang diberikan pleh berbagai pihak selama proses penyelesaian. Oleh karena itu, penulis ingin mengungkapkan rasa terimakasih yang mendalam kepada :

1. Tuhan Yang Maha Esa atas rahmat dan karunia-nya yang telah memberikan Penulis kemampuan dan kesempatan unyuk menyelesaikan Tugas Akhir Capstone Design "Pengujian Keamanan Server Berbasis *Website*" ini.
2. Bapak Dr.Eng. Favian Dewanta,S.T.,M.Eng. selaku Pembimbing I dan Bapak Bagus Aditya, S.T., M.T. selaku Pembimbing II yang dengan sabar memberikan arahan, nasihat, pengetahuan, dan masukan yang berharga selama proses pengerjaan Tugas Akhir ini.
3. Seluruh dosen di Program Studi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom Bandung, yang telah memberikan ilmu dan wawasan yang sangat berarti dalam perjalanan akademik penulis.
4. Rekan tim Tugas Akhir yang telah bekerjasama dengan baik untuk menyelesaikan Tugas Akhir ini.
5. Kedua orang tua penulis atas dukungan, doa, motivasi dan nasihat kepada Penulis yang menjadi sumber kekuatan dan motivasi dalam menyelesaikan studi ini.
6. Teman – teman seperjuangan yang selalu memberikan dukungan moral, bantuan dan kebersamaan selama masa studi hingga penyusunan Tugas Akhir ini.
7. Seluruh pihak lainnya yang tidak dapat disebutkan satu per satu, tetapi turut berperan dalam memberikan dukungan dan motivasi selama proses penyelesaian Tugas Akhir ini.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN

SQL injection	: *Structured Query Language Injection*
SSL	: *Secure Socket Layer*
TLS	: *Transport Layer Security*
DDoS	: *Distribution Denial Of Service*
Nmap	: *Network Mapper*
AI	: *Artificial Intelligence*
OWASP	: *Open Web Application Security Project*
CI/CD	: *Continuous Integration / Continuous Deployment*
SSRF	: *Server-Side Request Forgery*
OS	: *Operating System*
GPL	: *General Public License*
HTTP	: *Hypertext Transfer Protocol*
CVSS	: *Common Vulnerability Scoring System*
MVC	: *Model-View-Controller*
MVT	: *Models-View-Template*
URL	: *Uniform Resource Locator*
ORM	: *Object-Relational Mapping*
PDF	: *Portable Document Format*
IP	: *Internet Protocol*
FGD	: *Focus Group Discussion*
HTML	: *HyperText Markup Language*
CRUD	: *Create, Read, Update, Delete*
API	: *Application Programming Interface*
CSS	: *Cascading Style Sheets*

BAB 1
USULAN GAGASAN

1.1 Deskripsi Umum Masalah

Server merupakan sistem komputer yang menyediakan berbagai layanan dalam jaringan komputer[19]. Server terdiri dari berbagai jenis, diantaranya *web* server yang berfungsi untuk memberikan konten *web* kepada pengguna, *application* server yang digunakan untuk menjalankan dan mengelola aplikasi, *database* server yang digunakan untuk menyimpan dan mengelola data , email server yang digunakan untuk mengelola pengiriman dan penerimaan pesan, dan masih banyak lagi[18] . Saat ini, server berperan penting dalam keberlangsungan teknologi pada kehidupan sehari – hari.

Dengan semakin banyaknya penggunaan server dalam kehidupan sehari-hari di kalangan perusahaan, organisasi dan individu, risiko serangan *cyber* pada server semakin meningkat. Kompleksitas masalah dalam keamanan server terletak pada serangan *cyber* seperti pencurian data, peretasan, serangan kerentanan terhadap *malware* dan serangan lainnya yang berkelanjutan. Maka, keamanan server menjadi salah satu hal yang perlu diperhatikan dan ditingkatkan untuk melindungi data pengguna serta memastikan kelangsungan layanan pada server tersebut berjalan dengan baik.

1.2 Analisis Masalah

Pada bagian ini merupakan analisa masalah yang dituangkan berupa aspek – aspek diantarnya aspek keamanan, aspek teknis dan aspek lingkungan. Adapun analisis aspek aspek tersebut yaitu :

1.2.1 Aspek Keamanan

Server merupakan target utama dari berbagai serangan keamanan yang dapat membahayakan integritas, ketersediaan data, dan kerahasian. Ancaman – ancaman ini diantaranya *SQL injection* yang merupakan salah satu serangan umum yang terjadi pada server, dimana penyerang menyisipkan kode berbahaya untuk mengakses dan merusak basis data, selain itu ancaman *Cross-site Scripting*, serangan *brute force*, tidak ada implementasi enkripsi seperti SSL/TLS yang dapat membuat data yang dikirim antara server dan klien rentan terhadap penyadapan dan serangan *Distributed Denial of Service* (DDoS) yang dapat memenuhi server dengan lalu lintas palsu [2].

1.2.2 Aspek Teknis

Pada sebuah server, penting untuk memastikan server tersebut bebas dari celah keamanan yang dapat dimanfaatkan oleh penyerang. Dengan melakukan evaluasi keamanan secara berkala dan menyeluruh maka keamanan data pengguna, 5 mekanisme enkripsi data, kelemahan teknis dan fitur-fitur keamanan pada *Website* dapat didentifikasi dan diperbaiki keamanannya sehingga membantu aplikasi tetap berjalan dengan lancar[11].

1.2.3 Aspek Lingkungan

Keamanan pada server tidak hanya berdampak terhadap kualitas layanan dari server tersebut tetapi juga berdampak pada para pengguna. Server yang aman dan terpercaya memberikan kenyamanan pada pengguna, mengurangi adanya risiko kebocoran data yang akan meningkatkan kepercayaan pengguna[5]. Selain itu, adanya evaluasi keamanan yang dilakukan secara berkala juga turut berkontribusi untuk menangani insiden pada keamanan sehingga mendukung teknologi yang semakin ramah lingkungan.

1.3 Analisis Solusi yang Ada

Dalam konteks keamanan pada server dengan adanya tantangan serangan *cyber* berupa peretasan, pencurian data dan serangan *malware*. Maka, penyedia layanan diharapkan untuk mendeteksi, mencengah dan mengurangi risiko keamanan seperti menerapkan teknik autentikasi dan enkripsi yang baik serta melakukan pengecekan secara berkala guna memastikan bahwa fitur-fitur keamanan yang ada di penyedia layanan telah berfungsi dengan baik sehingga tidak ada celah yang dapat dimanfaatkan oleh penyerang.

1.4 Tujuan Tugas Akhir

Tujuan dari *capstone design* ini adalah untuk merancang dan mengembangkan solusi yang mampu mengatasi masalah keamanan *cyber* yang semakin mengingkat. Maka, pada *capstone design* ini menawarkan sebuah *platform* yang berisi *tools* yang dapat digunakan untuk pengujian keamanan server. *Platform* ini bertujuan untuk mengetahui celah keamanan pada suatu server sehingga *administrator* server dapat melakukan perbaikan terhadap temuan temuan celah keamanan sebelum dimanfaatkan oleh pihak yang tidak berwenang. *Platform* ini dikembangkan sebagai solusi keamanan server berbasis *Website* yang dapat diakses dengan mudah, sehingga mampu melayani berbagai kalangan, mulai dari pemilik bisnis kecil hingga perusahaan besar.

1.5 Batasan Tugas Akhir

Berikut merupakan batasan – batasan yang perlu diperhatikan dalam pengembangan *platform* ini :

- Solusi yang dikembangkan tidak mencakup pembuatan *tools* pengujian keamanan dari awal, melainkan mengembangkan sebuah *platform* berbasis *Website* yang mengintegrasikan *tools open source* yang sudah ada, dengan fokus utamanya adalah *tools* Nmap dan Nikto.
- Fitur pemindaian kerentanan yang disediakan *platform* ini terbatas pada identifikasi sistem operasi, pemindaian *port*, dan deteksi layanan atau aplikasi spesifik pada server target.
- *Platform* hanya berfungsi untuk mendeteksi, melaporkan temuan, dan memberikan rekomendasi perbaikan berbasis *generative* AI terhadap celah keamanan. Tindakan perbaikan atau mitigasi kerentanan tidak termasuk dalam ruang lingkup tugas akhir ini.
- Pengujian keamanan pada *platform* ini hanya terbatas pada keamanan server sebagai target utama. *Platform* ini tidak memeriksa keamanan seluruh perangkat dalam jaringan seperti router*, firewall*, atau komputer lainnya.
- Interaksi dengan sistem dibatasi hanya pada dua peran pengguna, yaitu "*user*" yang dapat menggunakan fitur pemindaian dan "*admin*" yang dapat mengelola data hasil pemindaian dan profil pengguna.

BAB 2
TINJAUAN PUSTAKA

2.1 Vulnerability *Scanning*

Vulnerability scanning merupakan pemindaian kerentanan yang dapat mengindentifikasi kerentanan atau celah keamanan yang dieksploitasi penyerang pada server. Tujuan dari *vulnerability scanning* ini adalah untuk mendeteksi potensi masalah keamanan sebelum dapat diekspoitasi oleh pihak yang tidak bertanggung jawab, *vulnerability scanning* ini dapat memberikan hasil temuan kerentanan serta memberikan rekomendasi perbaikan untuk menutup celah keamanan yang ditemukan.[14] Pemindaian keamanan ini dilakukan dengan menggunakan *tools* khusus yang dapat digunakan untuk mendeteksi celah keamanan pada server.

2.2 OWASP

OWASP (*Open Web Application Security Project*) adalah organisasi nirlaba internasional yang fokus pada keamanan aplikasi *web*. Prinsip utama OWASP adalah menyediakan semua materi secara gratis dan mudah diakses melalui *Website* organisasi tersebut, sehingga pengembang dan administrator dapat meningkatkan keamanan aplikasi *web* yang dimiliki. Dalam konteks pengujian keamanan server, OWASP Top 10 memberikan panduan yang sangat berharga untuk mengidentifikasi celah keamanan yang umum ditemukan pada aplikasi *web*. Hal ini menjadikan OWASP sebagai referensi penting bagi *administrator* sistem dan profesional keamanan dalam melakukan assessment keamanan server [7].

OWASP telah menjadi standar industri dalam mengidentifikasi resiko keamanan aplikasi *web* yang paling kritis. Salah satu kontribusi terpenting OWASP adalah daftar "OWASP Top 10" yang berisi sepuluh kerentanan keamanan aplikasi *web* paling berbahaya. Daftar ini diperbarui secara berkala setiap 3-4 tahun untuk mengikuti perkembangan ancaman keamanan terbaru [22]. Berikut merupakan TOP 10 daftar risiko keamanan paling kritis menurut OWASP tahun 2021 :

Tabel 1. 1 OWASP Top 10 *Vulnerability*

No	OWASP Top 10 *Vulnerability Types* 2021	Deskripsi
1	*Broken Access Control*	Kerentanan yang terjadi karena penerapan kontrol akses situs *web* yang tidak tepat. Hal ini memungkinkan penyerang masuk ke dalam sistem untuk mencuri informasi [22].
2	*Cryptographic Failures*	Kerentanan yang terjadi ketika sistem kriptografi rusak dan tidak dapat melindungi informasi sensitif [22].
3	*Injection*	Kerentanan pada server yang memungkinkan data masukan *user* dianggap sebagai perintah sehingga sistem dapat menjalankan perintah yang tidak diizinkan yang menyebabkan kebocoran data [22].
4	*Insecure Design*	Kerentanan yang disebabkan oleh desain yang tidak aman atau kelemahan pada desain [22].
5	*Security Misconfiguration*	Sistem yang tidak dikonfigurasi dengan benar sehingga penyerang dapat mengeksploitasi sistem [22].
6	*Vulnerable and Outdated Components*	Kerentanan yang disebabkan oleh perangkat lunak yang usang, rentan dan sudah tidak didukung [22].
7	*Identification and Authentication Failures*	Risiko keamanan terjadi ketika identitas, otentikasi, atau manajemen sesi pengguna

		tidak ditangani dengan hati-hati, yang dapat memungkinkan penyerang untuk memanfaatkan kata sandi, kunci, token sesi, atau kelemahan dalam perangkat lunak untuk mengambil alih identitas pengguna [22].
8	*Software and Data Integrity Failures*	Disebabkan oleh infrastruktur dan kode yang gagal melindungi dari serangan. Pembaruan perangkat lunak, data penting, dan *pipeline* CI/CD yang diterapkan tanpa verifikasi termasuk di dalamnya [22].
9	*Security Logging and Monitoring Failures*	Kerentanan yang terjadi ketika aktivitas sistem tidak diawasi secara aktif yang menyebabkan hilangnya kemampuan untuk serangan, menerima peringatan dini, dan melakukan investigasi [22].
10	*Server-Side Request Forgery*	Serangan SSRF dibuat untuk memanfaatkan cara server menangani informasi eksternal. Tujuan utama serangan ini adalah untuk mencuri informasi sensitif di server atau memanfaatkan kerentanan lain dengan menggunakan tindakan penanggulangan validasi *input* SSRF [22].

2.3 Nmap *scanner*

Nmap ("*Network Mapper*") adalah utilitas sumber terbuka dan gratis untuk penemuan jaringan dan audit keamanan. Banyak sistem dan administrator jaringan juga menganggapnya berguna untuk tugas-tugas seperti inventaris jaringan, mengelola jadwal pemutakhiran layanan, dan memantau waktu aktif host atau layanan [17]. Nmap mampu mengumpulkan beragam

informasi esensial mengenai topologi dan konfigurasi keamanan sebuah jaringan. Kemampuan utamanya meliputi proses enumerasi untuk menentukan *host* mana saja yang aktif, identifikasi layanan yang berjalan pada setiap *host* beserta nama dan versi aplikasinya, serta deteksi sistem operasi (*OS fingerprinting*) untuk mengetahui jenis dan versi OS yang digunakan. Lebih lanjut, Nmap dapat memberikan indikasi mengenai keberadaan dan jenis *firewall* atau filter paket yang melindungi *host*.

Nmap dirancang untuk memindai jaringan besar dengan cepat, tetapi juga berfungsi dengan baik terhadap *host* tunggal. Fleksibilitas ini menjadikan Nmap sebagai alat yang sangat berguna dalam berbagai skenario audit keamanan jaringan. Nmap berjalan pada semua sistem operasi komputer utama, dengan paket *biner* resmi yang tersedia untuk *windows, linux* dan *macOSx*.

Dalam konteks pemindaian keamanan jaringan, Nmap berperan sebagai alat utama untuk identifikasi aset jaringan, pemetaan topologi jaringan, deteksi layanan yang berjalan, identifikasi potensi kerentanan dan monitoring perubahan konfigurasi jaringan.

2.4 Nikto *Scanner*

Nikto adalah pemindai server *web Open Source* GPL (*General Public License*) yang melakukan pengujian menyeluruh terhadap server *web* untuk berbagai item, termasuk lebih dari 7.000 file/program yang berpotensi berbahaya, memeriksa versi lama dari lebih dari 1250 server, dan masalah khusus versi pada lebih dari 270 server [23]. Nikto juga memeriksa item konfigurasi server seperti keberadaan beberapa file indeks, opsi server HTTP, dan akan mencoba mengidentifikasi server *web* dan perangkat lunak yang terpasang.

Dalam menajalankan pemindaian, nikto melakukan pendekatan yang mengutamakan kecepatan dan kelengkapan analisis daripada beroperasi secara tersembunyi. Namun, setiap pengujian yang dilakukan menghasilkan aktivitas jaringan yang intensif sehingga jejaknya sangat mudah terindentifikasi.

Laporan yang dihasilkan oleh nikto tidak hanya sebatas pada celah keamanan yang bersifat kritis saja, nikto juga memberikan serangkaian temuan informatif yang berfungsi untuk memberi tahu administrator mengenai keberadaan direktori, berkas, atau konfigurasi server tertentu yang mungkin belum terdata, sekalipun tidak menimbulkan risiko keamanan secara langsung.

Nikto memiliki beberapa fungsi utama diantaranya kesalahan konfigurasi server dan perangkat lunak, deteksi file standar yang mungkin tidak dihapus setelah instalasi, pencarian file dan program yang tidak aman, identifikasi versi *software* yang memerlukan pembaruan

dan memberikan rekomendasi untuk melakukan pengujian manual yang lebih mendalam. Dalam pengujian keamanan *web*, nikto berperan sebagai *scanner* kerentanan otomatis, Mengidentifikasi teknologi dan versi *software* yang digunakan, Memberikan penilaian awal terhadap kondisi keamanan *web* server, dan Menjadi bagian dari *toolkit penetration testing* untuk *web applications*.

Item pemindaian dan *plugin* nikto sering diperbarui dan dapat diperbarui secara otomatis dari https://www.cirt.net , memastikan bahwa *tool* ini selalu memiliki informasi kerentanan terbaru untuk melakukan *scanning* yang efektif.

2.5 CVSS

Common Vulnerability Scoring System (CVSS) merupakan sebuah kerangka kerja standar global yang dirancang untuk mengkomunikasikan karakteristik dan mengukur tingkat keparahan kerentanan keamanan perangkat lunak. Versi terbarunya, CVSS 4.0, menyempurnakan metodologi penilaian dengan memperkenalkan nomenklatur metrik *Base, Threat,* dan *Environmental* (CVSS-BTE) untuk memberikan gambaran risiko yang lebih komprehensif [20]. Metrik Dasar (*Base Metrics*) mengevaluasi karakteristik fundamental kerentanan, seperti Vektor Serangan, Persyaratan Serangan (*Attack Requirements*), serta dampak terhadap kerahasiaan, integritas, dan ketersediaan, baik pada sistem yang rentan (*Vulnerable System*) maupun sistem selanjutnya (*Subsequent Systems*) [20].

Metrik Ancaman (*Threat Metrics*) menyesuaikan skor dasar dengan mempertimbangkan faktor-faktor temporal seperti ketersediaan bukti konsep eksploitasi. Sementara itu, Metrik Lingkungan (*Environmental Metrics*) memungkinkan organisasi untuk mengontekstualisasikan tingkat keparahan dengan memodifikasi metrik dasar sesuai dengan lingkungan spesifik dan kontrol keamanan yang ada. Penerapan kerangka kerja CVSS yang sistematis ini menjadi krusial dalam program manajemen kerentanan modern [3].

2.6 Django

Django adalah sebuah kerangka kerja pengembangan aplikasi *web* tingkat tinggi (*high-level*) berbasis *Python* yang dirancang untuk memfasilitasi pengembangan yang cepat dan pragmatis. Django menerapkan variasi dari pola arsitektur *Model-View-Controller* (MVC) yang dikenal sebagai *Model-View-Template* (MVT), yang secara tegas memisahkan antara logika bisnis (Model), logika presentasi (*View*), dan antarmuka pengguna (*Template*) [24]. Filosofi desain utamanya adalah *"batteries-included"*, yang mengindikasikan bahwa Django menyediakan serangkaian fungsionalitas inti yang luas dan terintegrasi, siap pakai untuk

menangani kebutuhan pengembangan *web* yang umum [13].

Komponen-komponen fundamental ini mencakup *Object-Relational Mapper* (ORM) untuk interaksi basis data yang aman dan intuitif, sistem perutean URL, mesin *template*, serta berbagai mekanisme perlindungan keamanan bawaan terhadap vektor serangan umum. Dengan menyediakan fondasi yang kokoh dan komponen yang dapat digunakan kembali, Django memungkinkan pengembang untuk lebih fokus pada aspek unik dari aplikasi mereka daripada membangun fungsionalitas dasar dari awal, sehingga secara signifikan mempercepat siklus hidup pengembangan (*development lifecycle*) untuk aplikasi yang kompleks dan berskala produksi [24].

2.7 Docker

Docker adalah sebuah *platform* terkemuka yang mengimplementasikan teknologi kontainerisasi untuk mengemas aplikasi dan seluruh dependensinya ke dalam unit-unit *port*abel yang terisolasi. Teknologi ini memanfaatkan virtualisasi pada tingkat sistem operasi (*OS-level virtualization*) untuk memungkinkan beberapa kontainer berjalan secara simultan pada satu *host* tanpa memerlukan *hypervisor* atau sistem operasi tamu (*guest OS*) penuh untuk setiap aplikasi [16]. Setiap kontainer mencakup semua yang dibutuhkan aplikasi untuk berjalan—kode, *runtime*, *tools*, pustaka, dan pengaturan sehingga menjamin konsistensi perilaku aplikasi di berbagai lingkungan komputasi.

Konsistensi ini didefinisikan secara deklaratif melalui sebuah *Dockerfile*, yang berisi instruksi untuk membangun sebuah *image* kontainer yang spesifik dan dapat direproduksi [16]. Dengan menghilangkan variabilitas lingkungan antara mesin pengembang dan server produksi, docker secara efektif memecahkan tantangan klasik dalam pengembangan perangkat lunak terkait aplikasi berjalan yang berjalan lancar di komputer pengembang, tetapi gagal berfungsi saat dipindahkan ke server produksi. Efisiensi sumber daya, *port*abilitas, dan skalabilitas yang ditawarkan oleh docker menjadikannya sebagai teknologi fondasional dalam paradigma rekayasa perangkat lunak modern, termasuk arsitektur *microservices*, serta alur kerja *Continuous Integration/Continuous Deployment* (CI/CD) dalam ekosistem *cloud-native* [16].

BAB 3
SPESIFIKASI DAN DESAIN SISTEM

3.1 Spesifikasi Sistem

Pengujian keamanan server bertujuan untuk meminimalisir celah keamanan yang dapat dimanfaatkan oleh *hacker* untuk merugikan pihak *administrator* server. Keamanan server harus mencakup perlindungan data sensitif, mencegah akses yang tidak sah, menjamin ketersediaan layanan, dan mencegah penyalahgunaan server[1]. Maka, terdapat beberapa spesifikasi dari *platform* pengujian keamanan server berbasis *Website* ini memenuhi standar yang telah ditentukan dan sesuai dengan kebutuhan pengguna. Berikut merupakan tabel yang dilampirkan mengenai spesifikasi yang menjadi fokus utama dalam pengembangan *platform* ini :

Tabel 3. 1 Spesifikasi Sistem

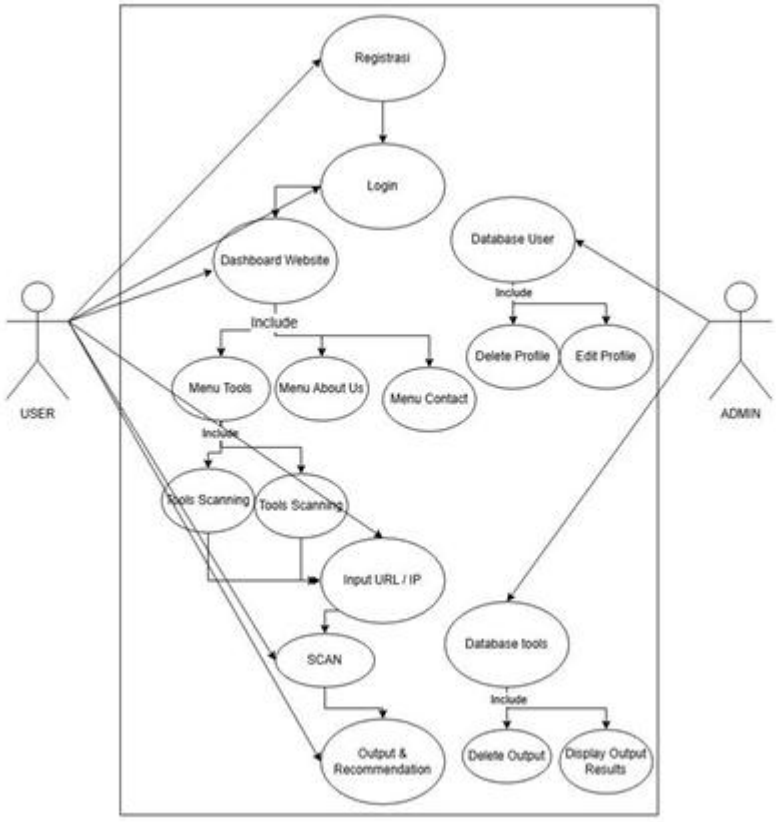
No	Aspek/Fitur	Keterangan
1.	*Scanning*	*Scanning* dalam pengujian keamanan server ini merupakan proses mendeteksi, mengidentifikasi, dan menganalisis kerentanan, konfigurasi salah, atau potensi ancaman keamanan pada server[4].
2.	*Reporting*	*Reporting* merupakan proses dokumentasi dan pelaporan hasil - hasil pengujian keamanan yang telah dilakukan terhadap sistem, jaringan,atau server. Tujuan dari *reporting* adalah untuk menyampaikan temuan dan menganalisis risiko. *Reporting* ini memberikan informasi yang jelas kepada pengguna untuk memahami kondisi keamanan dalam server tersebut [12].

3.	Saran/Rekomendasi	Saran atau rekomendasi dalam konteks pengujian keamanan server atau sistem adalah tindakan atau strategi yang diusulkan untuk mengatasi atau memitigasi kerentanan,ancaman,atau masalah yang ditemukan selama pengujian keamanan [6].

3.2 Desain Sistem

Desain sistem yang akan digunakan adalah pengembangan sebuah *platform* pengujian keamanan server berbasis *Website* yang mengintegrasikan beberapa *tools* pengujian keamanan, dengan fokus utamanya pada *tools* nmap dan nikto. Dengan adanya penggabungan beberapa *tools* pada *platfom* pengujian keamanan ini dapat mempermudah pengguna untuk mendapatkan kemampuan beberapa *tools* secara bersamaan. Dalam satu *platform*, pengguna dapat menjalankan pemindaian nmap untuk mengidentifikasi perangkat dalam jaringan, menentukan layanan yang berjalan, serta menemukan kerentanan pada *port* server tersebut[8] .Pengguna juga dapat menggunakan Nikto untuk menganalisis kerentanan pada server *web* seperti file sensitif, konfigurasi lemah, dan kelemahan plugin[25]. *Platform* ini juga mendukung kemampuan *reporting* berupa informasi umum *scan*, daftar kerentanan serta ringkasan kerentanan dalam bentuk *chart* dan file PDF yang dapat pengguna unduh, serta terdapat saran yang didukung oleh rekomendasi AI. Fitur ini memastikan laporan yang dihasilkan lebih akurat dan relevan, dengan meminimalkan risiko pelaporan palsu.

3.2.1 *Usecase* Diagram

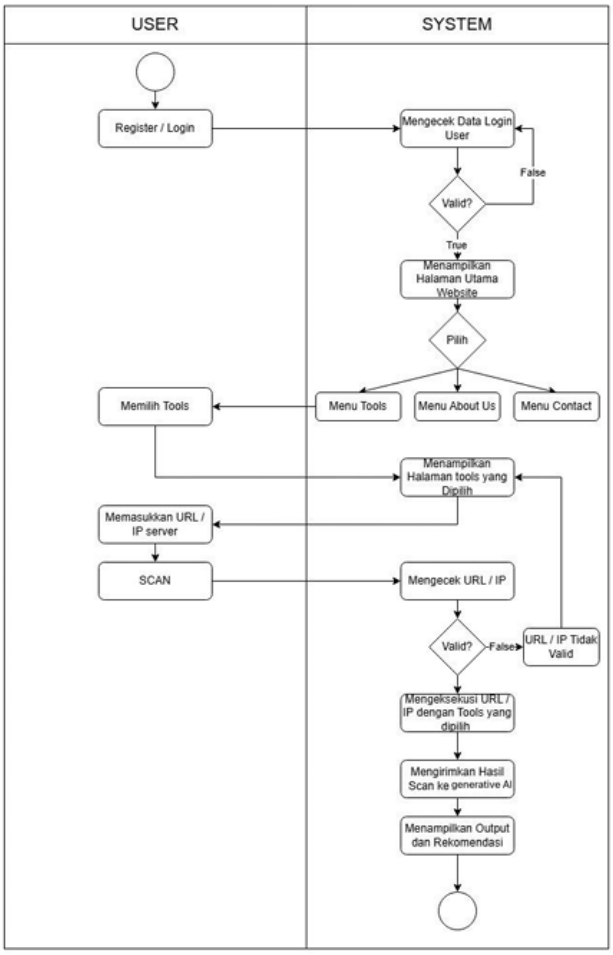


Gambar 3. 1 *Usecase* Diagram

Gambar 3.1 merupakan *Usecase* diagram yang menggambarkan sistem pengujian keamanan berbasis *Website* yang melibatkan dua pihak utama yaitu *user* dan *admin*. Pada sisi *user* dapat mulai menjalankan *Website* melalui tahap awal yaitu registrasi. Setelah berhasil melakukan registrasi, *user* dapat melakukan *login* untuk masuk ke *dashboard Website* yang terdiri dari *menu tools*, *menu about us*, *menu contact*. Pada *menu tools*, *user* dapat memulai pengujian keamanan dengan menggunakan *tools scanning* dan memasukan URL/IP yang akan menjadi target pengujian. Setelah itu, proses *scanning* dilakukan dan akan menghasilkan *output reporting* serta rekomendasi yang berisi hasil analisis dan rekomendasi perbaikan.

Pada sisi *admin* berfokus pada pengelolaan data. Dengan adanya *database user*, *admin* dapat mengelola informasi *user* seperti *delete* profil dan *edit* profil. *Admin* juga memiliki akses ke *database tools* untuk mengelola hasil *scanning* seperti menampilkan hasil *scanning* pada *display output result* dan menghapus hasil *scanning* melalui *delete output*.

3.2.2 *Activity* Diagram



Gambar 3. 2 Activity Diagram

Gambar 3.2 merupakan *activity* diagram yang menggambarkan mengenai alur kerja antara *user* dan sistem. Dimulai dari *user* melakukan proses registrasi/*login* lalu diverifikasi oleh sistem pada langkah memeriksa data *login user*. Jika *login* dinyatakan *valid* maka sistem akan mengarahkan *user* ke halaman utama *Website*, namun jika gagal maka sistem akan mengecek kembali data *login user*. Di dalam halaman utama *Website* terdapat 3 *menu* yang dapat dipilih oleh *user* yaitu *menu tools*, *about us* dan *contact*.

Jika *user* memilih *menu tools*, sistem akan menampilkan halaman sesuai dengan *tools* yang dipilih. Pada halaman tersebut *user* dapat memasukan URL/IP. Lalu, sistem akan mengecek URL/IP yang dimasukan *valid* atau tidak. Jika URL/IP tidak *valid* maka sistem akan kembali ke *menu tools*. Namun, jika URL/IP *valid* maka sistem akan mengeksekusi URL/IP menggunakan *tools* yang sebelumnya sudah dipilih oleh *user*. Lalu, sistem akan mengirimkan hasil *scanning* ke AI. setelah itu, sistem menampilkan *output* serta rekomendasi perbaikan di *Website*.

3.3 Metode Pengukuran yang Sesuai dengan Solusi Terpilih

Pada bagian ini merupakan metode pengukuran yang digunakan untuk mendukung implementasi solusi terpilih. Metode pengukuran ini mencakup proses *scanning*, *reporting* dan rekomendasi perbaikan. Berikut merupakan mekanisme pengukuran *scanning* yang tercantum pada Tabel 3.2 Pengukuran *Scanning*

Tabel 3. 2 Pengukuran *Scanning*

Keterangan	***Scanning***
Rincian	Memindai kerentanan target dengan menggunakan dua pilihan *tools* yaitu nmap dan nikto dengan fitur sebagai berikut : 1. Nmap *scanner* : OS *Detection*, *Port scan*, *Vurnerability*, dan *comperhensive*. 2. Nikto *Scanner* : *Basic scan*, *Full scan*, SSL/TSL *scan*, dan *custom scan* (menyesuaikan parameter lanjutan).
Mekanisme Pengukuran	Menguji fungsi *tools* untuk mengindentifikasi kerentanan pada sebuah target dengan menggunakan salah satu *tools* yang tersedia (nmap atau nikto)
Prosedur Pengukuran	- Pilih salah satu opsi *tools* (nmap atau nikto) - Masukan URL/IP target yang akan diuji kerentanannya - Memilih fitur *scan* yang tersedia pada masing – masing *tools* sesuai kebutuhan. - Pengecekan status terhadap URL/IP target - Status target dapat di akses maka *scanning* dilanjutkan

Berikut merupakan mekanisme pengukuran proses *reporting* yang tercantum pada **Tabel 3.3** *Reporting*.

Tabel 3. 3 *Reporting*

Keterangan	***Reporting***
Rincian	Menyediakan pelaporan berupa ringkasan pemindaian, skor keamanan, analisis keamanan detail dengan kategori tingkat kerentanan *Critical, High, Medium, Low, Info*. dan menyediakan pelaporan data hasil *security scan*.
Mekanisme Pengukuran	- Menampilkan ringkasan pemindaian secara umum diantaranya target yang di uji, *tools* yang digunakan, tipe fitur scan yang digunakan, total kerentanan yang ditemukan. - Pengecekan nilai keamanan berdasarkan kerentanan yang ditemukan dan ditampilkan dalam angka pada skor keamanan sesuai dengan standar perhitungan CVSS. - Menjelaskan temuan kerentanan secara detail dan mengelompokannya sesuai dengan kategori tingkat kerentanan berdasarkan standar OWASP. - Menampilkan ringkasan hasil *reporting* dalam bentuk point – point utama.
Prosedur Pengukuran	- Sistem menemukan kerentanan pada url/ip target. - Sistem mengelompokan kerentanan sesuai dengan level kategori kerentanan. - Menghitung skor kemanan berdasarkan standar perhitungan CVSS. - Menentukan skor keamanan url/ip secara keseluruhan dengan rentang 0-100. - Menampilkan seluruh data *reporting* ke halaman riwayat.

Berikut merupakan mekanisme pengukuran pada bagian rekomendasi perbaikan yang tercantum pada Tabel 3.4 Rekomendasi Perbaikan

Tabel 3. 4 Rekomendasi Perbaikan

Keterangan	Rekomendasi Perbaikan
Rincian	Menyediakan rekomendasi perbaikan berbasis *generative* AI.
Mekanisme Pengukuran	Menampilkan rekomendasi perbaikan dari kerentanan ditemukan dan dapat diunduh dalam format PDF.
Prosedur Pengukuran	- Menyimpan hasil *scanning* ke *database.* - Hasil *scanning* dikirim ke *generative* AI. - AI mengolah data hasil *scanning*. - *Generative* AI menghasilkan rekomendasi dari data tersebut. - Hasil rekomendasi dari *generative* AI disimpan di *database.* - Hasil rekomendasi perbaikan ditampilkan pada halaman riwayat.

BAB 4
IMPLEMENTASI

4.1 Deskripsi umum implementasi

Pada implementasi *platform* Pengujian Keamanan Server Berbasis *Website* dirancang sesuai dengan desain solusi terpilih sebelumnya yaitu dengan mengintegrasikan beberapa *tools* pengujian keamanan yang dibuat dalam bentuk *Website*. Sistem implementasi dirancang sesuai dengan desain solusi terpilih pada CD-3. Proses perancangan dimulai dengan melakukan *Focus Group Discussion* (FGD), FGD ini mendiskusikan *tools* yang akan digunakan untuk memindai ancaman atau kerentanan pada suatu server dan belajar untuk memahami serta mengetahui proses ancaman yang sering terjadi pada suatu server saat ini.

Pada platform ini, *tools* yang ditentukan berfokus pada dua *tools* utama yang akan digunakan yaitu Nmap dan Nikto. Nmap berfungsi untuk mengindentifikasi perangkat dalam jaringan, menentukan layanan yang berjalan, serta memetakan topologi jaringan. Sementara itu, Nikto dapat digunakan untuk menganalisis server *web* yang terdeteksi, mengevaluasi kerentanan umum seperti file sensitif yang dapat diakses, konfigurasi server yang lemah, atau kelemahan pada *plugin* yang digunakan.

Dalam pengembangan implementasi *platform* pengujian keamanan server berbasis *Website* ini terdapat tiga bagian utama yaitu *frontend*,*backend,* dan docker. Pada bagian *frontend* menggunakan HTM, CSS, dan JavaScript sebagai bahasa pemogramannya, sedangkan untuk *framework* dan l*ibrary frontend* menggunakan bootstrap, jQuery, *Font Awsome*, dan untuk *engine*-nya menggunakan *template* dari Django. Pada bagian *backend* menggunakan *framework* Django dengan bahasa pemrograman Python, serta integrasi *tools* Nmap dan Nikto sebagai inti dari aplikasi. Dan pada bagian Docker, platform ini menggunakan Docker untuk *containerization* seluruh komponen sistem (backend Django, Nmap, Nikto, dan server web) ke dalam microservice terpisah.

4.2 Detail Implementasi

Pada bagian detail implementasi ini merupakan paparan dari sistem yang telah dirancang sebelumnya. Terdapat 2 bagian pada detail implementasi ini yaitu *backend* dan *frontend*.

4.2.1 Backend

Backend merupakan bagian dari sistem aplikasi atau *Website* yang tidak terlihat oleh pengguna akhir (*end user*), yang berfungsi sebagai "mesin" di balik layar untuk memproses data, mengelola *database*, dan menangani logika aplikasi[15].

Berikut merupakan struktur *backend*, terdapat beberapa folder diantaranya *core*, nikto *scanner*, nmap *scanner*, scan *history*, *scanner project*. Pada folder tersebut terdapat file bawaan dari django diantaranya adalah:

File init.py, file ini khusus dalam python yang berfungsi sebagai “penanda” bahwa direktori tersebut adalah python *package*[9].

File admin.py adalah jantung dari Django admin *interface*, file ini berfungsi untuk mengelola data aplikasi. File ini berisi konfigurasi dan *customization* untuk Django *Admin panel*, yang secara otomatis menghasilkan *interface* CRUD (*Create, Read, Update, Delete*) berdasarkan model yang didefinisikan[9].

File apps.py, file ini berisi *class* konfigurasi aplikasi dan berfungsi sebagai pusat konfigurasi metadata aplikasi. Django menggunakan file ini untuk mengenali dan mengatur aplikasi dalam *registry* aplikasi global[9].

File models.py, file ini merupakan inti dari aplikasi django yang mendefinisikan struktur data dan kerangka kerja utama aplikasi melalui Django ORM (*Object-Relational Mapping*), file ini berisi definisi model-model yang mempresentasikan tabel *database* dan relasi antar tabel[9].

File test.py, file ini berfungsi untuk memastikan aplikasi berfungsi dengan benar. File ini menguji models, *views, forms*, dan berbagai komponen aplikasi secara otomatis[9].

File views.py, file yang berisikan *view function* atau *view classes* yang menangani HTTP *request* dan menghasilkan HTTP *response*. *Views* bertanggung jawab untuk memproses *request* data, berinteraksi dengan models, menjalankan cara kerja utama, dan mempersiapkan *context* data untuk *template* atau API *response*[9].

Ada beberapa aplikasi yang dibuat manual oleh penulis untuk menyesuaikan kebutuhan dalam pengembangan *platform*. Aplikasi tersebut dibuatkan di dalam

folder *platform*. Berikut beberapa *source code* tambahan yang disimpan dalam beberapa folder.

Bagian ini merupakan *source code* pada file *scoring*.py yang terletak di folder *core*. File ini merupakan file khusus yang dibuat untuk menangani logika perhitungan skor dalam penemuan kerentanan.

Folder *core*:

```
def calculate_security_score(scan_result):
def get_security_rating(score):
def get_score_color(score):
```

Berikut merupakan *source code* pada file *urls*.py yang terdapat di folder *core*, folder nikto_*scanner* dan nmap_*scanner* File ini berisi URLconf (URL *configuration*) yang mendefinisikan pola URL dan *Mapping* ke *view function* atau *classes*. File ini mencocokan URL dengan *regex* atau *path patterns*.

Folder *core* :

```
from django.urls import path
from . import views
urlpatterns = [
    # URL yang sudah ada - PERTAHANKAN
    path('', views.home_view, name='home'),
    path('about/', views.about_view, name='about'),
    path('contact/',views.contact_view,name='contac
t'),
    # URL UNTUK AUTHENTICATION
    path('dashboard/',
views.dashboard_view,name='dashboard'),
    path('profile/', views.profile_view,
name='profile'),
    path('logout/', views.custom_logout_view,
name='custom_logout'),
    # API URL
    path('api/save-scan/', views.save_scan_result,
name='save_scan_result'),
]
```

Folder nikto_*scanner* :

```
from django.urls import path
from . import views

urlpatterns = [
path('',views.nikto_scan_view,name='nikto_scan'),
]
```

Folder Nmap_*scanner* :

```
from django.urls import path
from . import views
urlpatterns = [
    path('',views.nmap_scan_view, name='nmap_scan'),
]
```

Folder Scan history :

```
from django.urls import path
from . import views

urlpatterns = [
```

Berikut ini merupakan file *context_processors*.py berisikan fungsi – fungsi *context processor* yang merupakan fitur *powerful* Django untuk menyediakan data yang tersedia di semua *template*. File ini merupakan fungsi python yang menerima Http*Request object* dan mengembalikan *dictionary* berisi variabel – variabel yang akan ditambahkan ke *context* setiap *template*.

Folder *core* :

```
def navigation(request):
    """
    Context processor untuk menentukan item navigasi
aktif
    """
    path = request.path.strip('/')

    # Default active state
nav_active = {
if path == '':
```

```
elif path.startswith('nmap'):
elif path.startswith('nikto'):
elif path.startswith('history'):
elif path.startswith('about'):
elif path.startswith('contact'):
return {'nav_active': nav_active}
```

Pada file forms.py ini merupakan file yang berfungsi untuk menangani semua aspek *form handling*, dari *rendering* HTML form hingga validasi data *server-side*.

Folder *core* :

```
# forms.py

from django import forms

from django.contrib.auth.models import User

from .models import UserProfile

class UserProfileForm(forms.ModelForm):
```

File utils.py merupakan modul utilitas dalam aplikasi ini yang terdapat pada folder nikto_*scanner* dan nmap_*scanner* yang berfungsi sebagai *layer* abstraksi antara aplikasi *web* Django dan *command-line security tools*. File ini berperan sebagai penghubung yang menghubungkan *interface* dengan *user* dengan *tools* yang dimiliki nmap dan nikto.

Pada Folder Nikto_*scanner* :

```
import subprocess
import json
import re
from core.models import ScanResult
def run_nikto_scan(target, scan_type, user):
def parse_nikto_output(output):
def aggressive_parse_nikto_output(output):
def classify_severity(description):
# Legacy functions kept for compatibility
def alternative_parse_nikto_output(output):
```

Pada folder nmap_*scanner* :

```
import nmap
import json
from core.models import ScanResult
def run_nmap_scan(target, scan_type, user):
def     classify_vulnerability_severity(script_name,
output):
```

File scan_extras.py adalah custom *templatetags* dan *filters* dalam django yang berfungsi untuk memperluas kemampuan *template engine* Django. File ini berisi *utilty functions* yang dapat digunakan langsung di dalam *template* HTML untuk melakukan operasi data *processing, mathematical calculations*, dan data *formating* yang tidak tersedia secara *default* di Django.

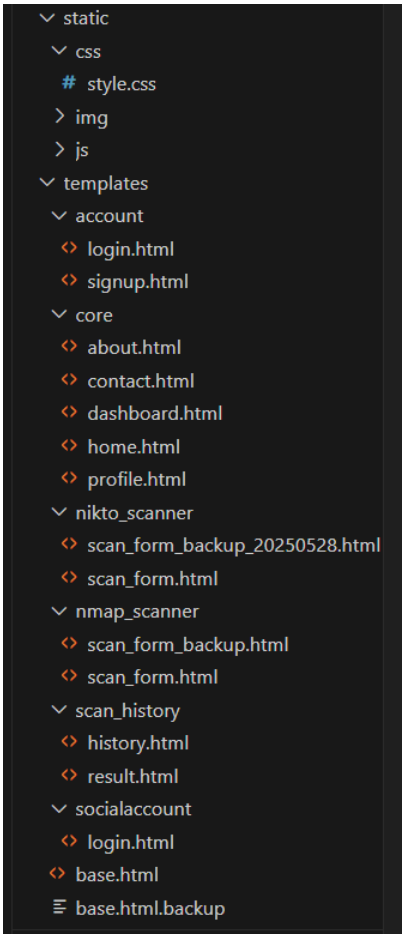
Folder *templatagestags* :

```
from django import template
from django.utils.safestring import mark_safe
import json
register = template.Library()
@register.filter
def jsonify(value):
@register.filter
def get_item(dictionary, key):
@register.filter
def mul(value, arg):
@register.filter
def div(value, arg):
@register.filter
def safe_div(value, arg):
```

4.2.2 Frontend

Frontend merupakan salah satu bagian dari sistem aplikasi atau *web* yang berperan untuk pengembangan bagian *Website* diantaranya desain *Website*, animasi *Website*, dan konten yang ditampilkan pada *Website* yang dapat dilihat dan berinteraksi langsung dengan pengguna[21].

Gambar 4.1 Struktur *Frontend* merupakan struktur file *frontend*, dimana pada folder *templates* menyimpan kode HTML untuk tampilan *Website* seperti struktur dasar halaman, tampilan seperti *heading*, dan *font default Website*. HMTL hanya bisa menghasilkan tampilan yang *basic* tanpa warna, terdapat juga folder CSS yang berisi style.css untuk mengatur tampilan *Website* seperti *font, layout, style*, dan mengatur urutan pada *Website* yang dapat yang dilihat *user*. Sedangkan folder js atau JavaScirpt didalamnya terdapat *source code* yang berfungsi untuk membuat *Website* dinamis dan interaktif, menangani interaksi *user* seperti klik, berpindah, dan mengetik, dan Mengubah halaman *Website* secara *realtime* sesuai dengan hasil temuan kerentanan.



Gambar 4. 1 Struktur *Frontend*

Berikut merupakan *source code* HTML *Structure* dan *Head Section* yang merupakan bagian awal dari file HMTL menggunakan *framework* Django digunakan untuk membuat antarmuka *Website*. Pada *source code* tersebut memuat file CSS dan JavaScript. Selain itu, pada *source code* ini menggunakan bootstrap untuk *styling*, *font awesome* untuk *icon* dan AOS (*Animate On Scroll*) untuk animasi pada halaman.

```
<!DOCTYPE html>
{% load static %}
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>{% block title %}Vulnerability Scanner{%
endblock %}</title>
    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <!-- Font Awesome -->
    <link                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">
    <!-- AOS - Animate On Scroll -->
    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css"
rel="stylesheet">
    <!-- Custom CSS -->
    <link rel="stylesheet" href="{% static 'css/style.css'
```

Berikut merupakan *source code* struktur navigasi utama yang menggunakan bootstrap sebagai *framework* CSS. Navbar ini berfungsi sebagai navigasi utama yang memungkinkan pengguna untuk berpindah antar halaman dengan mudah.

```
    <!-- Navbar -->
    <nav class="navbar navbar-expand-lg navbar-dark fixed-
top">
        <div class="container">
            <a class="navbar-brand" href="{% url 'home'
%}">
```

```
                    <i class="fas fa-shield-alt"></i>
                    Vulnerability Scanner
                </a>
    <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
                    <span class="navbar-toggler-icon"></span>
                </button>
      <div class="collapse navbar-collapse" id="navbarNav">
                    <ul class="navbar-nav me-auto">
                        {% if user.is_authenticated %}
                            <li class="nav-item">
      <li class="nav-item dropdown">
     <li class="nav-item">
      </li>
                        {% endif %}
     <li class="nav-item">
    </li>
    </ul>
    <!-- User Authentication Menu -->
     <ul class="navbar-nav">
```

Berikut merupakan *source code* pada bagian dari file CSS yang berfungsi untuk mengatur tampilan pada *platform* ini. Pada *source code* bagian *root* terdapat sejumlah variabel CSS seperti `–primary-color, --detikary-color, dan -danger-color` yang berfungsi untuk menetapkan warna warna utama yang digunakan pada *platform*. Terdapat juga *source code* bagian *body* diantaranya `padding-top; 56px;` yang berfungsi untuk memberikan jarak pada halaman. Terdapat juga *source code font* dan *color* untuk mengatur warna dan *font* yang digunakan.

```
root {
    --primary-color: #2c3e50;
    --detikary-color: #1abc9c;
    --danger-color: #e74c3c;
    --warning-color: #f39c12;
    --success-color: #2ecc71;
    --info-color: #3498db;
```

```
        --light-color: #ecf0f1;
        --dark-color: #34495e;
    }
    body {
        padding-top: 56px;
        font-family: 'Poppins', sans-serif;
        background-color: #f8f9fa;
        color: #333;
    }
```

Pada bagian ini merupakan *source code setupScanningForms* pada file Javascript yang berfungsi untuk melakukan proses validasi dan animasi saat pengguna melakukan pemindaian (*scanning*). Dengan adanya fungsi *setupScanningForms*() maka aplikasi akan mencari semua elemen yang memiliki kelas *.scan-form* dan melakukan validasi. Jika validasi berhasil maka akan ditampilkan animasi *loading* dengan menggunakan fungsi *showLoadingSpinner()* dengan keterangan bahwa proses *scanning* sedang berlangsung.

```
    function setup Scanning Forms() {
        const scanForms = 
document.querySelectorAll('.scan-form');

        scanForms.forEach(form => {
            form.addEventListener('submit', function(e) {
                // Validasi input
                const targetInput = 
form.querySelector('input[name="target"]');
                if (!targetInput.value.trim()) {
                    e.preventDefault();
                    showAlert('Target tidak boleh 
kosong!', 'danger');
                    return;
                }

                // Tampilkan animasi loading
                showLoadingSpinner('Scanning in 
progress... This may take a few minutes.');
            });
        });
    }
```

4.2.3 Docker

Berikut merupakan konfigurasi docker-compose.yml terbaru yang mendefinisikan beberapa service utama dalam sistem *Vulnerability Scanner*, yaitu: redis, web, nikto-scanner, dan nmap-scanner. Setiap service memiliki peran tersendiri yang saling terhubung melalui network vulnscanner_network.

```
redis:
  image: redis:7-alpine
  container_name: vulnscanner_redis
  volumes:
    - redis_data:/data
  healthcheck:
    test: ["CMD", "redis-cli", "ping"]
    interval: 30s
    timeout: 10s
    retries: 5
  restart: unless-stopped
  networks:
    - vulnscanner_network
```

Service redis berfungsi sebagai sistem caching dan task queue untuk aplikasi scanner. Redis digunakan untuk menyimpan data sementara secara efisien dan mengatur antrian tugas antar komponen. Service ini menggunakan image ringan redis:7-alpine, dengan volume redis_data untuk menyimpan data secara persisten. Terdapat juga konfigurasi healthcheck yang memastikan Redis berjalan dengan baik, serta network vulnscanner_network agar dapat berkomunikasi dengan container lain.

```
  web:
    build: .
    container_name: vulnscanner_web
    environment:
    volumes:
      - ./media:/app/media
      - ./static:/app/static
    ports:
      - "8000:8000"
    depends_on:
      redis:
        condition: service_healthy
    healthcheck:
                test:    ["CMD",    "curl",    "-f",
"http://localhost:8000/"]
      interval: 30s
      timeout: 10s
      retries: 3
    restart: unless-stopped
    networks:
      - vulnscanner_network
```

Service web merupakan *core* aplikasi berbasis Django yang dibangun langsung dari direktori proyek (build: .). Service ini mengatur berbagai environment variables penting seperti DATABASE_URL, REDIS_URL, dan DJANGO_SUPERUSER untuk pengaturan admin aplikasi. web terhubung dengan Redis, menyajikan web app melalui port 8000, dan menyimpan media serta file statis menggunakan volume lokal. Healthcheck digunakan untuk memastikan service aktif dengan mengakses URL lokal http://localhost:8000/.

```
    nikto-scanner:
      build: ./monitoring/nikto-worker
      container_name: nikto_scanner
      environment:
        - REDIS_URL=redis://redis:6379/0
        - WEBAPP_URL=http://vulnscanner_web:8000
        - WORKER_MODE=standby
      restart: unless-stopped
      depends_on:
        redis:
          condition: service_healthy
      networks:
        - vulnscanner_network
      healthcheck:
        test: ["CMD", "python", "-c", "import redis;
redis.Redis.from_url('redis://redis:6379/0').ping()"]
        interval: 30s
        timeout: 10s
        retries: 3
```

Service nikto-scanner merupakan *worker container* untuk menjalankan pemindaian keamanan berbasis Nikto. Service ini dibangun dari direktori ./monitoring/nikto-worker dan berjalan dalam mode siaga (standby) untuk menunggu tugas dari sistem. Komunikasi dilakukan melalui redis dan web app, tanpa membuka port publik karena hanya digunakan secara internal. Healthcheck memastikan koneksi redis aktif sebelum tugas dijalankan.

```
  nmap-scanner:
    build: ./monitoring/nmap-worker
    container_name: nmap_scanner
    environment:
      - REDIS_URL=redis://redis:6379/0
      - WEBAPP_URL=http://vulnscanner_web:8000
      - WORKER_MODE=standby
    restart: unless-stopped
    depends_on:
      redis:
        condition: service_healthy
    networks:
      - vulnscanner_network
    healthcheck:
      test: ["CMD", "python", "-c", "import redis;
redis.Redis.from_url('redis://redis:6379/0').ping()"]
      interval: 30s
      timeout: 10s
      retries: 3
```

Service nmap-scanner berfungsi serupa dengan nikto-scanner, namun menggunakan tool Nmap untuk melakukan pemindaian jaringan. Service ini dibangun dari direktori ./monitoring/nmap-worker dan juga berjalan dalam mode stand by. Konfigurasi depends_on memastikan bahwa Redis telah sehat sebelum container ini aktif, dan healthcheck memastikan konektivitas redis tersedia.

```
volumes:
  redis_data:
```

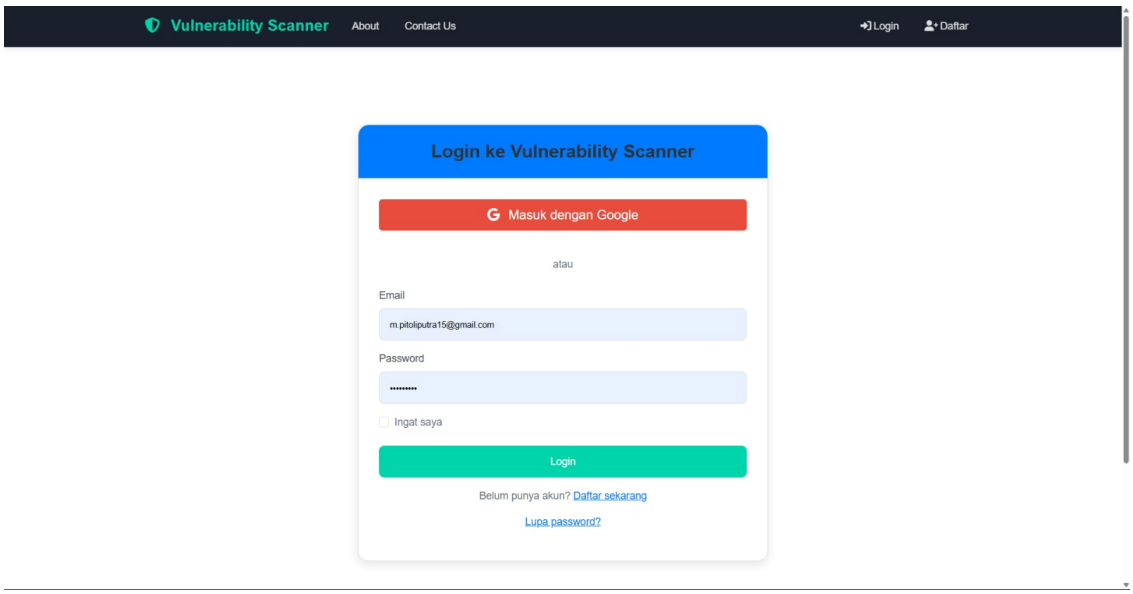
Volume redis_data digunakan untuk menyimpan data Redis secara persisten agar tetap tersedia meskipun container dimatikan atau di-*restart*. Hal ini menjaga stabilitas sistem caching dan task queue.

```
networks:
  vulnscanner_network:
    driver: bridge
```

Network vulnscanner_network menggunakan driver bridge untuk menghubungkan semua service dalam jaringan internal Docker. Konfigurasi ini memungkinkan setiap container saling berkomunikasi menggunakan nama servicenya sebagai hostname.

4.3 Prosedur Pengoperasian Solusi

Platform ini dapat diakses oleh pengguna melalui *browser* tanpa memerlukan instalasi tambahan, sehingga memudahkan pengguna dalam melakukan pengujian terhadap keamanan server secara mandiri.



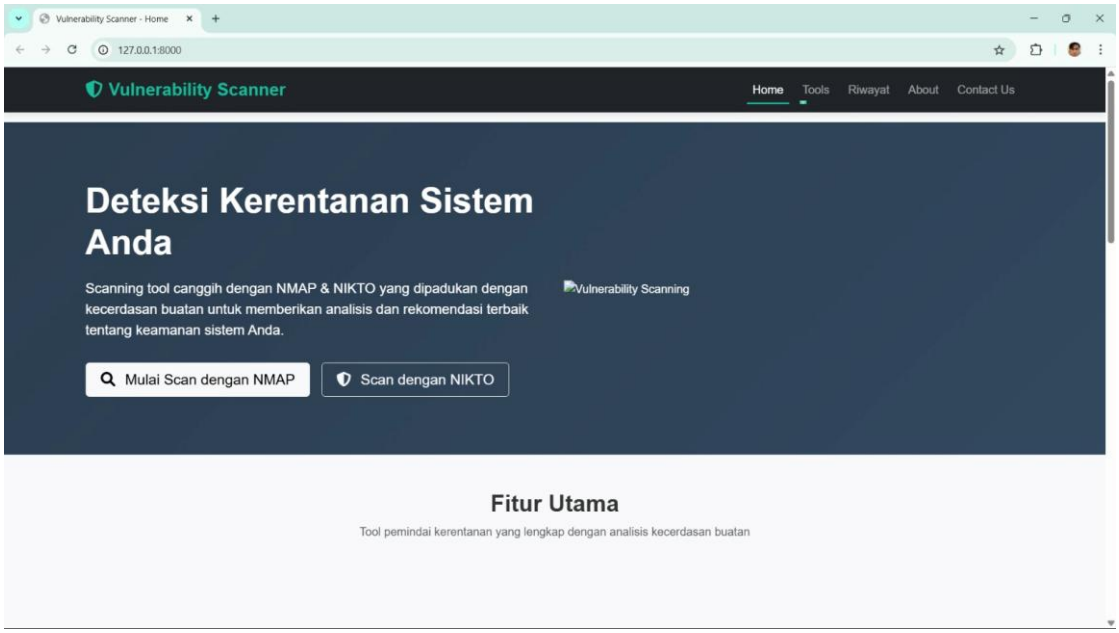
Gambar 4.2 *Login Page*

Setelah *platform* dapat diakses, langkah selanjutnya pengguna dapat melakukan *login* terlebih dahulu pada halaman *login page* apabila pengguna telah memiliki akun. Namun, apabila pengguna belum memiliki akun maka terlebih dahulu melakukan pendaftaran pada menu "Daftar Sekarang".

Gambar 4.3 *Menu Bar*

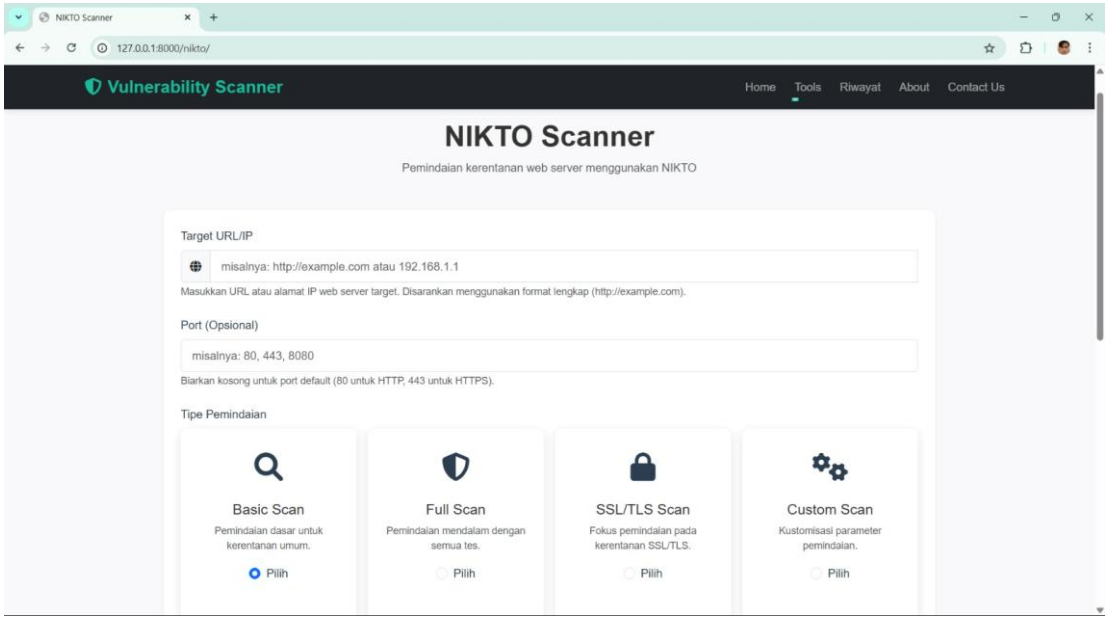
Gambar 4.3 Menu Bar merupakan tampilan menu bar pada *platfrom* ini, terdapat beberapa *menu* yang dapat diakses diantaranya "*Home*" yang mengarahkan pengguna ke halaman utama, *menu "Tools"* yang dapat digunakan oleh pengguna untuk mengakses berbagai fitur *scanning* dalam *platform*, *menu* "Riwayat" yang menampilkan catatan hasil pemindaian yang telah dilakukan, *menu "About"* yang memberikan informasi mengenai

platform ini dan *menu "Contact Us"* yang dapat digunakan oleh pengguna untuk menghubungi tim layanan pelanggan.



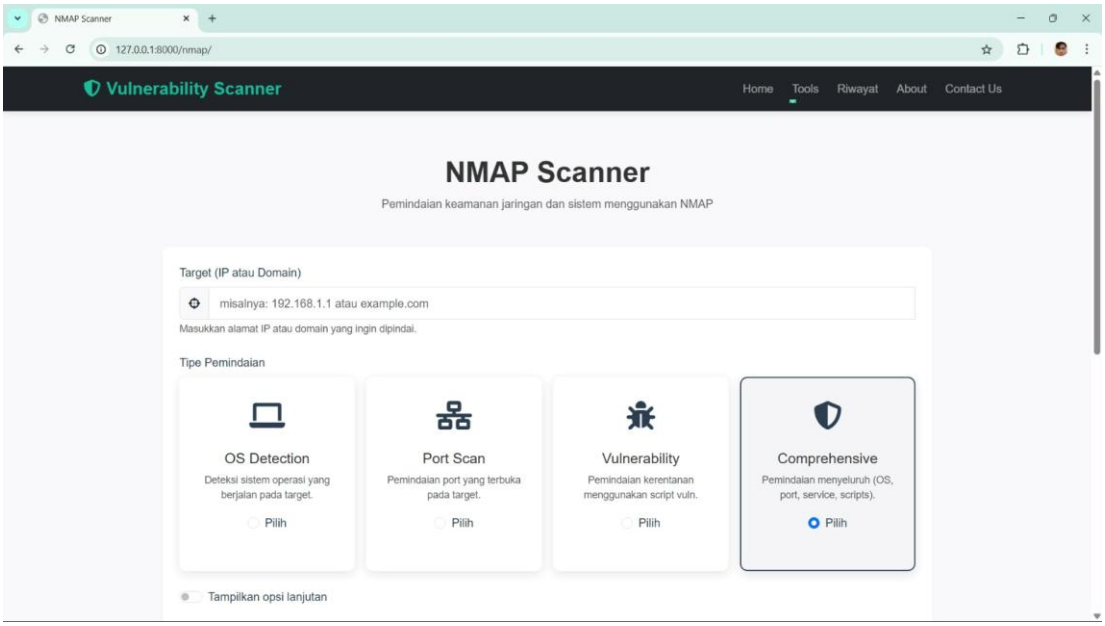
Gambar 4.4 *Home Page*

Gambar 4.4 *Home Page* merupakan halaman "*Home"* yang merupakan fitur utama dari *platform* ini. Pada *menu "home"* tersebut pengguna dapat memilih melakukan *scanning* menggunakan nikto atau nmap.



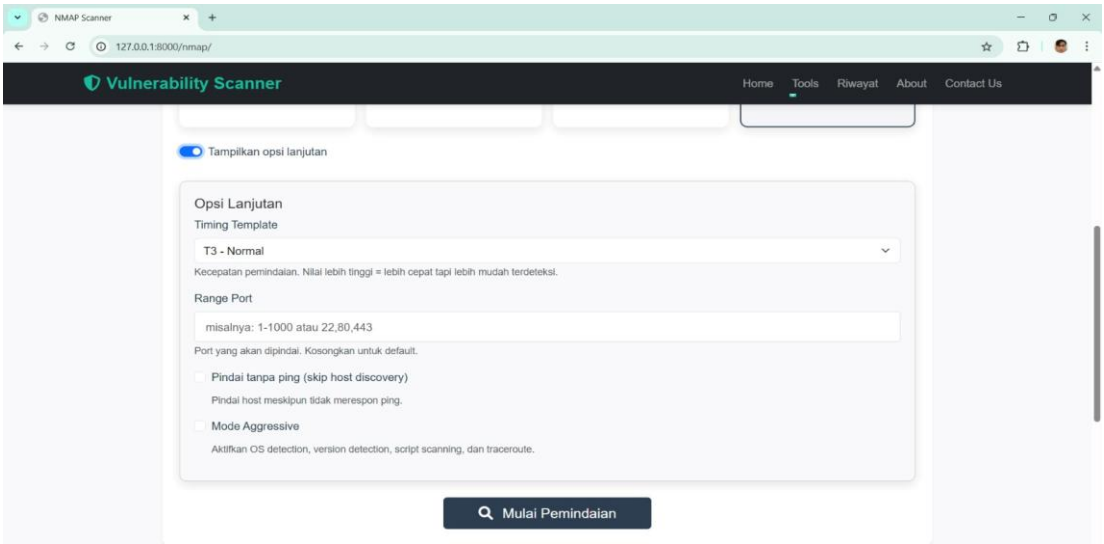
Gambar 4.5 Fitur-fitur Nikto *Scanner*

Gambar 4.5 Fitur-fitur Nikto *Scanner* merupakan *page* dari fitur Nikto *Scanner*, pada *page* ini pengguna dapat memasukan target URL/IP dan *Port*. Pengguna dapat memilih tipe pemindaian yang ingin dilakukan, terdapat beberapa tipe diantaranya *Basic Scan*, *Full Scan*, SSL/TLS scan dan *custom scan*.



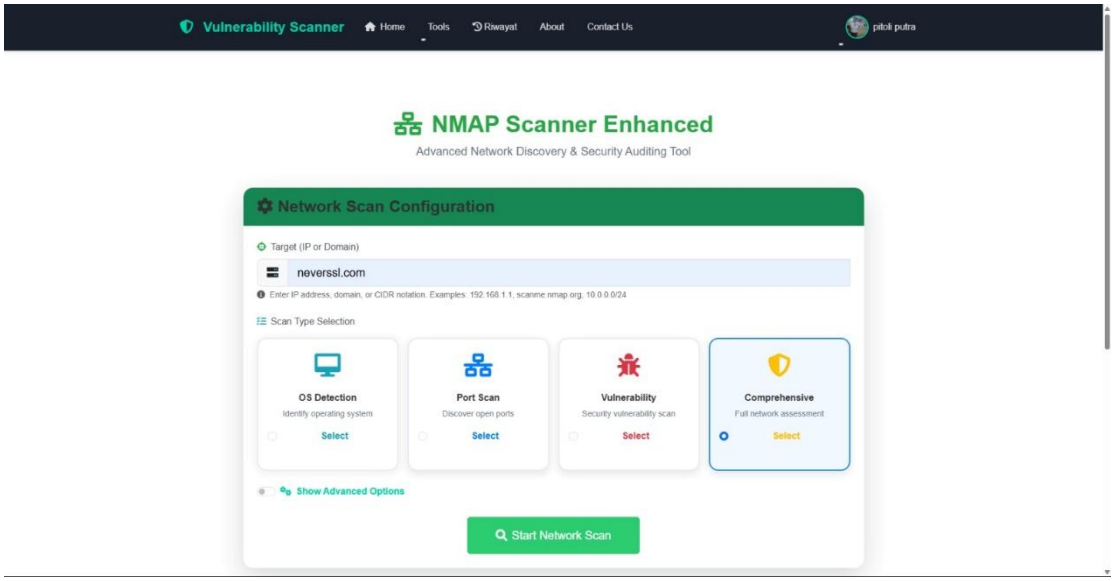
Gambar 4.6 Fitur-fitur Nmap *Scanner*

Gambar 4.6 Fitur-fitur Nmap *Scanner* merupakan *page* dari fitur Nmap *Scanner* pada *platform* pengujian ini. Pada *page* ini pengguna dapat memasukan target IP atau *domain* lalu memilih tipe pemindaian yang ingin dilakukan, terdapat beberapa tipe diantaranya *OS Detection, Port Scan, Vulnerability* dan *Comperhensive*.



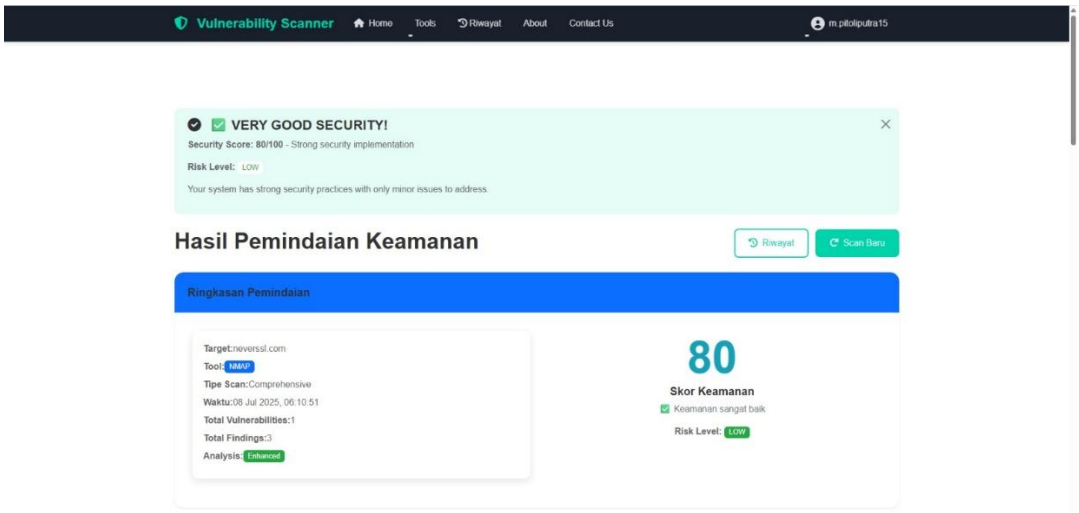
Gambar 4.7 Opsi Lanjutan Nmap *Scanner*

Pada Nmap *scanner* ini pengguna juga dapat menggunakan opsi lanjutan yang berisi pilihan waktu pemindaian, terdapat juga pilihan *range port* yang akan dipindai apabila pengguna hanya ingin memindai salah satu *port* saja, terdapat juga pilihan "pindai tanpa *ping*" serta pilihan "mode *aggressive*".



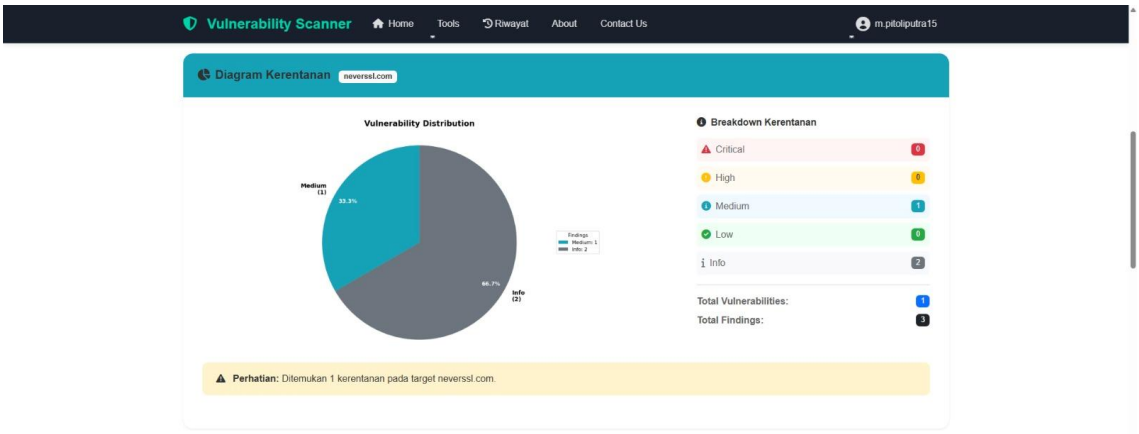
Gambar 4.8 Fitur *Port Scan*

Gambar 4.8 Fitur *Port scan* merupakan contoh cara menggunakan *Comperhensive Scan* pada Nmap *Scanner*. Pengguna dapat memasukan IP pada *menu* Target (IP atau *Domain*) dan memilih opsi *Comperhensive Scan* dan dapat mengklik tombol mulai pemindaian untuk memulai pengujian keamanan.



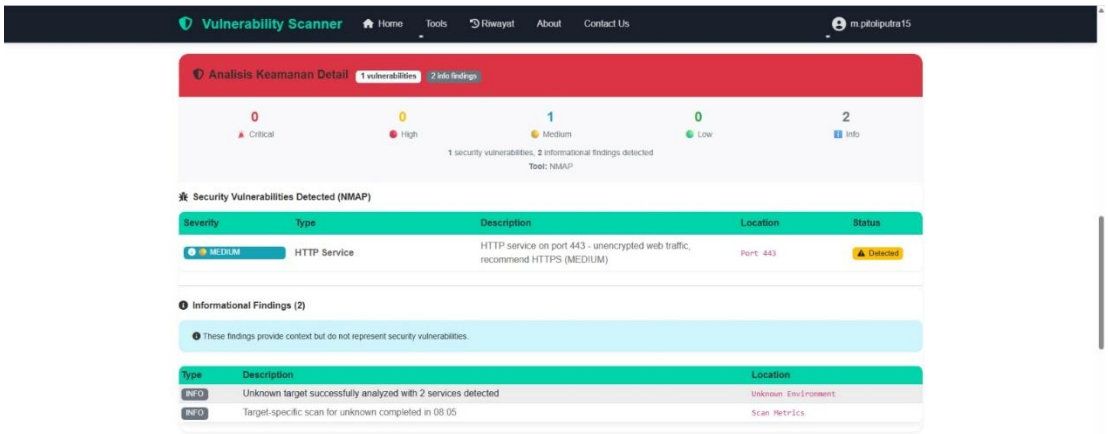
Gambar 4.9 Hasil Pemindaian

Setelah pemindaian selesai dilakukan, *platform* akan memunculkan halaman hasil pemindaian seperti pada Gambar 4.9 Hasil Pemindaian Pada halaman hasil pemindaian ini pengguna dapat melihat ringkasan pemindaian dan skor keamanan.



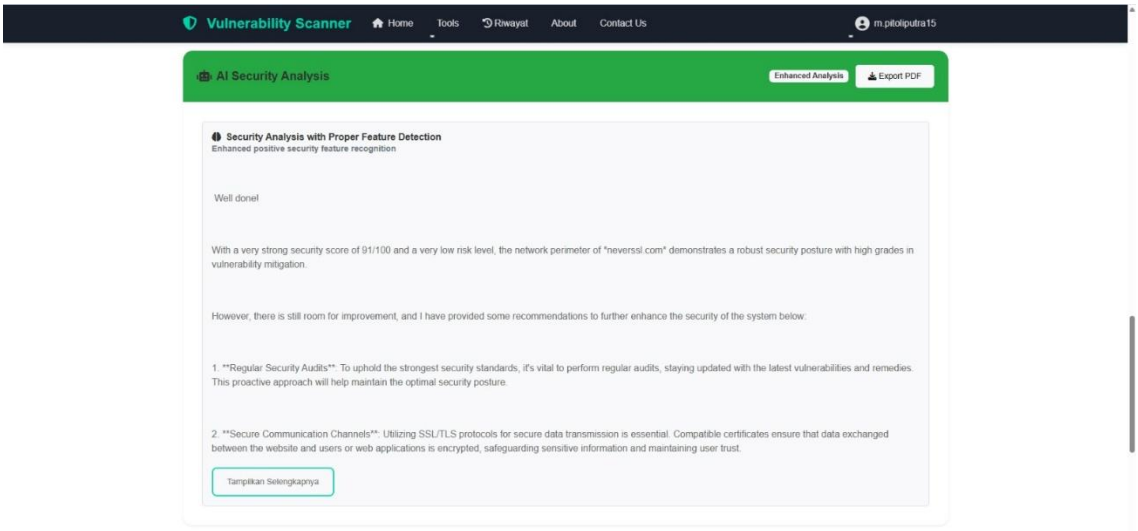
Gambar 4. 10 Diagram Kerentanan

Pada halaman hasil pemindaian juga pengguna dapat melihat diagram kerentanan seperti pada Gambar 4. 10 sesuai dengan kategori temuan kerentanan.



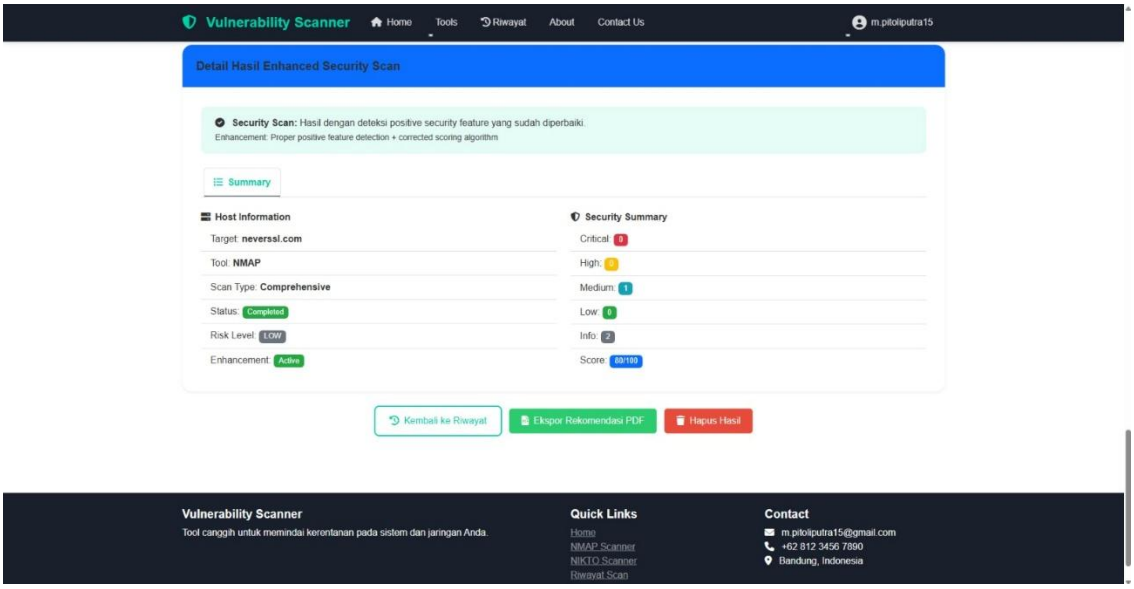
Gambar 4.11 Analisis Keamanan Detail

Pada halaman hasil pemindaian, pengguna dapat melihat analisis keamanan detail seperti pada gambar 4. 11, Terdapat ringkasan temuan kerentanan yang dikelompokan sesuai dengan kategori temuan yaitu critical, high, medium, low dan info. Pengguna dapat melihat detail type kerentanan apa saja yang ditemukan dan juga terdapat informational finding yang memberikan info konteks namun bukan kerentanan keamanan.



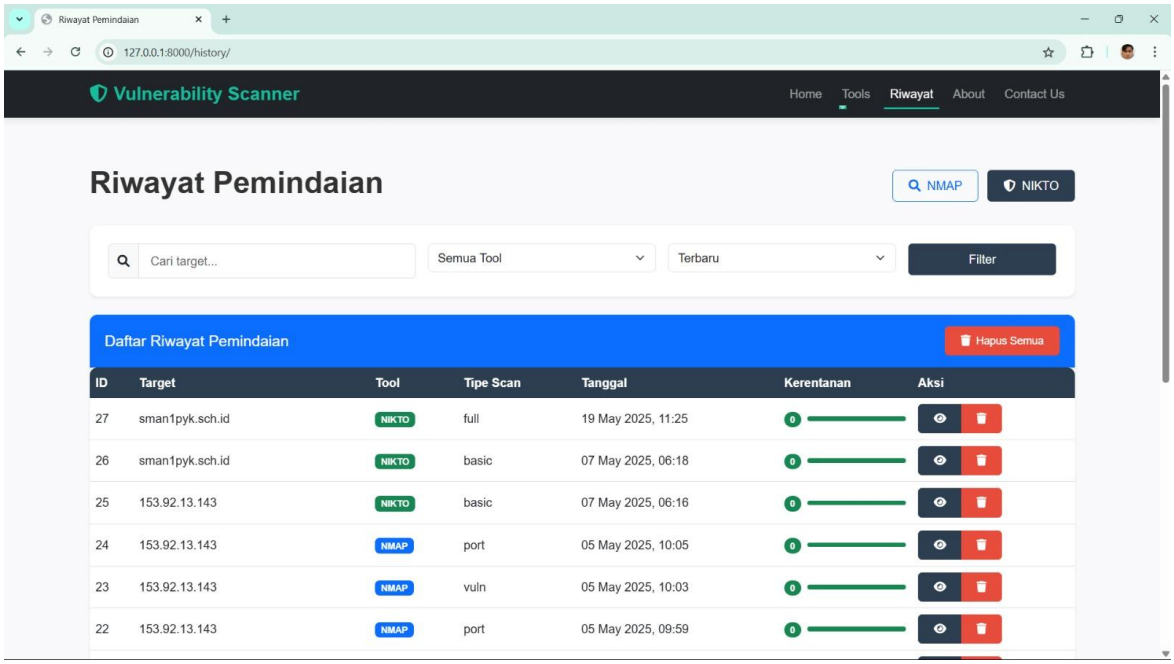
Gambar 4.12 Rekomendasi Perbaikan Berbasis *Generative* AI

Pada halaman hasil pemindaian juga pengguna dapat melihat rekomendasi perbaikan berbasis *generative* AI seperti Gambar 4.10 Rekomendasi Perbaikan Berbasis *Generative AI* sehingga pengguna dapat mengetahui perbaikan seperti apa yang perlu dilakukan terhadap target yang telah dipindai.



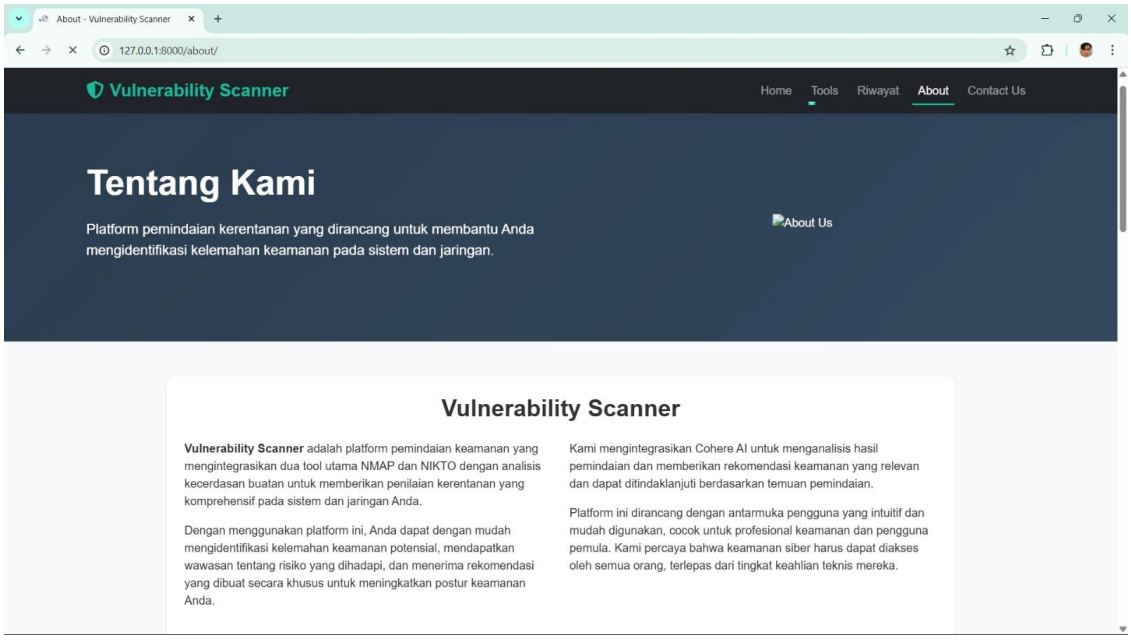
Gambar 4.13 Unduh Ringkasan Pemindaian

Gambar 4.13 merupakan menu detail hasil pemindaian. Pada *menu* ini pengguna dapat mengunduh hasil pemindaian tersebut dalam bentuk file PDF.



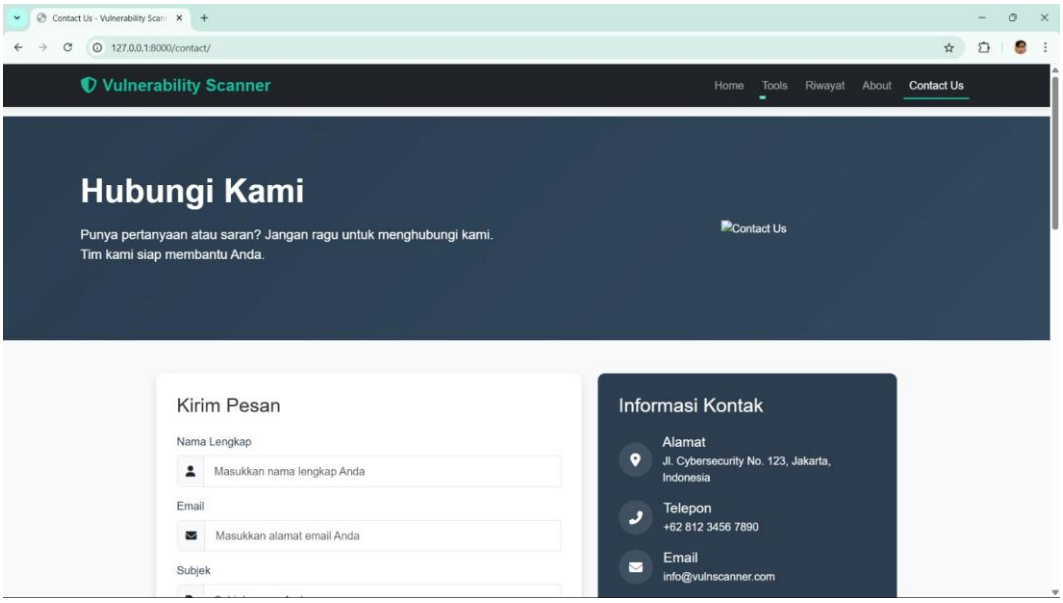
Gambar 4.14 *Menu* Riwayat Pemindaian

Gambar 4.12 *Menu* Riwayat Pemindaian Menampilkan *menu* "Riwayat pemindaian", pada halaman ini pengguna dapat mengakses pemindaian yang telah dilakukan dan melihat riwayat hasil pemindaian pada *menu* aksi. Pengguna juga dapat menghapus riwayat pemindaian apabila sudah tidak diperlukan lagi.



Gambar 4.15 *Menu About US*

Gambar 4.13 *Menu About US* menjelaskan bahwa *platform* ini dirancang untuk membantu pengguna dalam mengidentifikasi kelemahan keamanan pada sistem dan jaringan.



Gambar 4.16 *Menu Contact Us*

Gambar 4.14 *Menu Contact Us* merupakan yang dapat digunakan oleh pengguna dalam menyampaikan pertanyaan, saran, maupun kendala yang pengguna alami.

BAB 5
PENGUJIAN

5.1 Skema Pengujian Sistem

Pengujian ini bertujuan untuk mengumpulkan data tentang kondisi keamanan server melalui proses *scanning*. Pengujian ini dilakukan untuk mengidentifikasi dan mendokumentasikan celah keamanan pada server yang memungkinkan untuk diekspoitasi oleh penyerang. Selain itu, pengujian ini juga bertujuan untuk memvalidasi efektivitas *Website* yang sudah dibuat dalam mendeteksi kerentanan yang ada. Lokasi pengujian dilakukan di Laboratorium Elektronika Komunikasi, Telkom University dan waktu pengujian dilakukan pada hari Senin, 7 Juli 2025.

5.1.1 Daftar Pengujian

Pengujian yang akan dilakukan berfokus pada beberapa aspek penting yang diperlukan untuk mendapatkan hasil maksimal. Maka, aspek aspek yang akan diuji adalah sebagai berikut:

- Waktu Proses *Scanning*,
- Konsistensi hasil temuan kerentanan,
- Pengujian Komparasi Nmap dan Nikto,
- Validasi hasil rekomendasi perbaikan,
- Pengujian performa backend, frontend dan platform pengujian keamanan server menggunakan apache Jmeter,
- monitoring penggunaan CPU dan memori pada sistem backend menggunakan *tools* HTOP, dan
- Pengujian performa docker *container*

5.1.2 Skenario Pengujian

Dalam skenario pengujian ini, dilakukan pengujian menggunakan dua *tools* utama yang ada pada *website* vulnerability scanning yaitu *tools* Nikto dan *tools* Nmap. Pada masing – masing *tools* disediakan 3 jenis *website* untuk dijadikan target pengujian yaitu *website* localhost 1 device, *website* localhost dari device lain, dan pengujian *website* dari internet. Pada ketiga *website* tersebut dilakukan masing – masing 3 kali percobaan untuk melihat konsistensi dari hasil temuan kerentanan *website* tersebut. Pengujian performa dilakukan terhadap backend, frontend, dan platform pengujian

keamanan server menggunakan Apache JMeter. Selain itu, pemantauan performa sistem backend dilakukan menggunakan tools HTOP, dengan metode observasi selama 5 menit dan pencatatan lima data dari kondisi minimum hingga maksimum. Monitoring ini mencakup tiga kondisi: website dalam keadaan normal, saat proses pemindaian dengan Nikto, dan saat pemindaian dengan Nmap. Untuk menguji performa container, dilakukan pula pemantauan melalui Docker Desktop guna melihat penggunaan CPU dan memori dari masing-masing layanan Docker saat proses berjalan.

5.2 Proses Pengujian dan Analisis Hasil

Bagian ini merupakan detail proses pengujian dan hasil analisis yang akan dijabarkan pada bagian-bagian berikut.

5.2.1 Pengujian Menggunakan Nikto *Scanner*

Pengujian menggunakan Nikto *scanner* ini mencakup 3 metode dengan menggunakan 3 *Website* per masing – masing metodenya.

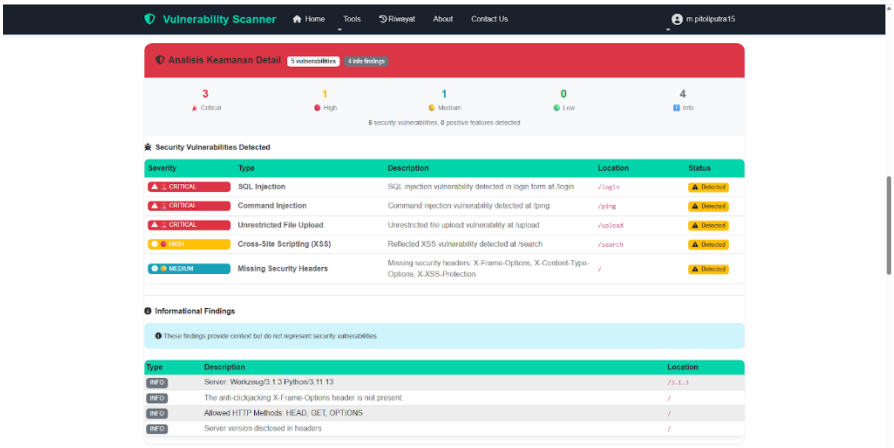
5.2.1.1 Pengujian *Website Localhost* 1 *Device*

Pada pengujian *Website localhost* 1 *device* menggunakan 3 buah *Website* sebagai berikut :

a. *Website* 1 (localhost:5001)

- Pengujian Pertama :

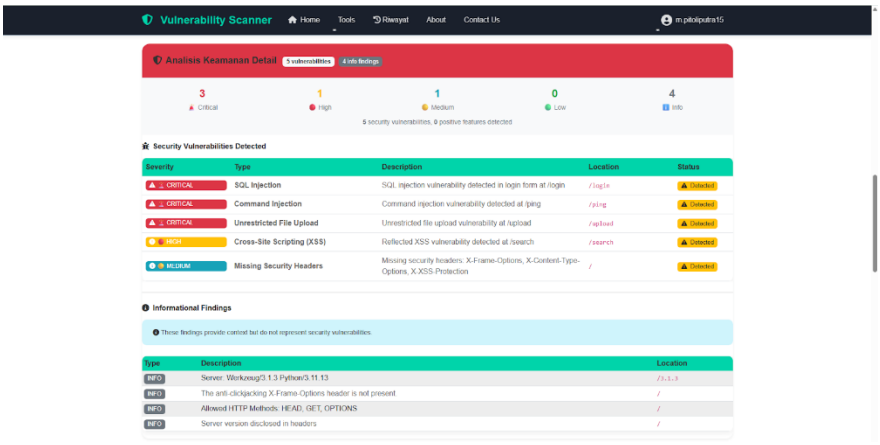
Gambar 5.1 merupakan hasil pengujian ketiga *Website* localhost:5001 dengan durasi *scanning* selama 1.92 Detik yang menunjukkan hasil analisis temuan celah keamanan detail yaitu 3 kerentanan kategori *critical* adalah *SQL Injection*, *Command Injection* dan *Unrestricted* File *Upload*. 1 kerentanan kategori *high* yaitu XSS (*Cross-Site Scripting*). 1 kerentanan kategori *medium* yaitu *Missing Security Headers*, 0 kerentanan kategori *low* dan 4 kategori info. Berdasarkan temuan kerentanan tersebut skor keamanan *Website* localhost:5001 adalah 0 yang berarti bahwa *Website* ini sangat rentan terhadap serangan.



Gambar 5.1 Pengujian Pertama *Website* 1 *Device* (Localhost:5001) dengan Nikto *scanner*

- Pengujian Kedua :

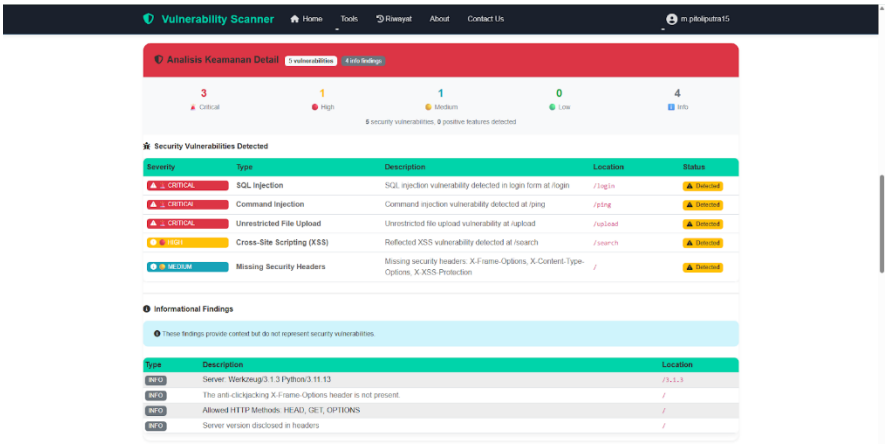
Gambar 5.2 merupakan hasil pengujian ketiga *Website* localhost:5001 dengan durasi *scanning* selama 1.73 Detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 3 kerentanan kategori *critical* adalah *SQL Injection*, *Command Injection* dan *Unrestricted* File *Upload*. 1 kerentanan kategori *high* yaitu XSS (*Cross-Site Scripting*). 1 kerentanan kategori *medium* yaitu *Missing Security Headers*, 0 kerentanan kategori *low* dan 4 kategori info. Berdasarkan temuan kerentanan tersebut skor keamanan *Website* localhost:5001 adalah 0 yang berarti bahwa *Website* ini sangat rentan terhadap serangan.



Gambar 5.2 Pengujian kedua *Website* 1 *device* (Localhost:5001) dengan Nikto *Scanner*

- Pengujian Ketiga :

Gambar 5.3 merupakan hasil pengujian ketiga *Website* localhost:5001 dengan durasi *scanning* selama 1.67 Detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 3 kerentanan kategori *critical* adalah SQL Injection, Command Injection dan Unrestricted File Upload. 1 kerentanan kategori *high* yaitu XSS (Cross-Site Scripting). 1 kerentanan kategori *medium* yaitu Missing *Security* Headers, 0 kerentanan kategori *low* dan 4 kategori info. Berdasarkan temuan kerentanan tersebut skor keamanan *Website* localhost:5001 adalah 0 yang berarti bahwa *Website* ini sangat rentan terhadap serangan.



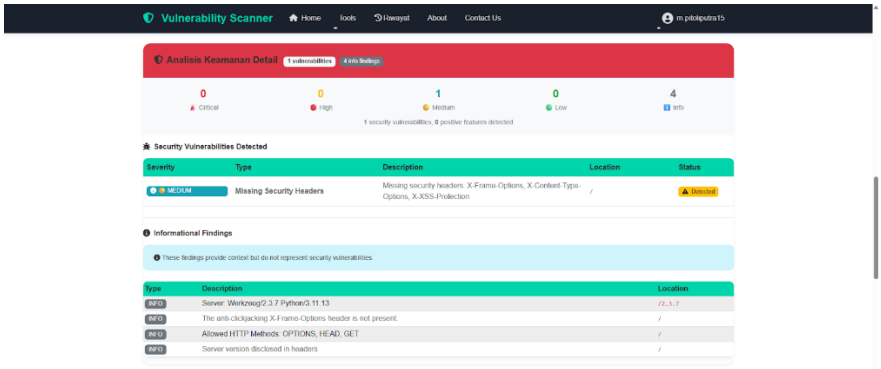
Gambar 5.3 Pengujian ketiga *Website* 1 *device* (Localhost:5001) dengan Nikto *Scanner*

b. *Website* 2(localhost:5002)

- Pengujian Pertama :

Gambar 5.4 merupakan hasil pengujian pertama pada *Website* localhost:5002 dengan durasi *scanning* selama 1.03 detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 0 kategori *critical*, 0 kategori *high*, 1 kategori *medium* yaitu Missing *Security* Headers, 0 kategori *low* dan 4 kategori info. Berdasarkan temuan kerentanan tersebut skor keamanan *Website* localhost:5002 adalah 80 yang berarti

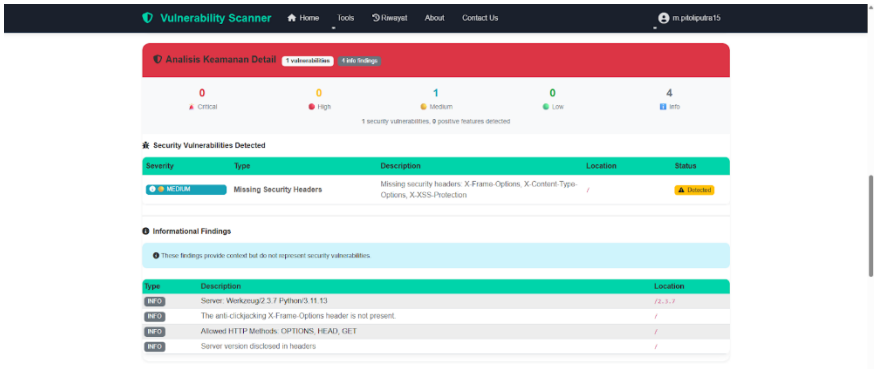
bahwa *Website* ini cukup aman dari serangan namun perlu adanya perbaikan.



Gambar 5.4 Pengujian pertama *Website* 1 *device* (Localhost:5002) dengan Nikto *Scanner*

- Pengujian Kedua :

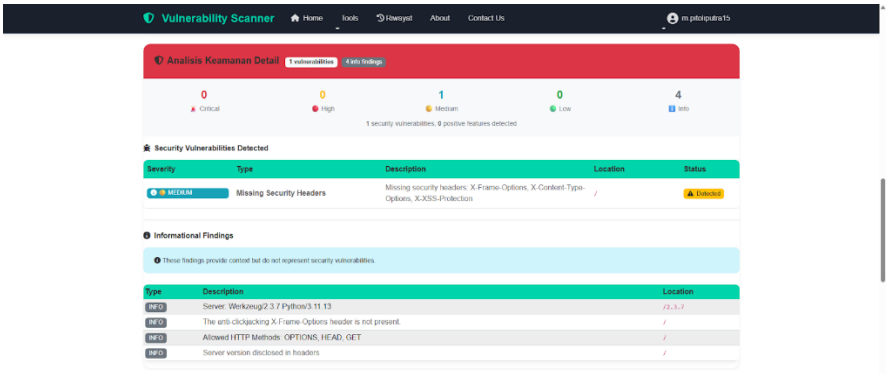
Gambar 5.5 merupakan hasil pengujian keduaa pada *Website* localhost:5002 dengan durasi *scanning* selama 1.08 detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 0 kategori *critical*, 0 kategori *high*, 1 kategori *medium* yaitu Missing *Security* Headers, 0 kategori *low* dan 4 kategori info. Berdasarkan temuan kerentanan tersebut skor keamanan *Website* localhost:5002 adalah 80 yang berarti bahwa *Website* ini cukup aman dari serangan namun perlu adanya perbaikan.



Gambar 5.5 Pengujian kedua *Website* 1 *device* (Localhost:5002) dengan Nikto *Scanner*

- Pengujian Ketiga :

Gambar 5.6 merupakan Hasil pengujian ketiga pada *Website* localhost:5002 dengan durasi *scanning* selama 0.85 detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 0 kategori *critical*, 0 kategori *high*, 1 kategori *medium* yaitu Missing *Security* Headers, 0 kategori *low* dan 4 kategori info. Berdasarkan temuan kerentanan tersebut skor keamanan *Website* localhost:5002 adalah 80 yang berarti bahwa *Website* ini cukup aman dari serangan namun perlu adanya perbaikan.

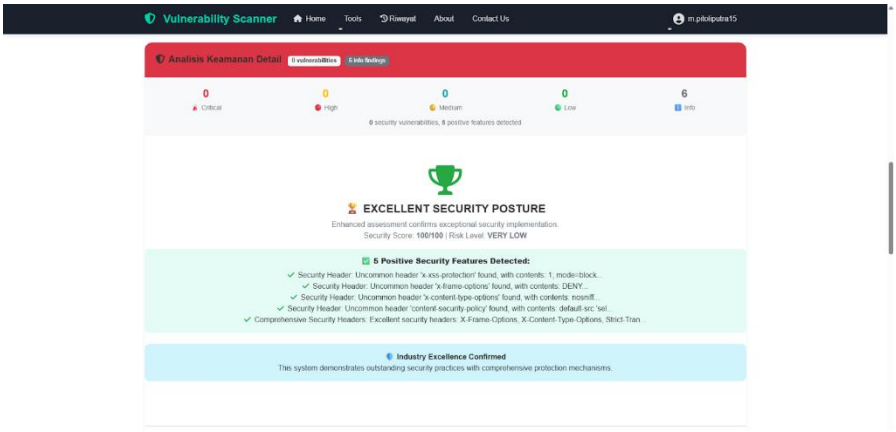


Gambar 5.6 Pengujian ketiga *Website* 1 *device* (Localhost:5002) dengan Nikto *Scanner*

c. *Website* 3 (localhost:5003)

- Pengujian pertama :

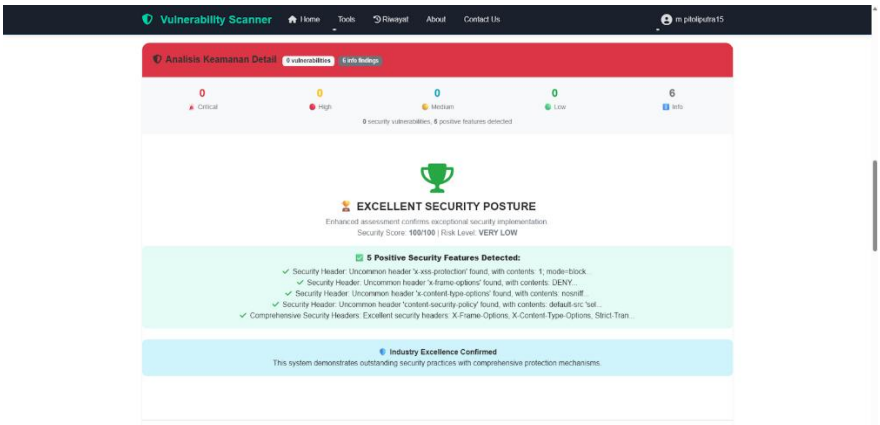
Gambar 5.7 merupakan Hasil dari pengujian pertama pada *Website* localhost:5003 dengan durasi *scanning* selama 0.94 detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 0 kategori *critical*, 0 kategori *high*, 0 kategori meidum, 0 kategori *low* dan 6 kategori info. Berdasarkan hasil temuan tersebut didapatkan *security score Website* localhost:5003 adalah 100 yang menunjukkan tingkat risiko serangan sangat rendah.



Gambar 5.7 Pengujian pertama *Website* 1 *device* (Localhost:5003) dengan Nikto *Scanner*

- Pengujian kedua :

Gambar 5.8 merupakan merupakan hasil pengujian kedua pada *Website* localhost:5003 dengan durasi *scanning* selama 1.25 detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 0 kategori *critical*, 0 kategori *high*, 0 kategori meidum, 0 kategori *low* dan 6 kategori info. Berdasarkan hasil temuan tersebut didapatkan *security score Website* localhost:5003 adalah 100 yang menunjukkan tingkat risiko serangan sangat rendah.

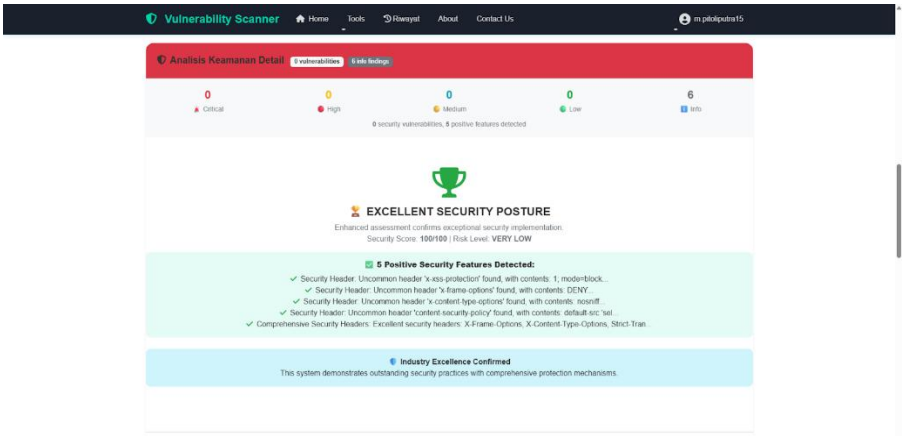


Gambar 5.8 Pengujian kedua *Website* 1 *device* (Localhost:5003) dengan Nikto *Scanner*

- Pengujian ketiga :

Gambar 5.9 merupakan merupakan hasil pengujian kedua pada *Website* localhost:5003 dengan durasi *scanning* selama 1.46 detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 0 kategori *critical*, 0

kategori *high*, 0 kategori meidum, 0 kategori *low* dan 6 kategori info. Berdasarkan hasil temuan tersebut didapatkan *security score Website* localhost:5003 adalah 100 yang menunjukkan tingkat risiko serangan sangat rendah.



Gambar 5.9 Pengujian ketiga *Website* 1 *device* (Localhost:5003) dengan Nikto *Scanner*

Tabel 5.1 Pengujian *Website* Localhost 1 *Device* Menggunakan Nikto *Scanner*

No.	Waktu Pengujian			Konsistensi Hasil Temuan			Keterangan
	P1	P 2	P 3	P 1	P 2	P 3	
Website 1	1.92 detik	1.73 detik	1.67 detik	Skor kerentanan 0	Skor kerentanan 0	Skor kerentanan 0	Hasil konsisten
Website 2	1.03 detik	1.08 detik	0.85 detik	Skor kerentanan 80	Skor kerentanan 80	Skor kerentanan 80	Hasil konsisten
Website 3	0.94 detik	1.25 detik	1.46 detik	Skor kerentanan 100	Skor kerentanan 100	Skor kerentanan 100	Hasil konsisten

5.2.1.2 Pengujian *Website* Localhost dari *Device* yang Berbeda

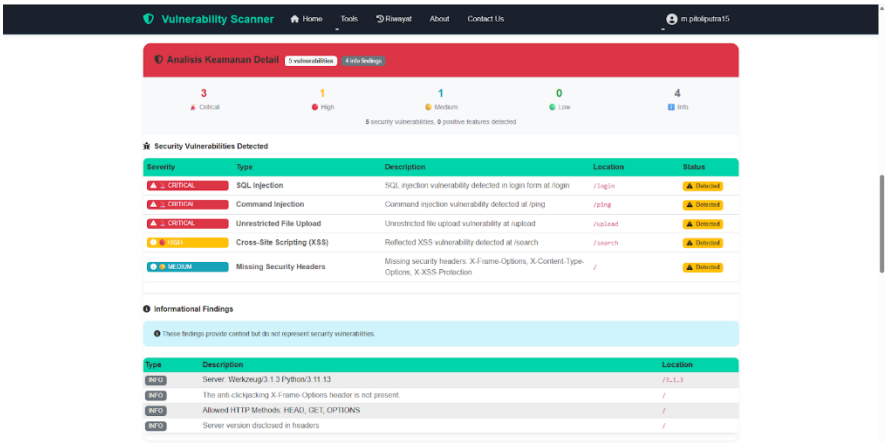
Pengujian *Website* localhost dari *device* yang berbeda menggunakan 3 jenis *Website* sebagai berikut :

a. *Website* 1 (loaclhost:5001)

- Pengujian Pertama :

  Gambar 5.10 merupakan merupakan hasil pengujian pertama pada *Website* localhost:5001 dengan durasi *scanning* selama 10.51 detik

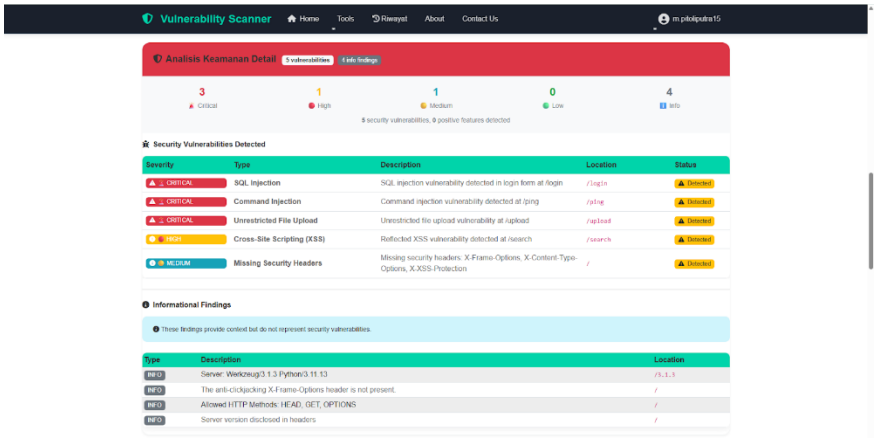
menunjukkan hasil analisis temuan celah keamanan detail yaitu 3 kategori *critical* diantaranya SQL Injection, command Injection dan Unrestriced File Upload. 1 kategori *high* yaitu Cross-Site Scripting (XSS). 1 kategori *medium* yaitu missing *security* headers. 0 kategori *low* dan 4 kategori info. Dari hasil termuan tersebut didapatkan *security score Website* localhost:5001 adalah 0 yang menandakan *Website* tersebut sangat rentan terhadap serangan.



Gambar 5.10 Pengujian pertama *Website* beda *device* (Localhost:5001) dengan Nikto *Scanner*

- Pengujian Kedua :

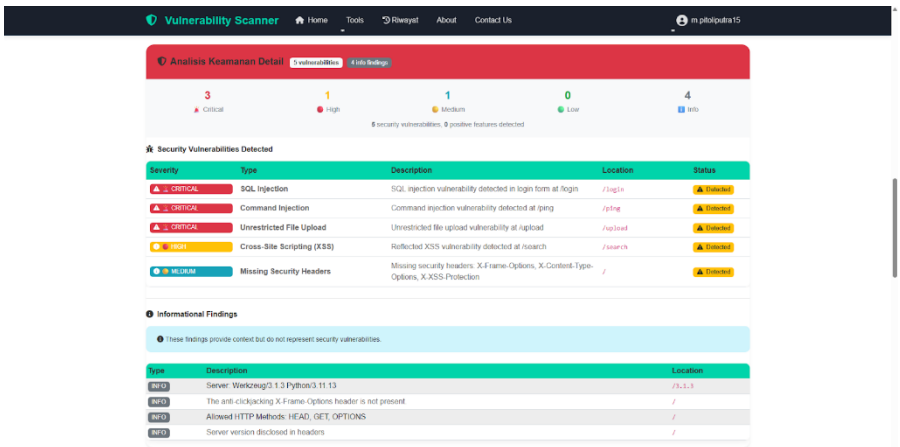
Gambar 5.11 merupakan hasil pengujian kedua pada *Website* localhost:5001 dengan durasi *scanning* selama 9.71 detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 3 kategori *critical* diantaranya SQL Injection, command Injection dan Unrestriced File Upload. 1 kategori *high* yaitu Cross-Site Scripting (XSS). 1 kategori *medium* yaitu missing *security* headers. 0 kategori *low* dan 4 kategori info. Dari hasil termuan tersebut didapatkan *security score Website* localhost:5001 adalah 0 yang menandakan *Website* tersebut sangat rentan terhadap serangan.



Gambar 5.11 Pengujian kedua *Website* beda *device* (Localhost:5001) dengan Nikto *Scanner*

- Pengujian Ketiga :

Gambar 5.12 merupakan hasil pengujian ketiga pada *Website* localhost:5001 dengan durasi *scanning* selama 10.14 detik menunjukkan hasil analisis temuan celah keamanan detail yaitu 3 kategori *critical* diantaranya SQL Injection, command Injection dan Unrestriced File Upload. 1 kategori *high* yaitu Cross-Site Scripting (XSS). 1 kategori *medium* yaitu missing *security* headers. 0 kategori *low* dan 4 kategori info. Dari hasil termuan tersebut didapatkan *security score Website* localhost:5001 adalah 0 yang menandakan *Website* tersebut sangat rentan terhadap serangan.

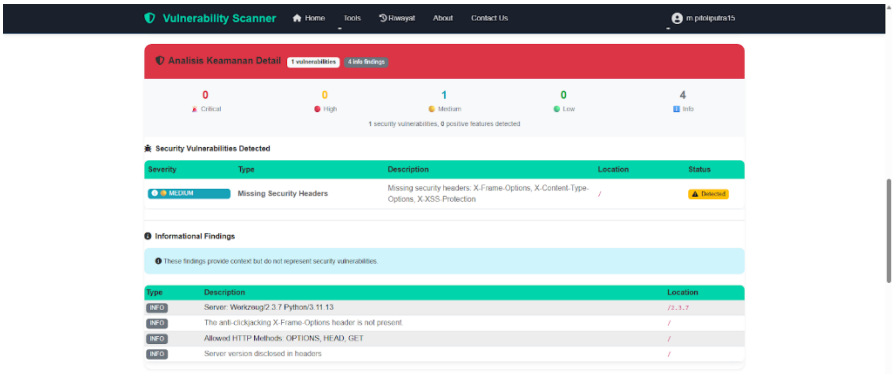


Gambar 5.12 Pengujian ketiga *Website* beda *device* (Localhost:5001) dengan Nikto *Scanner*

b. *Website* 2 (localhost:5002)

- Pengujian pertama :

Gambar 5.13 merupakan Hasil pengujian pertama pada *Website* localhost:5002 dengan durasi *scanning* selama 8.23 detik, menunjukkan hasil temuan celah keamanan berupa 0 kategori *critical*, 0 kategori *high*, 1 kategori *medium* yaitu missing *security* header, 0 kategori *low* dan 4 kategori info. Dari hasil temuan tersebut didapatkan *security score Website* ini adalah 80 yang menunjukkan bahwa *Website* ini sudah cukup aman namun perlu dilakukan perbaikan untuk memaksimalkan *Website* tersebut.

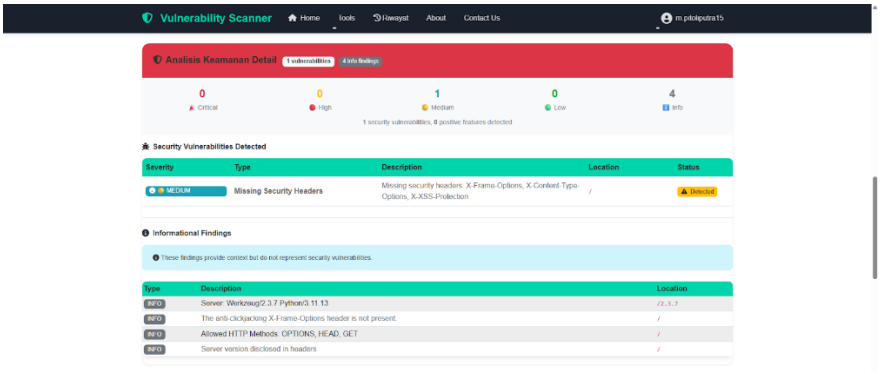


Gambar 5.13 Pengujian pertama *Website* beda *device* (Localhost:5002) dengan Nikto *Scanner*

- Pengujian kedua :

Gambar 5.14 merupakan hasil pengujian kedua pada *Website* localhost:5002 dengan durasi *scanning* selama 6.12 detik, menunjukkan hasil temuan celah keamanan berupa 0 kategori *critical*, 0 kategori *high*, 1 kategori *medium* yaitu missing *security* header, 0 kategori *low* dan 4 kategori info. Dari hasil temuan tersebut didapatkan *security score Website* ini adalah 80 yang menunjukkan bahwa *Website* ini sudah cukup

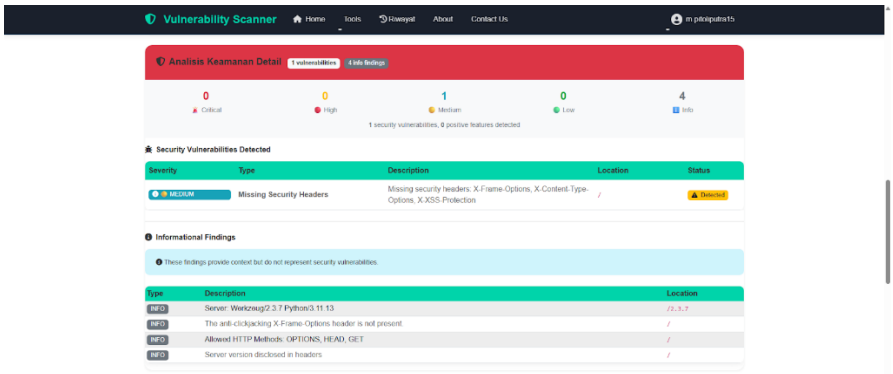
aman namun perlu dilakukan perbaikan untuk memaksimalkan *Website* tersebut.



Gambar 5.14 Pengujian kedua *Website* beda *device* (Localhost:5002) dengan Nikto *Scanner*

- Pengujian ketiga :

Gambar 5.15 merupakan hasil pengujian ketiga pada *Website* localhost:5002 dengan durasi *scanning* selama 11.82 detik, menunjukkan hasil temuan celah keamanan berupa 0 kategori *critical*, 0 kategori *high*, 1 kategori *medium* yaitu missing *security* header, 0 kategori *low* dan 4 kategori info. Dari hasil temuan tersebut didapatkan *security score Website* ini adalah 80 yang menunjukkan bahwa *Website* ini sudah cukup aman namun perlu dilakukan perbaikan untuk memaksimalkan *Website* tersebut.

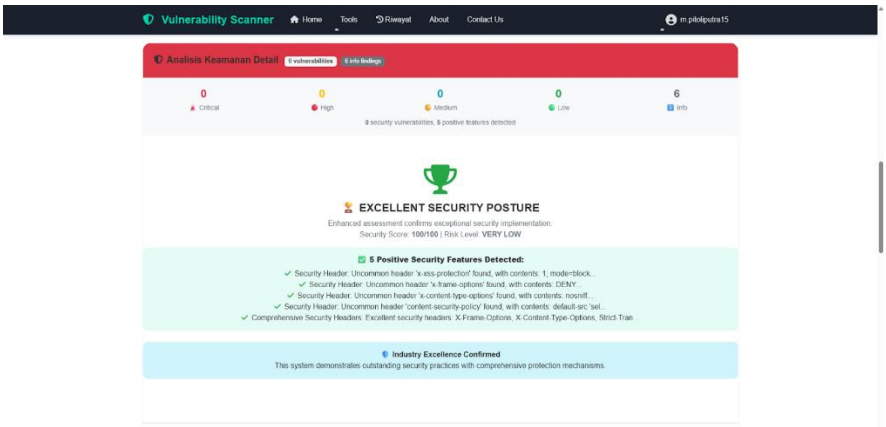


Gambar 5.15 Pengujian ketiga *Website* beda *device* (Localhost:5002) dengan Nikto *Scanner*

c. *Website* 3 (localhost:5003)

- Pengujian pertama :

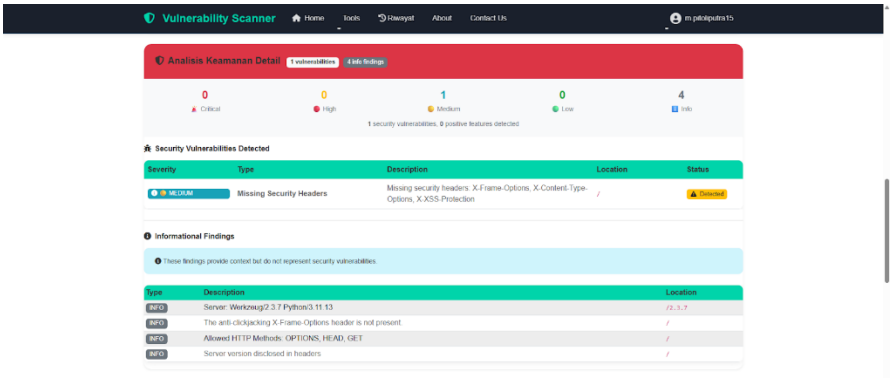
Gambar 5.16 merupakan hasil pengujian ketiga pada *Website* localhost:5003 dengan durasi scan selama 4.15 detik, menjukan hasil temuan kerentanan celah keamanan berupa 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 6 kategori info. Dari hasil yang didapatkan maka *security score Website* localhost:5003 adalah 100 yang menunjukkan *Website* ini aman dan tingkat risiko serangan sangat rendah.



Gambar 5.16 Pengujian pertama *Website* beda *device* (Localhost:5003) dengan Nikto *Scanner*

- Pengujian kedua :

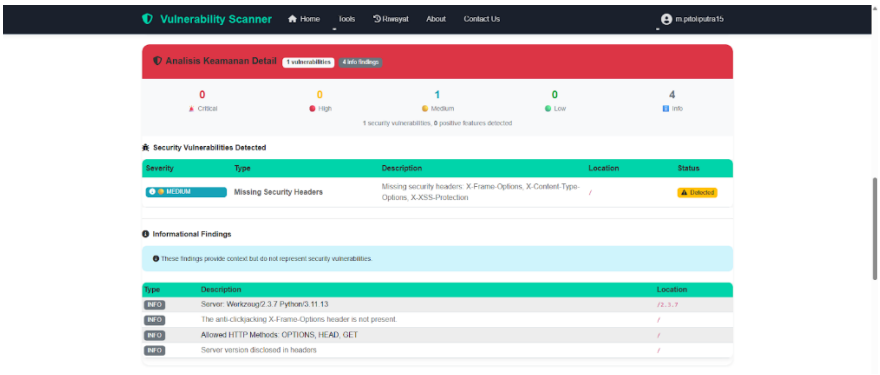
Gambar 5.17 merupakan hasil pengujian ketiga pada *Website* localhost:5003 dengan durasi scan selama 5.35 detik, menjukan hasil temuan kerentanan celah keamanan berupa 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 6 kategori info. Dari hasil yang didapatkan maka *security score Website* localhost:5003 adalah 100 yang menunjukkan *Website* ini aman dan tingkat risiko serangan sangat rendah.



Gambar 5.17 Pengujian kedua *Website* beda *device* (Localhost:5003) dengan Nikto *Scanner*

- Pengujian ketiga :

Gambar 5.18 merupakan hasil pengujian ketiga pada *Website* localhost:5003 dengan durasi scan selama 4.03 detik, menjukan hasil temuan kerentanan celah keamanan berupa 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 6 kategori info. Dari hasil yang didapatkan maka *security score Website* localhost:5003 adalah 100 yang menunjukkan *Website* ini aman dan tingkat risiko serangan sangat rendah.



Gambar 5.18 Pengujian ketiga *Website* beda *device*

Tabel 5.2 Pengujian *Website* Localhost Beda *Device* Menggunakan Nikto *Scanner*

No.	Waktu Pengujian			Konsistensi Hasil Temuan			Keterangan
	P1	P 2	P 3	P 1	P 2	P 3	
Website 1	10.51 detik	9.71 detik	10.14 detik	Skor kerentanan 0	Skor kerentanan 0	Skor kerentanan 0	Hasil konsisten
Website 2	8.23 detik	6.12 detik	11.82 detik	Skor kerentanan 80	Skor kerentanan 80	Skor kerentanan 80	Hasil konsisten
Website 3	4.15 detik	5.35 detik	4.03 detik	Skor kerentanan 100	Skor kerentanan 100	Skor kerentanan 100	Hasil konsisten

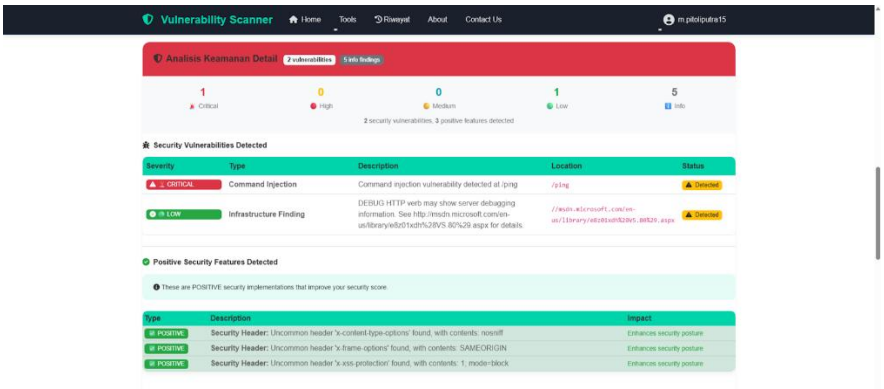
5.2.1.3 Pengujian *Website* dari Internet

Pengujian *Website* dari internet menggunakan 3 buah *Website* sebagai berikut :

a. *Website* 1 (https://www.pln.co.id)

- Pengujian pertama :

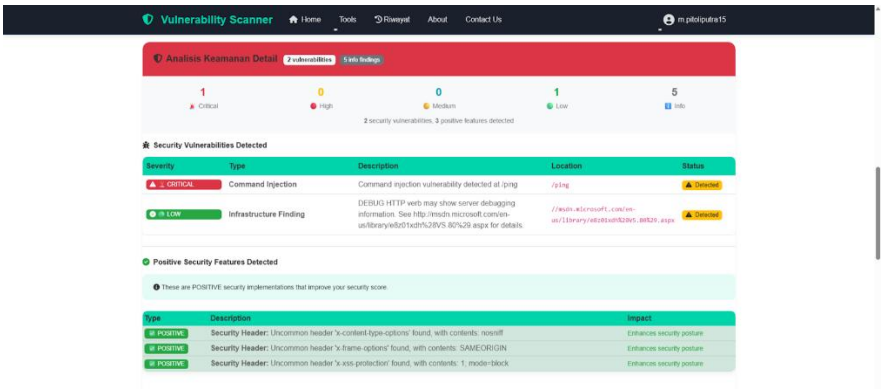
Gambar 5.19 merupakan hasil pengujian pertama pada *Website* https://www.pln.co.id dengan durasi scan 20.53 detik, menunjukkan hasil temuan kerentanan berupa 1 kategori *critical* yaitu command injection, 0 kategori *high*, 0 kategori *medium*, 1 kategori *low* yaitu infrastructure finding, dan 5 kategori info. Dari hasil temuan kerentanan tersebut didapatkan *security score Website* https://www.pln.co.id adalah 16 yang menujukan bahwa *Website* ini sangat rentan terhadap serangan.



Gambar 5.19 Pengujian pertama *Website* internet (https://www.pln.co.id) dengan Nikto *Scanner*

- Pengujian kedua :

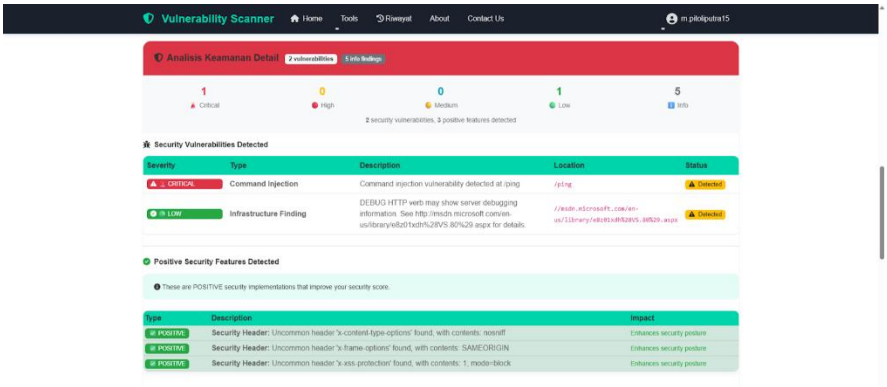
Gambar 5.20 merupakan hasil pengujian kedua pada *Website* https://www.pln.co.id dengan durasi scan 19.78 detik, menunjukkan hasil temuan kerentanan berupa 1 kategori *critical* yaitu command injection, 0 kategori *high*, 0 kategori *medium*, 1 kategori *low* yaitu infrastructure finding, dan 5 kategori info. Dari hasil temuan kerentanan tersebut didapatkan *security score Website* https://www.pln.co.id adalah 16 yang menujukan bahwa *Website* ini sangat rentan terhadap serangan.



Gambar 5.20 Pengujian kedua *Website* internet (https://www.pln.co.id) dengan Nikto *Scanner*

- Pengujian ketiga :

Gambar 5.21 merupakan hasil pengujian ketiga pada *Website* https://www.pln.co.id dengan durasi scan 20.22 detik, menunjukkan hasil temuan kerentanan berupa 1 kategori *critical* yaitu command injection, 0 kategori *high*, 0 kategori *medium*, 1 kategori *low* yaitu infrastructure finding, dan 5 kategori info. Dari hasil temuan kerentanan tersebut didapatkan *security score Website* https://www.pln.co.id adalah 16 yang menujukan bahwa *Website* ini sangat rentan terhadap serangan.

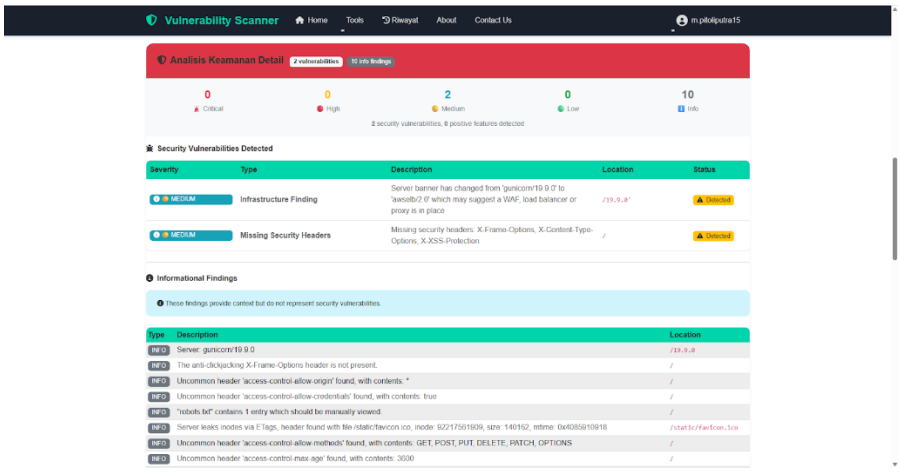


Gambar 5.21 Pengujian ketiga *Website* internet (https://www.pln.co.id) dengan Nikto *Scanner*

b. *Website* 2 (http://httpbin.org)

- Pengujian pertama :

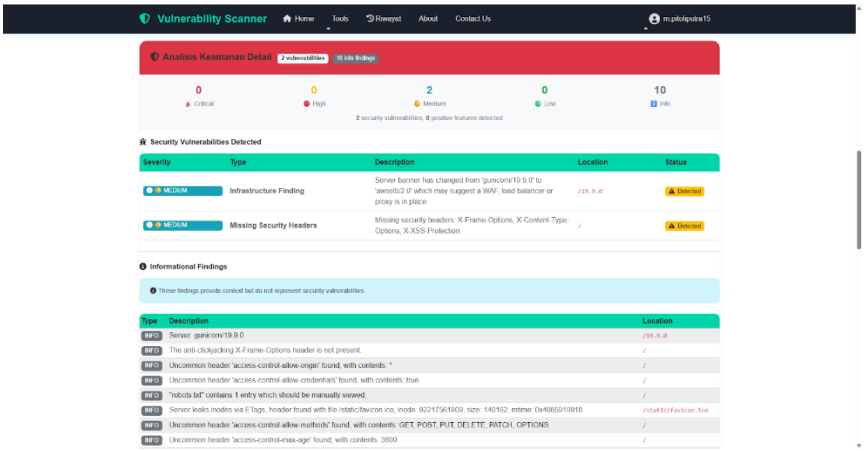
Gambar 5.22 merupakan hasil pengujian pertama pada *Website* http://httpbin.org dengan durasi scan selama 83.10 detik, menunjukkan hasil temuan kerentanan berupa 0 kategori *critical*, 0 kategori *high*, 2 kategori *medium* yaitu infrastructure Finding dan missing *security* headers, 0 kategori *low* dan 10 kategori info. Berdasarkan hasil temuan kerentanan tersebut, maka didapatkan *security score Website* http://httpbin.org adalah 60 yang menunjukkan bahwa *Website* ini cukup aman dari serangan namun perlu dilakukan perbaikan.



Gambar 5.22 Pengujian pertama *Website* internet (http://httpbin.org) dengan Nikto *Scanner*

- Pengujian kedua :

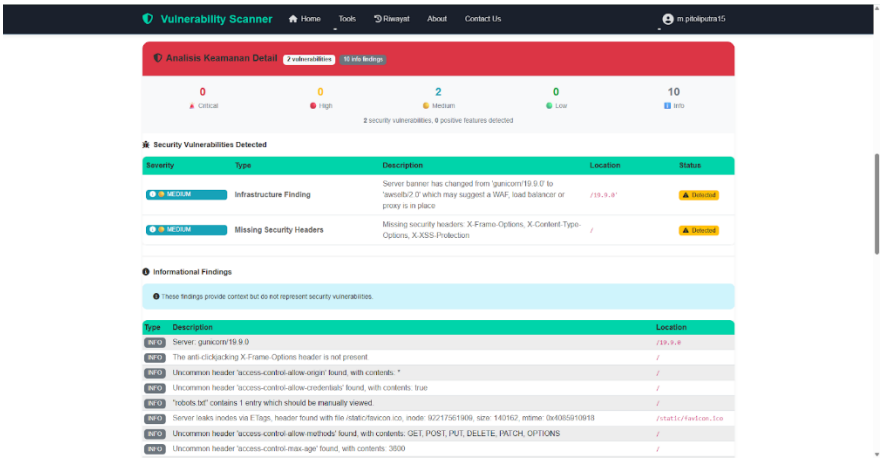
Gambar 5.23 merupakan hasil pengujian kedua pada *Website* http://httpbin.org dengan durasi scan selama 83.97 detik, menunjukkan hasil temuan kerentanan berupa 0 kategori *critical*, 0 kategori *high*, 2 kategori *medium* yaitu *infrastructure Finding* dan *missing security* headers, 0 kategori *low* dan 10 kategori info. Berdasarkan hasil temuan kerentanan tersebut, maka didapatkan *security score Website* http://httpbin.org adalah 60 yang menunjukkan bahwa *Website* ini cukup aman dari serangan namun perlu dilakukan perbaikan.



Gambar 5.23 Pengujian kedua *Website* internet (http://httpbin.org) dengan Nikto *Scanner*

- Pengujian ketiga :

Gambar 5.24 merupakan hasil pengujian ketiga pada *Website* http://httpbin.org dengan durasi scan selama 81.21 detik, menunjukkan hasil temuan kerentanan berupa 0 kategori *critical*, 0 kategori *high*, 2 kategori *medium* yaitu infrastructure Finding dan missing *security* headers, 0 kategori *low* dan 10 kategori info. Berdasarkan hasil temuan kerentanan tersebut, maka didapatkan *security score Website* http://httpbin.org adalah 60 yang menunjukkan bahwa *Website* ini cukup aman dari serangan namun perlu dilakukan perbaikan.

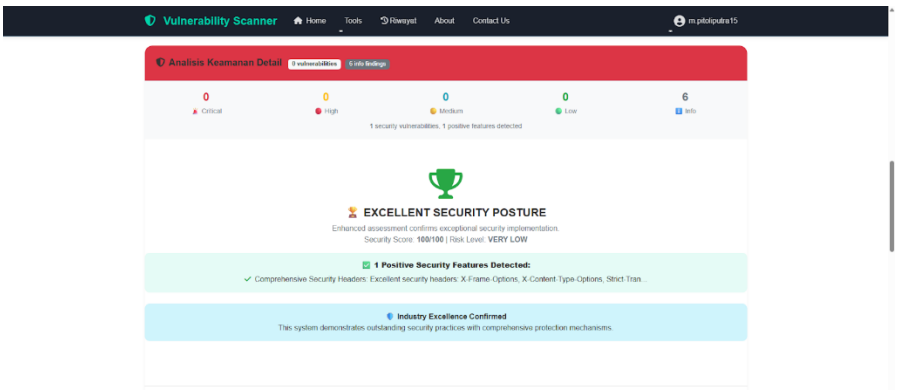


Gambar 5.24 Pengujian ketiga *Website* internet (http://httpbin.org) dengan Nikto *Scanner*

c. *Website* 3 (https://www.komdigi.go.id/)

- Pengujian pertama :

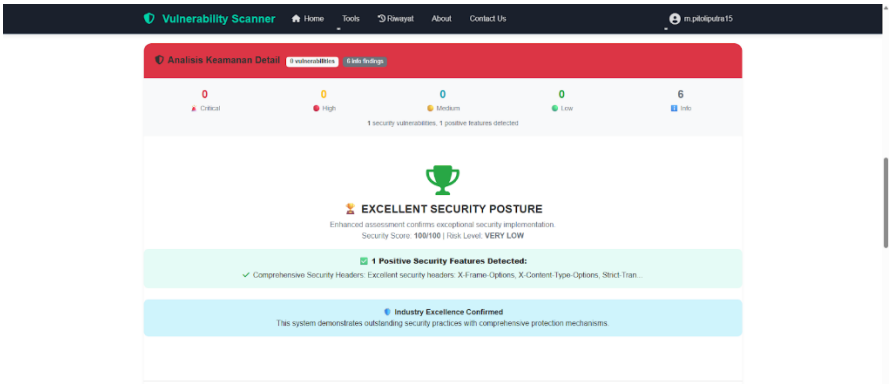
Gambar 5.25 merupakan hasil pengujian pertama pada *Website* https://www.komdigi.go.id/ dengan durasi scan selama 22.52 detik, menunjukkan hasil temuan kerentanan berupa 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 6 kategori info. Berdasarkan hasil temuan kerentanan tersebut, maka didapatkan *security score Website* https://www.komdigi.go.id/ adalah 100 yang menunjukkan bahwa *Website* sangat aman dan sangat minim terhadap risiko serangan.



Gambar 5.25 Pengujian pertama *Website* internet (https://www.komdigi.go.id/) dengan Nikto *Scanner*

- Pengujian kedua :

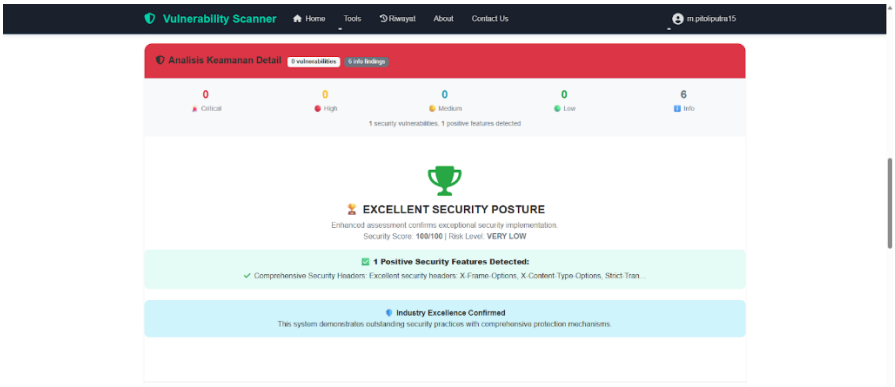
Gambar 5.26 merupakan hasil pengujian kedua pada *Website* https://www.komdigi.go.id/ dengan durasi scan selama 24.12 detik, menunjukkan hasil temuan kerentanan berupa 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 6 kategori info. Berdasarkan hasil temuan kerentanan tersebut, maka didapatkan *security score Website* https://www.komdigi.go.id/ adalah 100 yang menunjukkan bahwa *Website* sangat aman dan sangat minim terhadap risiko serangan.



Gambar 5.26 Pengujian kedua *Website* internet (https://www.komdigi.go.id/) dengan Nikto *Scanner*

- Pengujian ketiga :

Gambar 5.27 merupakan hasil pengujian ketiga pada *Website* https://www.komdigi.go.id/ dengan durasi scan selama 23.88 detik, menunjukkan hasil temuan kerentanan berupa 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 6 kategori info. Berdasarkan hasil temuan kerentanan tersebut, maka didapatkan *security score Website* https://www.komdigi.go.id/ adalah 100 yang menunjukkan bahwa *Website* sangat aman dan sangat minim terhadap risiko serangan.



Gambar 5.27 Pengujian ketiga*Website* internet (https://www.komdigi.go.id/) dengan Nikto *Scanner*

Tabel 5. 3 Pengujian *Website* Internet Menggunakan Nikto *Scanner*

No.	Waktu Pengujian			Konsistensi Hasil Temuan			Keterangan
	P1	P 2	P 3	P 1	P 2	P 3	
Website 1	20.53 detik	19.78 detik	20.22 detik	Skor kerentanan 16	Skor kerentanan 16	Skor kerentanan 16	Hasil konsisten
Website 2	83.10 detik	83.97 detik	81.21 detik	Skor kerentanan 60	Skor kerentanan 60	Skor kerentanan 60	Hasil konsisten
Website 3	22.52 detik	24.12 detik	23.88 detik	Skor kerentanan 100	Skor kerentanan 100	Skor kerentanan 100	Hasil konsisten

5.2.2 Pengujian Menggunakan Nmap *Scanner*

Pengujian menggunakan Nmap *scanner* ini mencakup 3 metode dengan menggunakan 3 *Website* per masing – masing metodenya.

5.2.2.1 Pengujian *Website* Localhost 1 *Device*

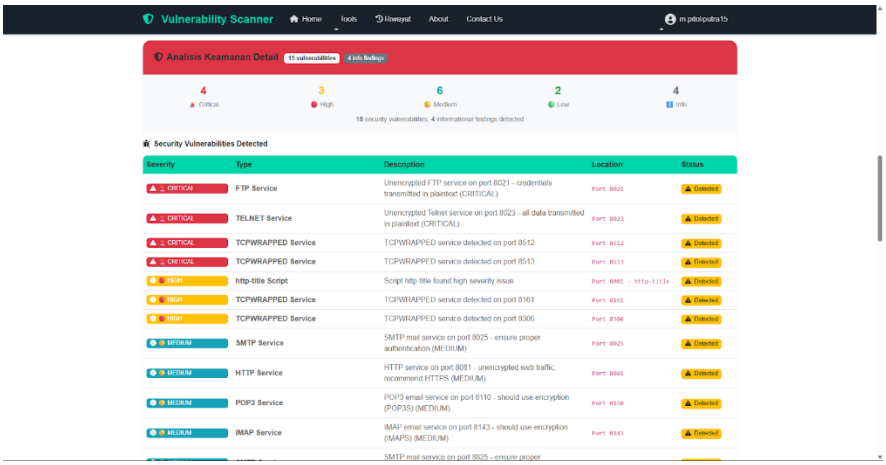
Pada pengujian *Website* localhost 1 *device* menggunakan 3 buah *Website* sebagai berikut :

a. *Website* 1 (localhost:8081)

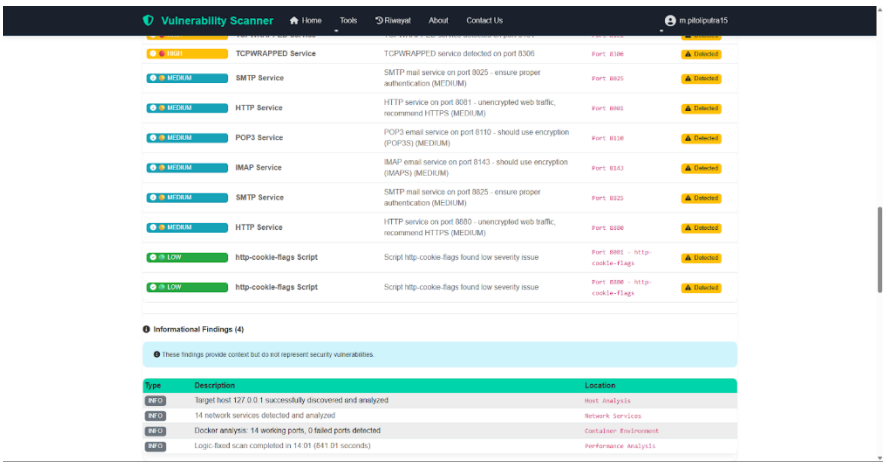
- Pengujian pertama :

  Gambar 5.28 dan Gambar 5.29 merupakan hasil pengujian pertama pada *Website* target localhost:8081 dengan durasi scan selama 840.91 detik, didapatkan hasil temuan kerentanan dengan 4 kategori *critical*

diantaranya FTP *service*, TELNET *service*, dan TCPWRAPPED *service*. Terdapat 3 kategori *high* yaitu http-title script dan TCPWRAPPED *service*s. Terdapat 6 kategori *medium* yaitu SMTP *service*, HTTP *Service*, dan POP3 *Service* IMAP *service*. Terdapat 2 kategori *low* yaitu http-cookiep-flags script dan 4 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8081 mendapatkan *security score* sebesar 0 yang menunjukkan bahwa *Website* ini sangat rentan terhadap serangan.



Gambar 5.28 Pengujian pertama *Website* 1 *device* (Localhost:8081) dengan nmap *Scanner*

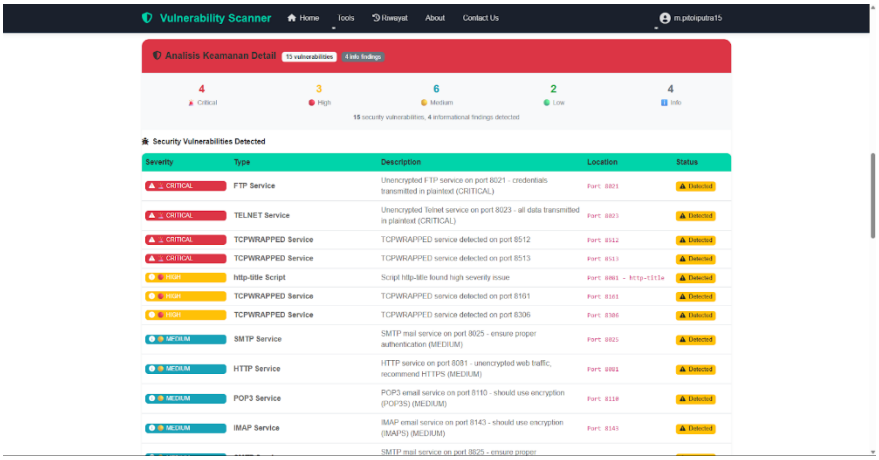


Gambar 5.29 Pengujian pertama *Website* 1 *device* (Localhost:8081) dengan nmap *Scanner*

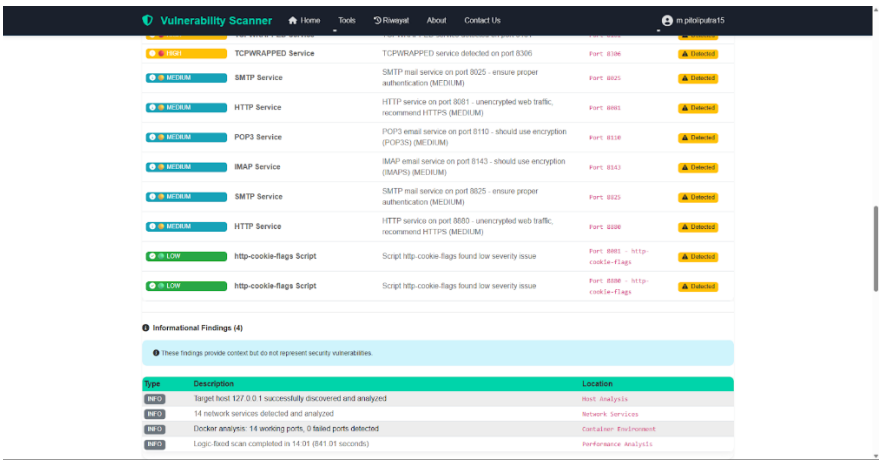
- Pengujian kedua :

Gambar 5.30 dan Gambar 5.31 merupakan hasil pengujian kedua pada *Website* target localhost:8081 dengan durasi scan selama 851.11

detik, didapatkan hasil temuan kerentanan dengan 4 kategori *critical* diantaranya FTP *service*, TELNET *service*, dan TCPWRAPPED *service*. Terdapat 3 kategori *high* yaitu http-title script dan TCPWRAPPED *service*s. Terdapat 6 kategori *medium* yaitu SMTP *service*, HTTP *Service*, dan POP3 *Service* IMAP *service*. Terdapat 2 kategori *low* yaitu http-cookiep-flags script dan 4 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8081 mendapatkan *security score* sebesar 0 yang menunjukkan bahwa *Website* ini sangat rentan terhadap serangan.



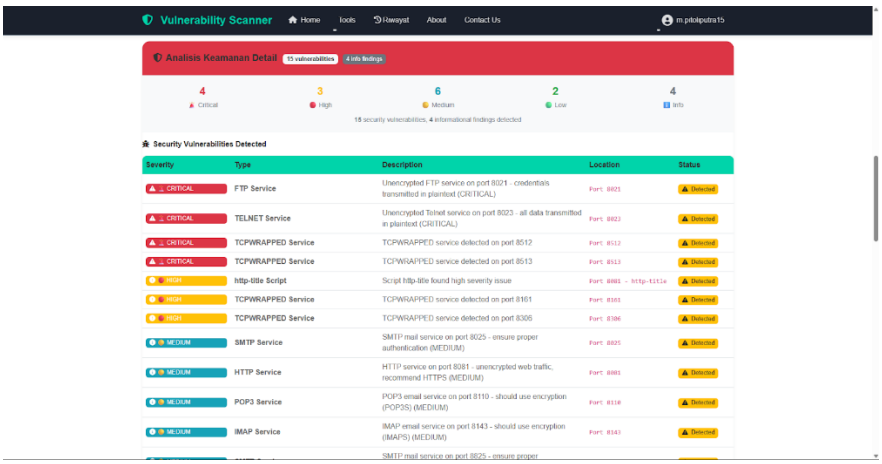
Gambar 5.30 Pengujian kedua *Website* 1 *device* (Localhost:8081) dengan nmap *Scanner*



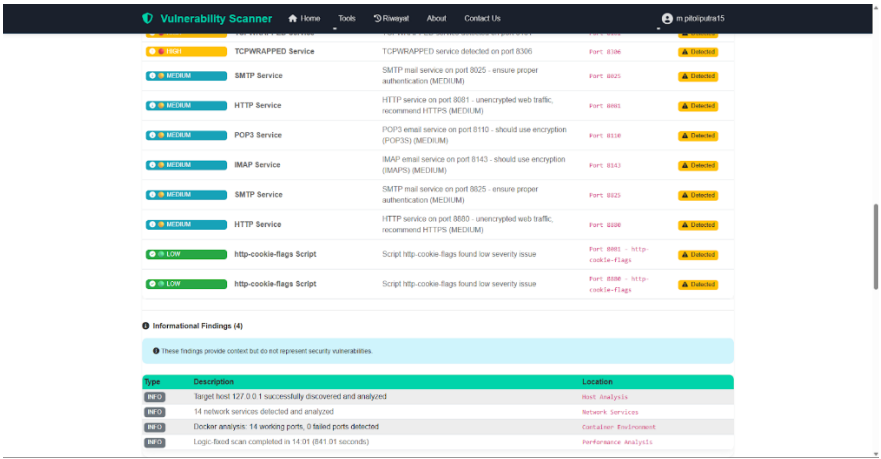
Gambar 5.31 Pengujian kedua *Website* 1 *device* (Localhost:8081) dengan nmap *Scanner*

- Pengujian ketiga :

Gambar 5.32 dan Gambar 5.33 merupakan hasil pengujian ketiga pada *Website* target localhost:8081 dengan durasi scan selama 849.41 detik, didapatkan hasil temuan kerentanan dengan 4 kategori *critical* diantaranya FTP *service*, TELNET *service*, dan TCPWRAPPED *service*. Terdapat 3 kategori *high* yaitu http-title script dan TCPWRAPPED *service*s. Terdapat 6 kategori *medium* yaitu SMTP *service*, HTTP *Service*, dan POP3 *Service* IMAP *service*. Terdapat 2 kategori *low* yaitu http-cookiep-flags script dan 4 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8081 mendapatkan *security score* sebesar 0 yang menunjukkan bahwa *Website* ini sangat rentan terhadap serangan.



Gambar 5.32 Pengujian ketiga *Website* 1 *device* (Localhost:8081) dengan nmap *Scanner*

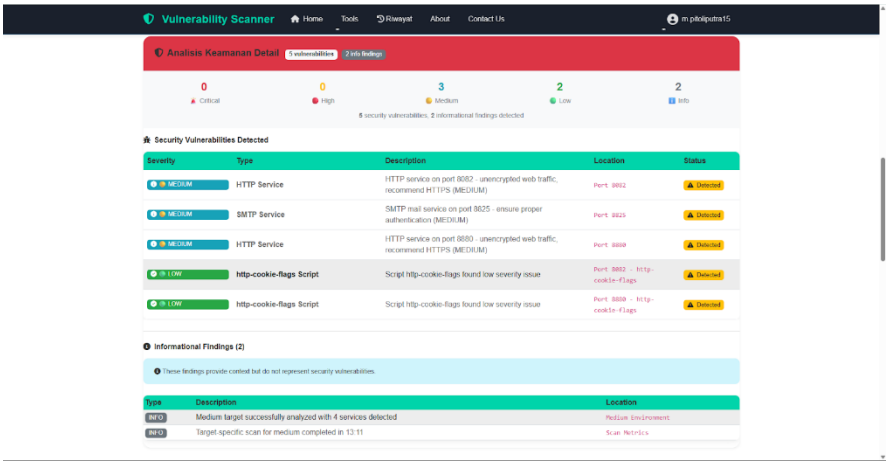


Gambar 5.33 Pengujian ketiga *Website* 1 *device* (Localhost:8081) dengan nmap *Scanner*

b. *Website* 2 (localhost:8082)

- Pengujian pertama :

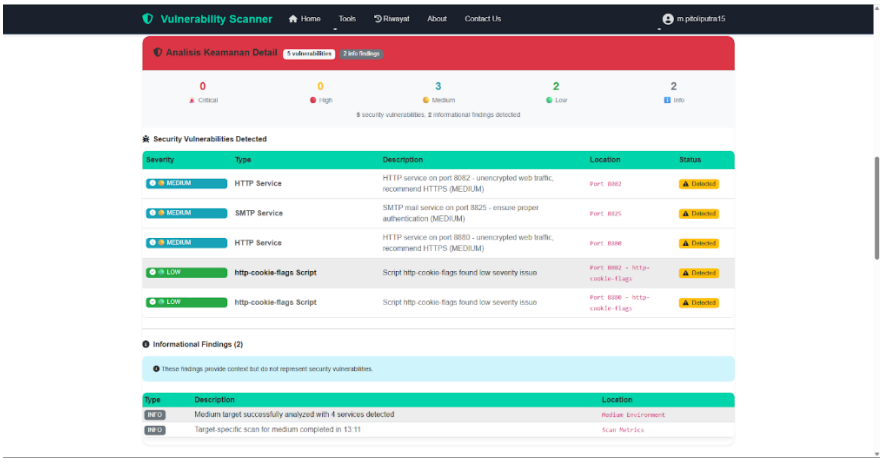
Gambar 5.34 merupakan hasil pengujian pertama pada *Website* target localhost:8082 dengan durasi scan selama 791.64 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 2 kategori *low* yaitu http-cookiep-flags script dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8082 mendapatkan *security score* sebesar 57 yang menunjukkan bahwa *Website* ini cukup aman namun perlu dilakukan perbaikan.



Gambar 5.34 Pengujian pertama *Website* 1 *device* (Localhost:8082) dengan nmap *Scanner*

- Pengujian kedua :

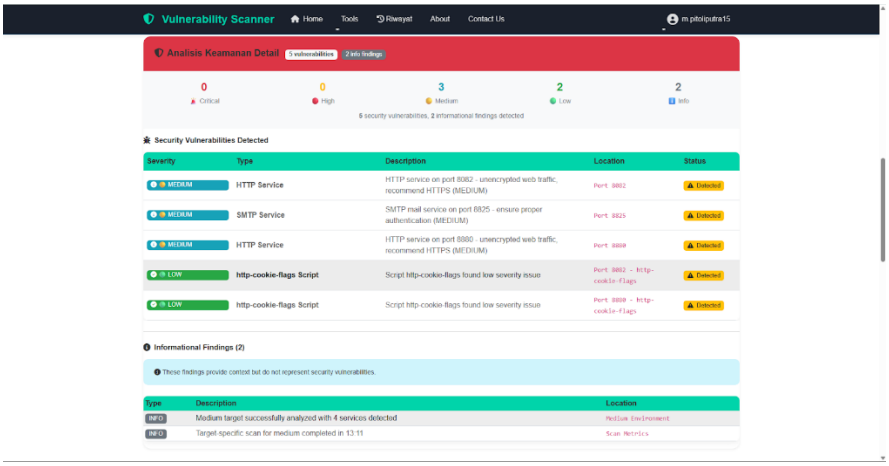
Gambar 5.35 merupakan hasil pengujian kedua pada *Website* target localhost:8082 dengan durasi scan selama 801.13 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 2 kategori *low* yaitu http-cookiep-flags script dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8082 mendapatkan *security score* sebesar 57 yang menunjukkan bahwa *Website* ini cukup aman namun perlu dilakukan perbaikan.



Gambar 5. 35 Pengujian kedua *Website* 1 *device* (Localhost:8082) dengan nmap *Scanner*

- Pengujian ketiga :

Gambar 5.36 merupakan hasil pengujian ketiga pada *Website* target localhost:8082 dengan durasi scan selama 782.34 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 2 kategori *low* yaitu http-cookiep-flags script dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8082 mendapatkan *security score* sebesar 57 yang menunjukkan bahwa *Website* ini cukup aman namun perlu dilakukan perbaikan.

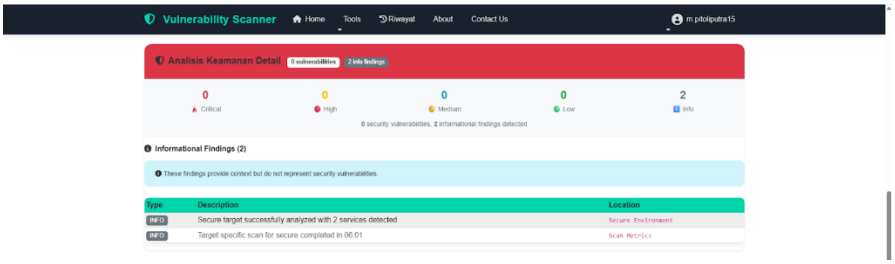


Gambar 5.36 Pengujian ketiga *Website* 1 *device* (Localhost:8082) dengan nmap *Scanner*

c. *Website* 3 (localhost:8083)

- Pengujian pertama :

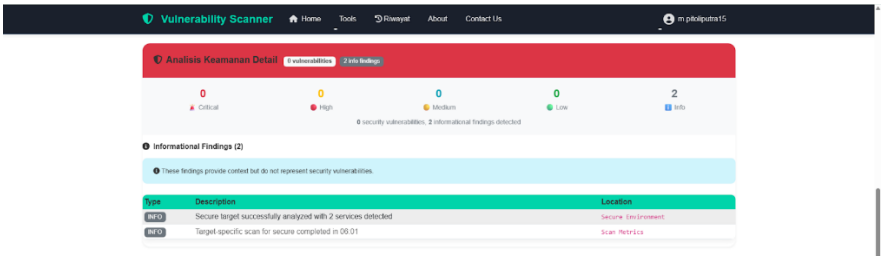
Gambar 5.37 merupakan hasil pengujian pertama pada *Website* target localhost:8083 dengan durasi scan selama 361.34 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8083 mendapatkan *security score* sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendapat serangan sangat rendah.



Gambar 5.37 Pengujian pertama *Website* 1 *device* (Localhost:8083) dengan nmap *Scanner*

- Pengujian kedua :

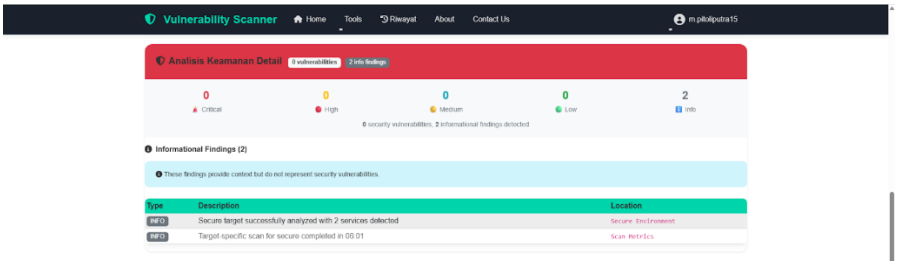
Gambar 5.38 merupakan hasil pengujian kedua pada *Website* target localhost:8083 dengan durasi scan selama 354.09 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8083 mendapatkan *security score* sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendapat serangan sangat rendah.



Gambar 5.38 Pengujian kedua *Website* 1 *device* (Localhost:8083) dengan nmap *Scanner*

- Pengujian ketiga :

Gambar 5.39 merupakan hasil pengujian ketigaa pada *Website* target localhost:8083 dengan durasi scan selama 359.03 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8083 mendapatkan *security score* sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendapat serangan sangat rendah.



Gambar 5.39 Pengujian ketiga *Website* 1 *device* (Localhost:8083) dengan nmap *Scanner*

Tabel 5.4 Tabel Pengujian *Website* 1 *Device* Menggunakan Nmap *Scanner*

No.	Waktu Pengujian			Konsistensi Hasil Temuan			Keterangan
	P1	P 2	P 3	P 1	P 2	P 3	
Website 1	840.91 detik	851.11 detik	849.41 detik	Skor kerentanan 0	Skor kerentanan 0	Skor kerentanan 0	Hasil konsisten
Website 2	791.64 detik	801.13 detik	782.34 detik	Skor kerentanan 57	Skor kerentanan 57	Skor kerentanan 57	Hasil konsisten
Website 3	361.43 detik	354.09 detik	359.03 detik	Skor kerentanan 100	Skor kerentanan 100	Skor kerentanan 100	Hasil konsisten

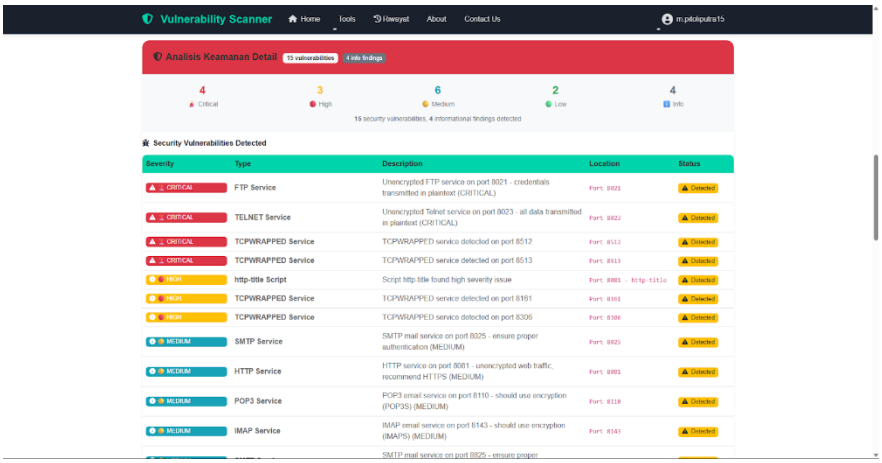
5.2.2.2 Pengujian *Website* Localhost dari *Device* yang Berbeda

Pada pengujian *Website* localhost dari *device* yang berbeda menggunakan 3 buah *Website* sebagai berikut :

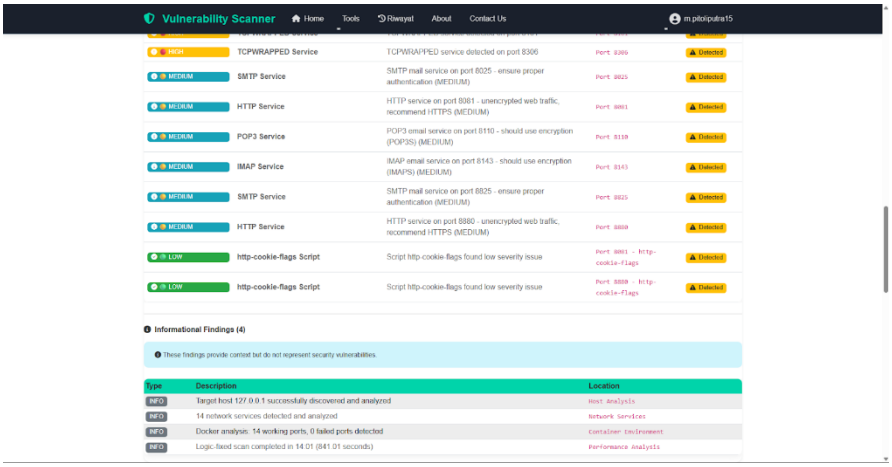
a. *Website* 1 (localhost:8081)

- Pengujian pertama :

  Gambar 5.40 dan Gambar 5.41 merupakan hasil pengujian pertama pada *Website* target localhost:8081 dari *device* lain dengan durasi scan selama 982.12 detik, didapatkan hasil temuan kerentanan dengan 4 kategori *critical* diantaranya FTP *service*, TELNET *service*, dan TCPWRAPPED *service*. Terdapat 3 kategori *high* yaitu http-title script dan TCPWRAPPED *service*s. Terdapat 6 kategori *medium* yaitu SMTP *service*, HTTP *Service*, dan POP3 *Service* IMAP *service*. Terdapat 2 kategori *low* yaitu http-cookiep-flags script dan 4 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8081 mendapatkan *security score* sebesar 0 yang menunjukkan bahwa *Website* ini sangat rentan terhadap serangan.



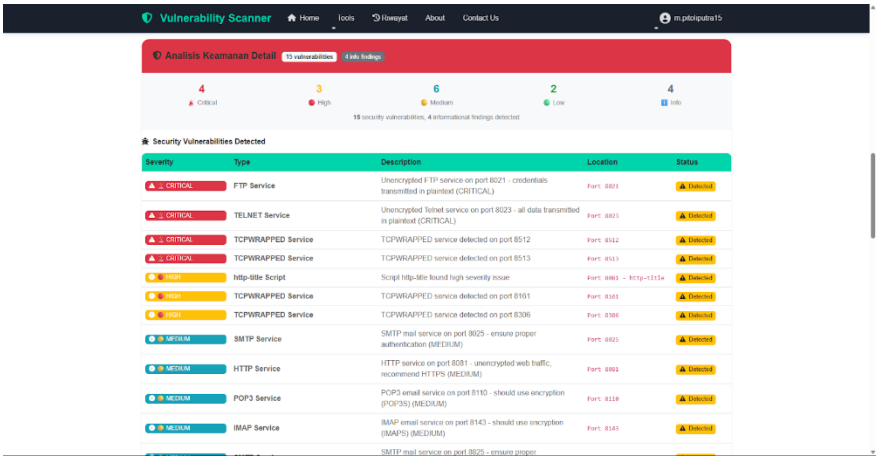
Gambar 5.40 Pengujian pertama *Website* beda *device* (Localhost:8081) dengan nmap *Scanner*



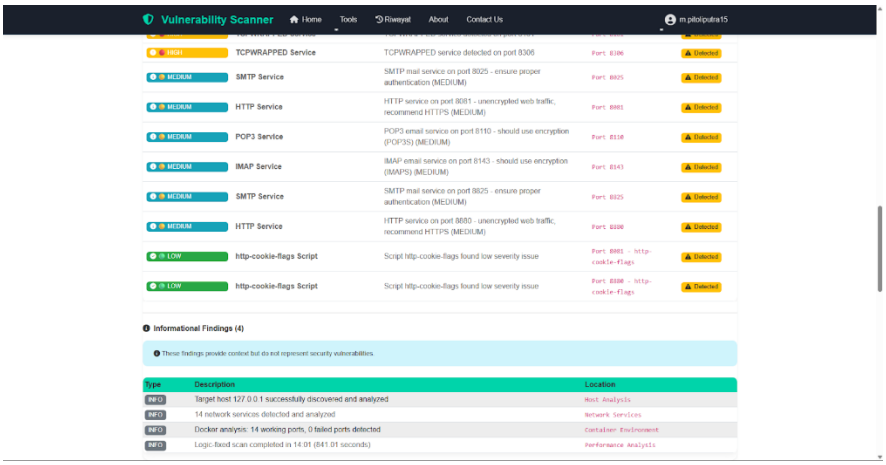
Gambar 5.41 Pengujian pertama *Website* beda *device* (Localhost:8081) dengan nmap *Scanner*

- Pengujian kedua :

Gambar 5.42 dan Gambar 5.43 merupakan hasil pengujian kedua pada *Website* target localhost:8081 dari *device* lain dengan durasi scan selama 993.42 detik, didapatkan hasil temuan kerentanan dengan 4 kategori *critical* diantaranya FTP *service*, TELNET *service*, dan TCPWRAPPED *service*. Terdapat 3 kategori *high* yaitu http-title script dan TCPWRAPPED *service*s. Terdapat 6 kategori *medium* yaitu SMTP *service*, HTTP *Service*, dan POP3 *Service* IMAP *service*. Terdapat 2 kategori *low* yaitu http-cookiep-flags script dan 4 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8081 mendapatkan *security score* sebesar 0 yang menunjukkan bahwa *Website* ini sangat rentan terhadap serangan.



Gambar 5.42 Pengujian kedua *Website* beda *device* (Localhost:8081) dengan nmap *Scanner*

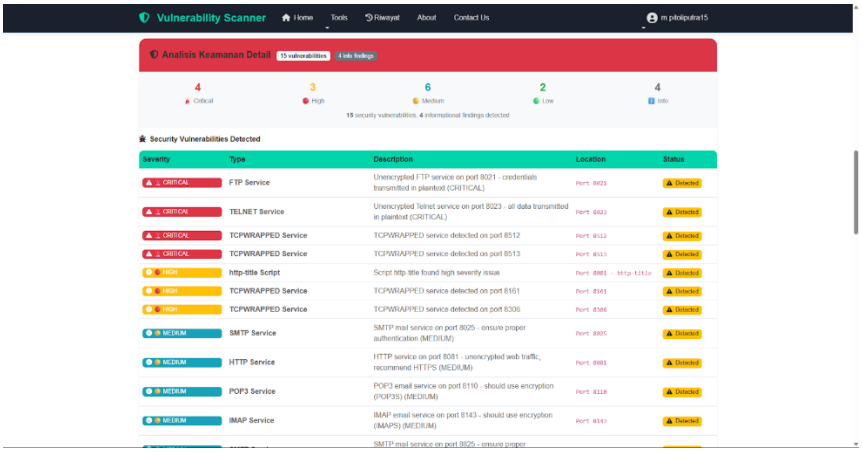


Gambar 5.43 Pengujian kedua *Website* beda *device* (Localhost:8081) dengan nmap *Scanner*

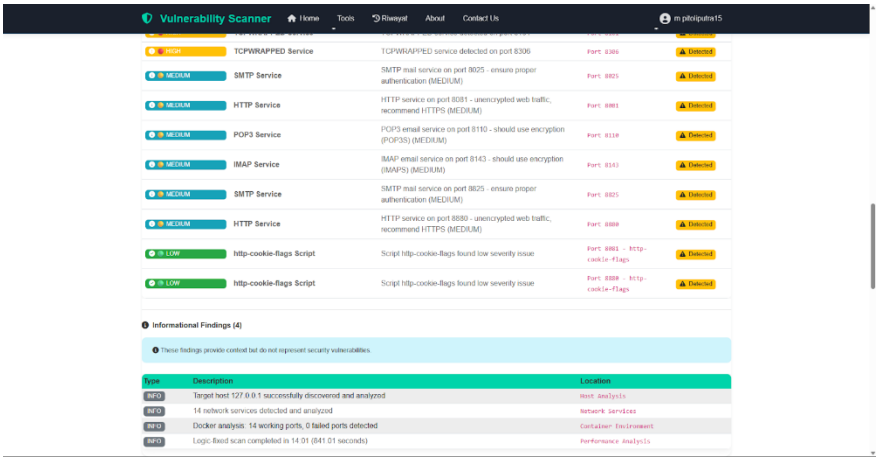
- Pengujian ketiga :

Gambar 5.44 dan Gambar 5.45 merupakan hasil pengujian ketiga pada *Website* target localhost:8081 dari *device* lain dengan durasi scan selama 992.22 detik, didapatkan hasil temuan kerentanan dengan 4 kategori *critical* diantaranya FTP *service*, TELNET *service*, dan TCPWRAPPED *service*. Terdapat 3 kategori *high* yaitu http-title script dan TCPWRAPPED *service*s. Terdapat 6 kategori *medium* yaitu SMTP *service*, HTTP *Service*, dan POP3 *Service* IMAP *service*. Terdapat 2 kategori *low* yaitu http-cookiep-flags script dan 4 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8081 mendapatkan

security score sebesar 0 yang menunjukkan bahwa *Website* ini sangat rentan terhadap serangan.



Gambar 5.44 Pengujian ketiga *Website* beda *device* (Localhost:8081) dengan nmap *Scanner*



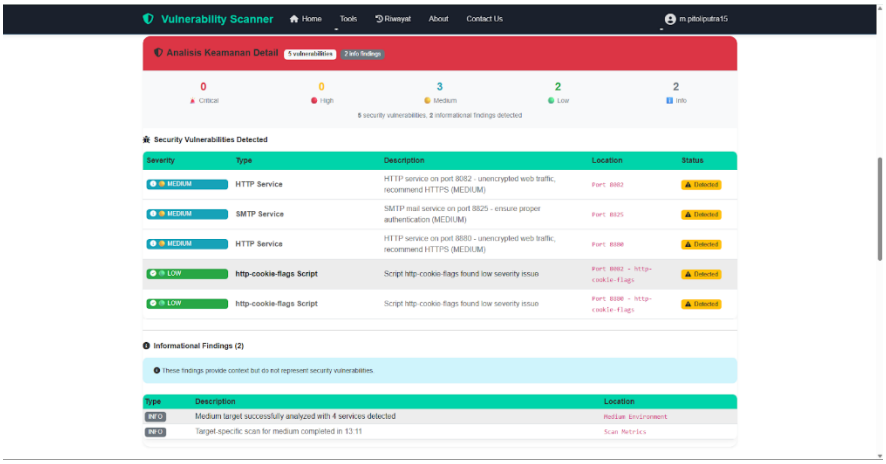
Gambar 5.45 Pengujian ketiga *Website* beda *device* (Localhost:8081) dengan nmap *Scanner*

b. *Website* 2 (localhost:8082)

- Pengujian pertama :

Gambar 5.46 merupakan hasil pengujian pertama pada *Website* target localhost:8082 dengan durasi scan selama 850.21 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 2 kategori *low* yaitu http-cookiep-flags script dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8082 mendapatkan *security score* sebesar 57

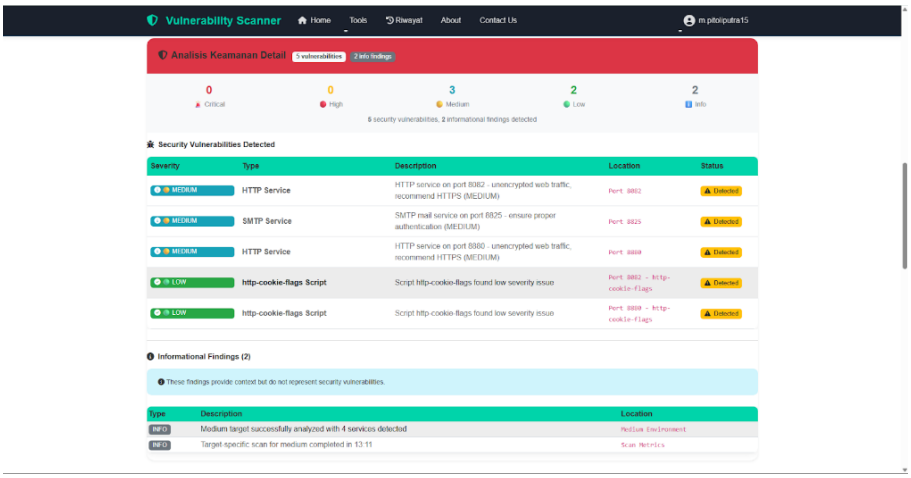
yang menunjukkan bahwa *Website* ini cukup aman namun perlu dilakukan perbaikan.



Gambar 5.46 Pengujian pertama *Website* beda *device* (Localhost:8082) dengan nmap *Scanner*

- Pengujian kedua :

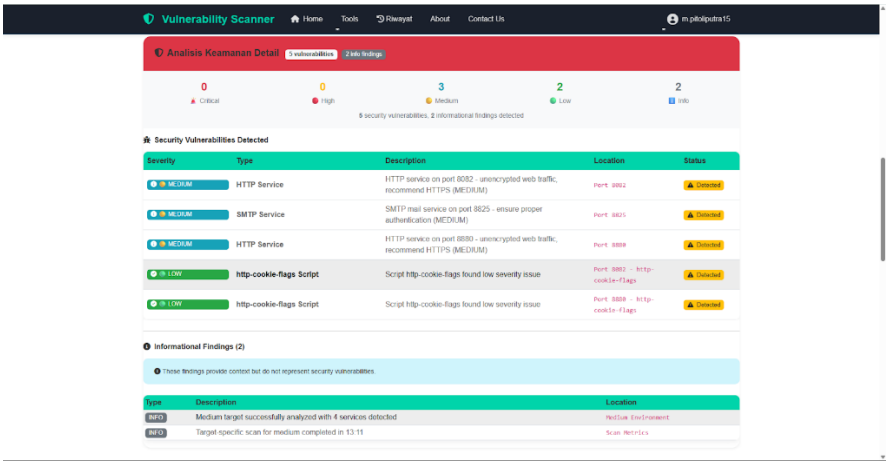
Gambar 5.47 merupakan hasil pengujian kedua pada *Website* target localhost:8082 dari *device* lain dengan durasi scan selama 862.12 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 2 kategori *low* yaitu http-cookiep-flags script dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8082 mendapatkan *security score* sebesar 57 yang menunjukkan bahwa *Website* ini cukup aman namun perlu dilakukan perbaikan.



Gambar 5.47 Pengujian kedua *Website* beda *device* (Localhost:8082) dengan nmap *Scanner*

- Pengujian ketiga :

Gambar 5.48 merupakan hasil pengujian ketiga pada *Website* target localhost:8082 dari *device* lain dengan durasi scan selama 859.33 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 2 kategori *low* yaitu http-cookiep-flags script dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8082 mendapatkan *security score* sebesar 57 yang menunjukkan bahwa *Website* ini cukup aman namun perlu dilakukan perbaikan.

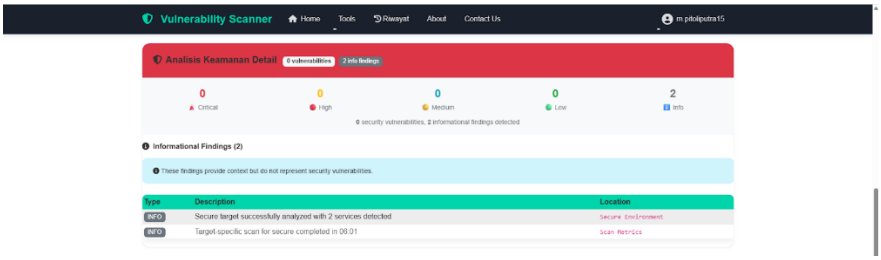


Gambar 5.48 Pengujian ketiga *Website* beda *device* (Localhost:8082) dengan nmap *Scanner*

c. *Website* 3 (localhost:8083)

- Pengujian pertama :

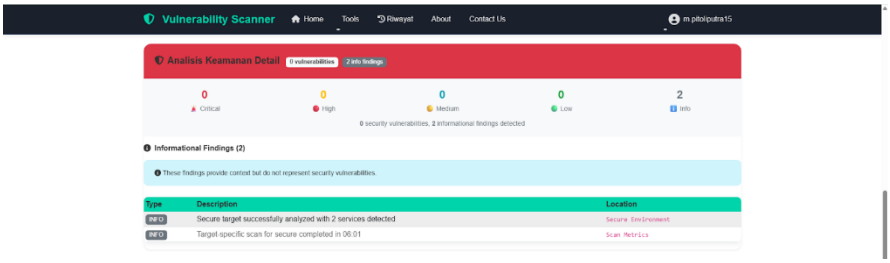
Gambar 5.49 merupakan hasil pengujian pertama pada *Website* target localhost:8083 dengan durasi scan selama 442.76 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8083 mendapatkan *security score* sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendapat serangan sangat rendah.



Gambar 5.49 Pengujian pertama *Website* beda *device* (Localhost:8083) dengan nmap *Scanner*

- Pengujian kedua :

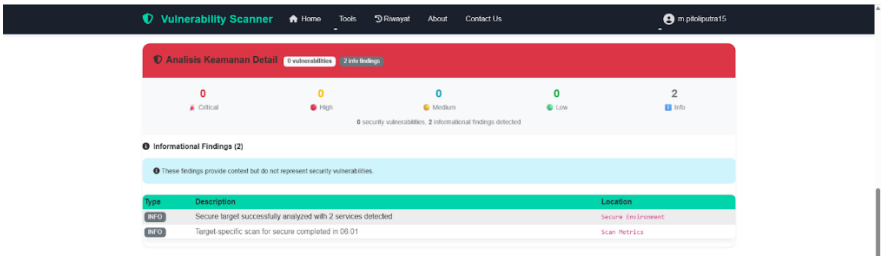
Gambar 5.50 merupakan hasil pengujian kedua pada *Website* target localhost:8083 dengan durasi scan selama 437.21 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8083 mendapatkan *security score* sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendapat serangan sangat rendah.



Gambar 5.50 Pengujian kedua *Website* beda *device* (Localhost:8083) dengan nmap *Scanner*

- Pengujian ketiga :

Gambar 5.51 merupakan hasil pengujian ketigaa pada *Website* target localhost:8083 dengan durasi scan selama 459.45 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 3 kategori *medium* yaitu SMTP *service*, HTTP *Service*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* localhost:8083 mendapatkan *security score* sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendapat serangan sangat rendah.



Gambar 5.51 Pengujian ketiga *Website* beda *device* (Localhost:8083) dengan nmap *Scanner*

Tabel 5.5 Pengujian *Website* Beda *Device* Menggunakan Nmap *Scanner*

No.	Waktu Pengujian			Konsistensi Hasil Temuan			Keterangan
	P1	P 2	P 3	P 1	P 2	P 3	
Website 1	982.12 detik	993.42 detik	992.22 detik	Skor kerentanan 0	Skor kerentanan 0	Skor kerentanan 0	Hasil konsisten
Website 2	850.12 detik	862.12 detik	859.33 detik	Skor kerentanan 57	Skor kerentanan 57	Skor kerentanan 57	Hasil konsisten
Website 3	422.76 detik	437.21 detik	459.45 detik	Skor kerentanan 100	Skor kerentanan 100	Skor kerentanan 100	Hasil konsisten

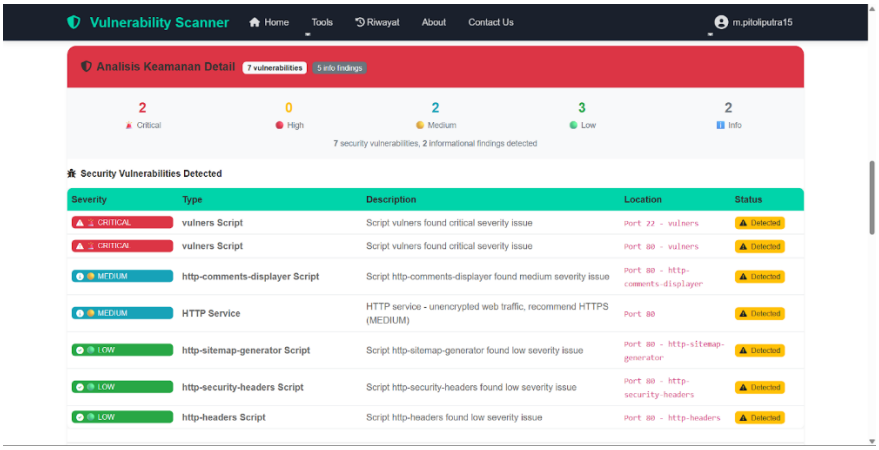
5.2.2.3 Pengujian *Website* dari Internet

Pada pengujian *Website* dari internet menggunakan 3 buah *Website* sebagai berikut :

a. *Website* 1 (scanme.nmap.org)

- Pengujian pertama :

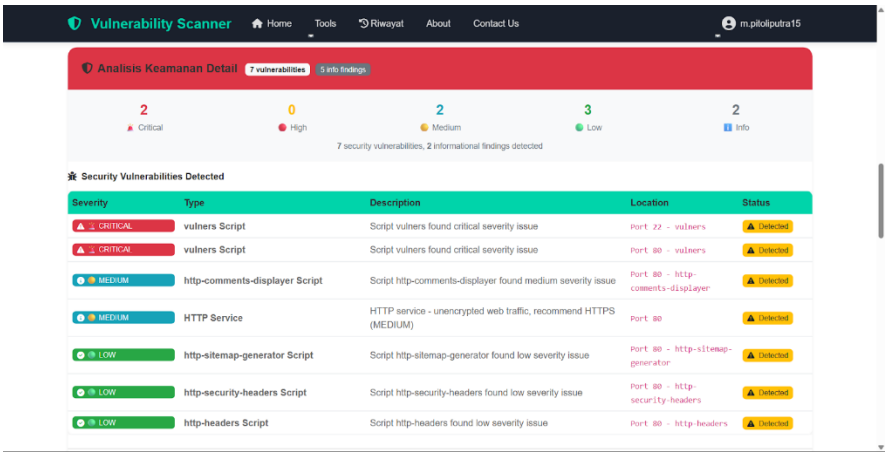
Gambar 5.52 merupakan hasil pengujian pertama pada *Website* target scanme.nmap.org dengan durasi scan selama 176.18 detik, didapatkan hasil temuan kerentanan dengan 2 kategori *critical* yaitu vulners script, 0 kategori *high*, 2 kategori *medium* yaitu HTTP *Service* dan http-comments-displayer script, 3 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* scanme.nmap.org mendapatkan *security score* sebesar 0 yang menunjukkan bahwa *Website* sangat rentan terhadap serangan.



Gambar 5.52 Pengujian pertama *Website* internet (scanme.nmap.org) dengan nmap *Scanner*

- Pengujian kedua :

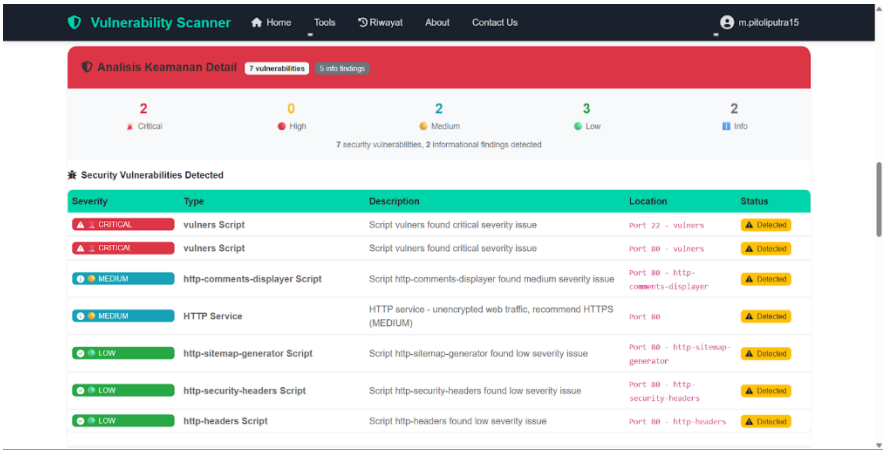
Gambar 5.53 merupakan hasil pengujian kedua pada *Website* target scanme.nmap.org dengan durasi scan selama 180.01 detik, didapatkan hasil temuan kerentanan dengan 2 kategori *critical* yaitu vulners script, 0 kategori *high*, 2 kategori *medium* yaitu HTTP *Service* dan http-comments-displayer script, 3 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* scanme.nmap.org mendapatkan *security score* sebesar 0 yang menunjukkan bahwa *Website* sangat rentan terhadap serangan.



Gambar 5.53 Pengujian kedua *Website* internet (scanme.nmap.org) dengan nmap *Scanner*

- Pengujian ketiga :

Gambar 5.54 merupakan hasil pengujian ketigas pada *Website* target scanme.nmap.org dengan durasi scan selama 166.12 detik, didapatkan hasil temuan kerentanan dengan 2 kategori *critical* yaitu vulners script, 0 kategori *high*, 2 kategori *medium* yaitu HTTP *Service* dan http-comments-displayer script, 3 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* scanme.nmap.org mendapatkan *security score* sebesar 0 yang menunjukkan bahwa *Website* sangat rentan terhadap serangan.

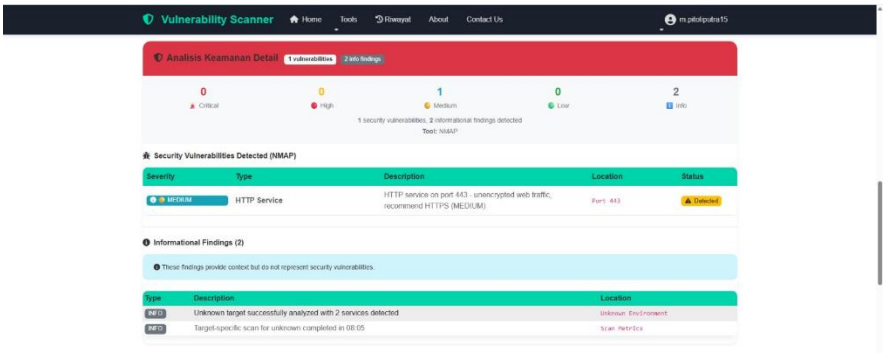


Gambar 5.54 Pengujian ketiga *Website* internet (scanme.nmap.org) dengan nmap *Scanner*

b. *Website* 2 (neverssl.com)

- Pengujian pertama :

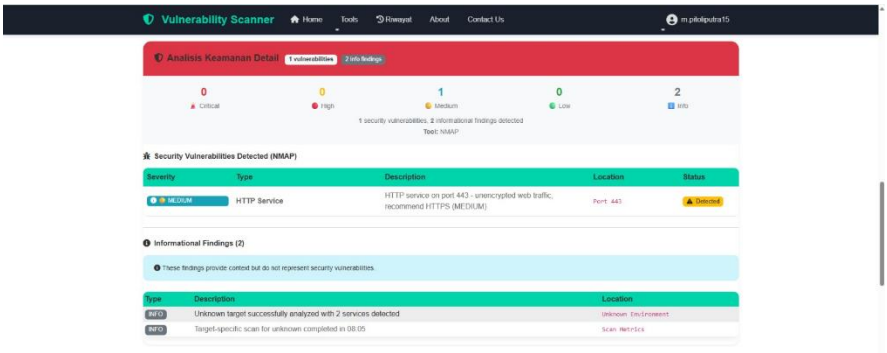
Gambar 5.55 merupakan hasil pengujian pertama pada *Website* target neverssl.com dengan durasi scan selama 481.57 detik, didapatkan hasil temuan kerentanan dengan 1 kategori *medium* yaitu HTTP *Service* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* neverssl.com mendapatkan *security score* sebesar 80 yang menunjukkan bahwa *Website* sangat rentan terhadap serangan.



Gambar 5.55 Pengujian pertama *Website* internet (neverssl.com) dengan nmap *Scanner*

- Pengujian kedua :

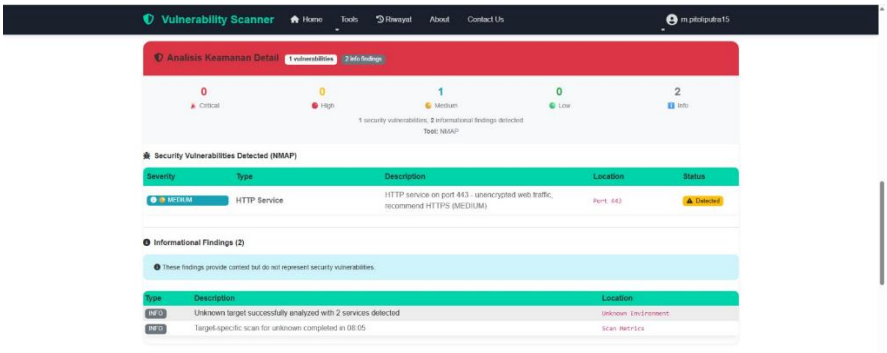
Gambar 5.56 merupakan hasil pengujian pertama pada *Website* target neverssl.com dengan durasi scan selama 485.81 detik, didapatkan hasil temuan kerentanan dengan 1 kategori *medium* yaitu HTTP *Service* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* neverssl.com mendapatkan *security score* sebesar 80 yang menunjukkan bahwa *Website* sangat rentan terhadap serangan.



Gambar 5.56 Pengujian kedua *Website* internet (neverssl.com) dengan nmap *Scanner*

- Pengujian ketiga :

Gambar 5.57 merupakan hasil pengujian pertama pada *Website* target neverssl.com dengan durasi scan selama 477.32 detik, didapatkan hasil temuan kerentanan dengan 1 kategori *medium* yaitu HTTP *Service* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* neverssl.com mendapatkan *security score* sebesar 80 yang menunjukkan bahwa *Website* sangat rentan terhadap serangan.

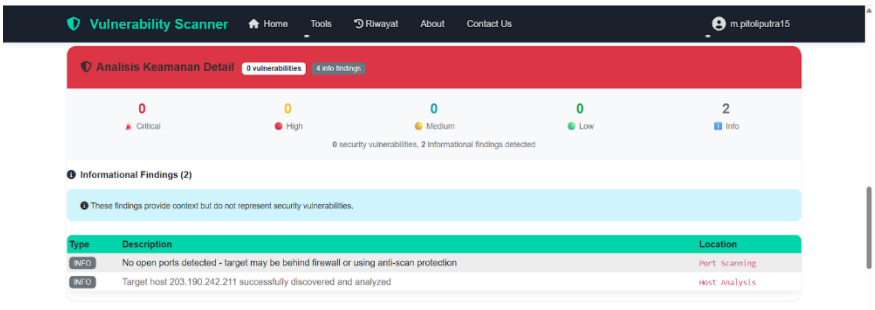


Gambar 5.57 Pengujian ketiga *Website* internet (neverssl.com) dengan nmap *Scanner*

c. *Website* 3 (www.detik.com)

- Pengujian pertama :

Gambar 5.58 merupakan hasil pengujian pertama pada *Website* target www.detik.com dengan durasi scan selama 513.35 detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* www.detik.com mendapatkan *security score* sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendappat serangan sangat rendah.

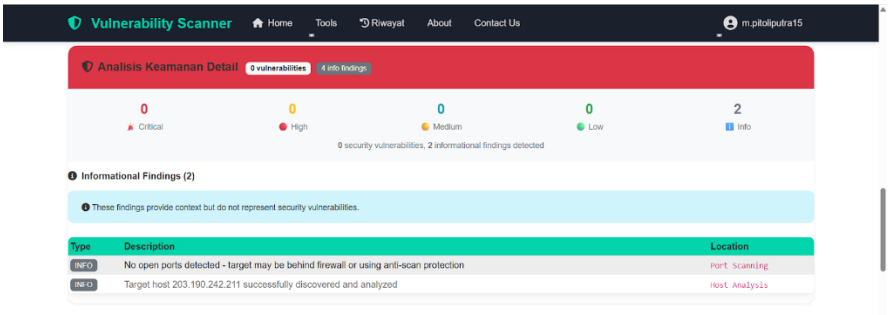


Gambar 5.58 Pengujian pertama *Website* internet (www.detik.com) dengan nmap *Scanner*

- Pengujian kedua :

Gambar 5.59 merupakan hasil pengujian kedua pada *Website* target www.detik.com dengan durasi scan selama **502.01** detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* www.detik.com mendapatkan *security*

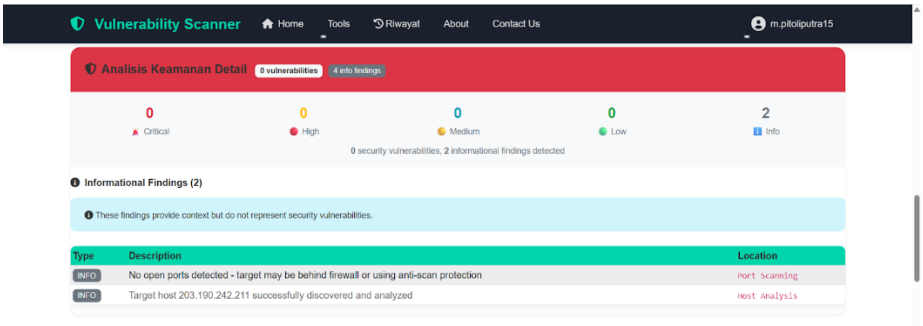
score sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendappat serangan sangat rendah.



Gambar 5.59 Pengujian kedua *Website internet* (www.detik.com) dengan nmap Scanner

- Pengujian ketiga :

Gambar 5.60 merupakan hasil pengujian ketiga pada *Website* target **www.detik.com** dengan durasi scan selama **521.33** detik, didapatkan hasil temuan kerentanan dengan 0 kategori *critical*, 0 kategori *high*, 0 kategori *medium*, 0 kategori *low* dan 2 kategori info. Dari hasil temuan kerentanan tersebut, *Website* **www.detik.com** mendapatkan *security score* sebesar 100 yang menunjukkan bahwa *Website* sangat aman dan kemungkinan mendappat serangan sangat rendah.



Gambar 5. 60 Pengujian ketiga Website internet (www.detik.com) dengan nmap Scanner

Tabel 5. 6 Pengujian *Website* Internet Menggunakan Nmap *Scanner*

No.	Waktu Pengujian			Konsistensi Hasil Temuan			Keterangan
	P1	P 2	P 3	P 1	P 2	P 3	
Website 1	176.18 detik	180.01 detik	166.12 detik	Skor kerentanan 0	Skor kerentanan 0	Skor kerentanan 0	Hasil konsisten
Website 2	481.57 detik	485.81 detik	477.32 detik	Skor kerentanan 80	Skor kerentanan 80	Skor kerentanan 80	Hasil konsisten
Website 3	513.35 detik	502.01 detik	521.33 detik	Skor kerentanan 100	Skor kerentanan 100	Skor kerentanan 100	Hasil konsisten

Berdasarkan semua pengujian di atas, didapatkan data hasil analisis rangkuman seperti yang tertera pada **Tabel 5.7** Rekap Hasil Analisis Pengujian *Website*

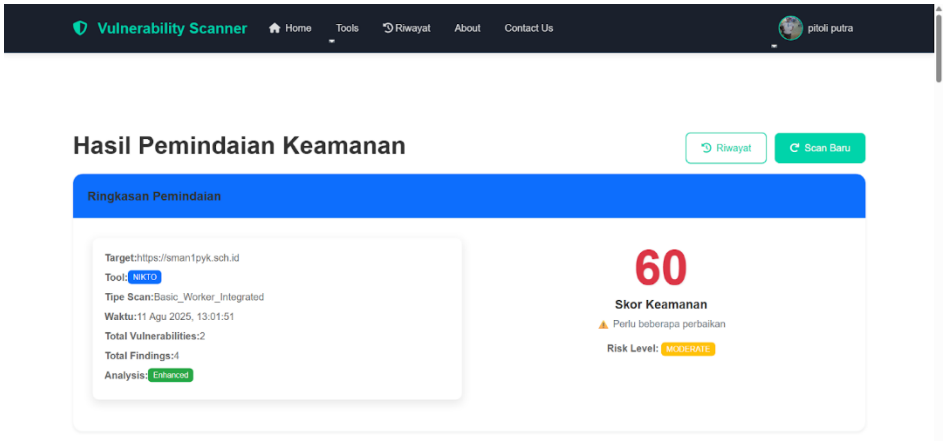
Tabel 5. 7 Rekap Hasil Analisis Pengujian *Website*

Jenis Pengujian	*TOOLS*		Hasil Pengukuran		
	NIKTO	NMAP	Rata-Rata Durasi Scan Nikto	Rata-Rata Durasi Scan Nmap	Konsisntensi Temuan
Local Host 1 *Device*	*Website* 1 (5001)	*Website* 1 (8081)	1.77 detik	174.10 detik	Konsisten
	Website 2 (5002)	*Website* 2 (8082)	0.99 detik	481.57 detik	Konsisten
	Website 3 (5003)	*Website* 3 (8083)	1.22 detik	512.23 detik	Konsisten
Local Host Beda *Device*	*Website* 1 (5001)	*Website* 1 (8081)	10.12 detik	989.25 detik	Konsisten
	Website 2 (5002)	*Website* 2 (8082)	8.72 detik	857.19 detik	Konsisten
	Website 3 (5003)	*Website* 3 (8083)	4.51 detik	439.81 detik	Konsisten
Internet	www.pln.co.id	scanme.nmap	20.18 detik	847.14 detik	Konsisten
	httpbin.org	neverssl.com	82.76 detik	791.70 detik	Konsisten
	www.komdigi.go.id	www.detik.com	23.51 detik	358.18 detik	Konsisten

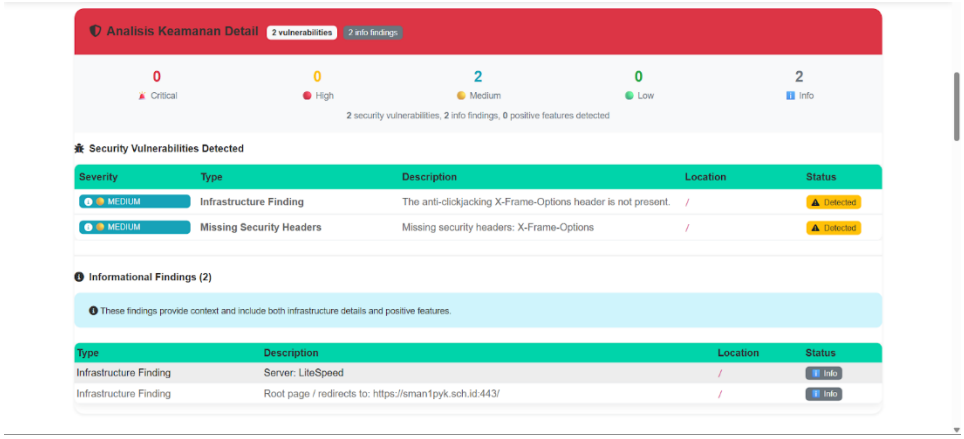
5.2.3 Pengujian Komparasi Nmap dan Nikto

5.2.3.1 Website https://sman1pyk.sch.id/

a. Menggunakan Tools Nikto



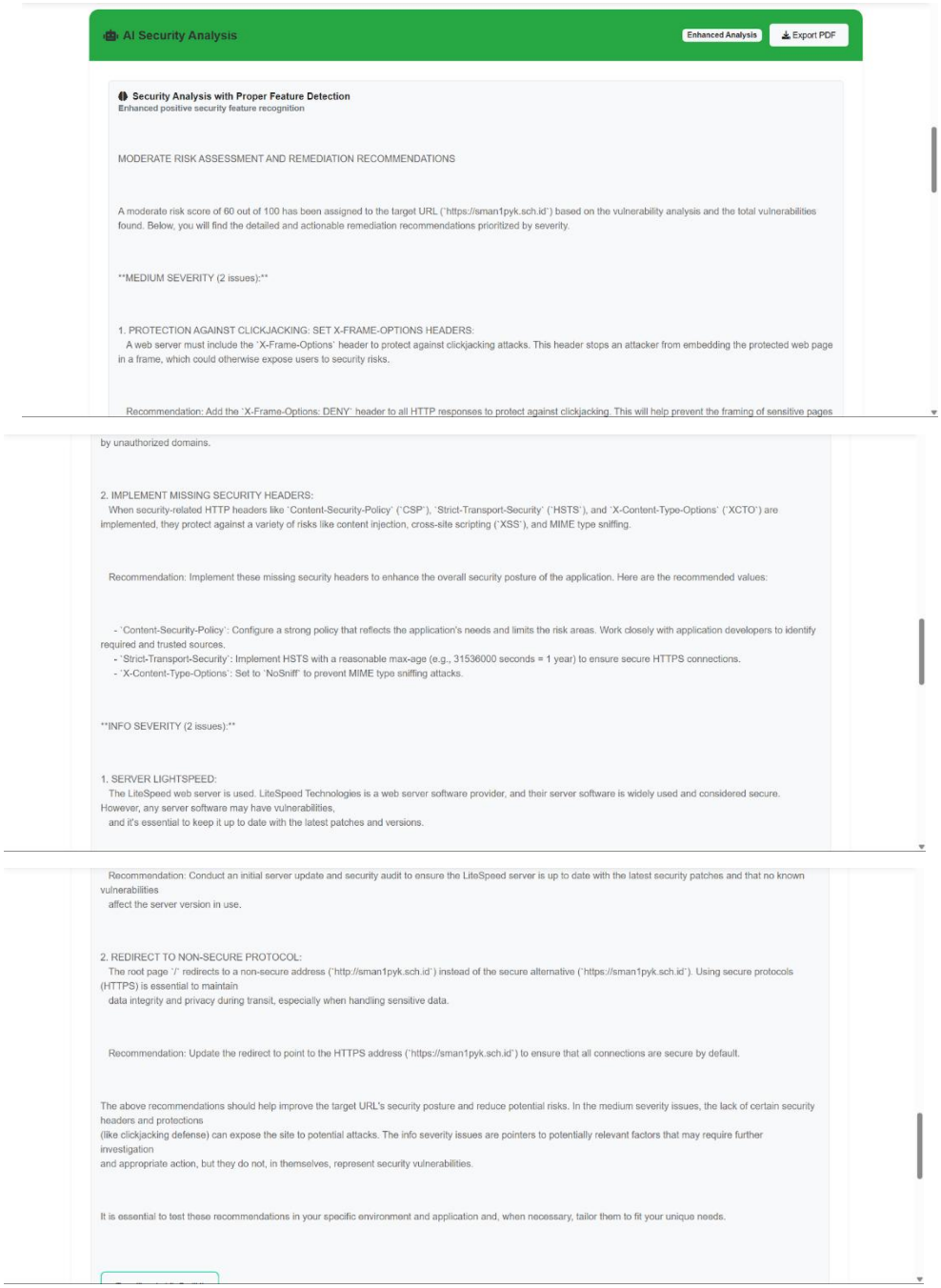
Gambar 5. 61 Skor Keamanan *Website* https://sman1pyk.sch.id/ Berdasarkan Hasil Pemindaian Nikto



Gambar 5. 62 Hasil Temuan Kerentanan *Website* Https://Sman1pyk.Sch.Id/ Berdasarkan Hasil Pemindaian Nikto

Gambar 5.61 dan Gambar 5.62 menampilkan hasil pengujian terhadap *website* target https://sman1pyk.sch.id/ menggunakan *tools* Nikto. Pengujian ini menghasilkan dua temuan dengan tingkat *medium*. Temuan pertama diidentifikasi sebagai tipe *Infrastructure Finding*, yaitu tidak adanya *header* anti-*clickjacking* X-Frame-Options. Temuan kedua diidentifikasi sebagai tipe *Missing Security Headers* dengan deskripsi yang sama, yakni ketiadaan *header* X-Frame-Options. Selain itu, ditemukan dua informasi pendukung (*informational findings*), yaitu penggunaan server LiteSpeed serta adanya pengalihan (*redirect*) dari halaman utama menuju

https://sman1pyk.sch.id:443/. Berdasarkan keseluruhan temuan, *website* ini memperoleh nilai keamanan (*security score*) sebesar 60, yang menunjukkan tingkat kerentanan yang cukup signifikan terhadap potensi serangan.

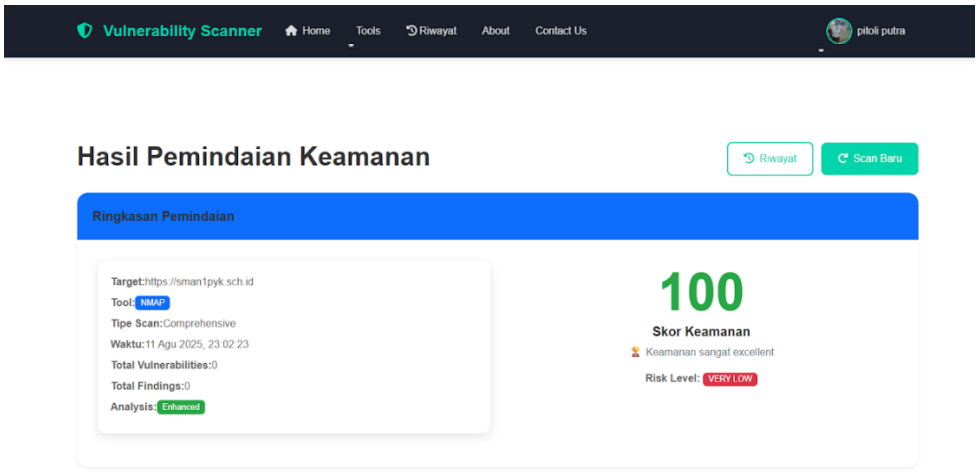


Gambar 5. 63 Hasil Rekomendasi AI *Website* https://sman1pyk.sch.id/ Berdasarkan Hasil Pemindaian Nikto

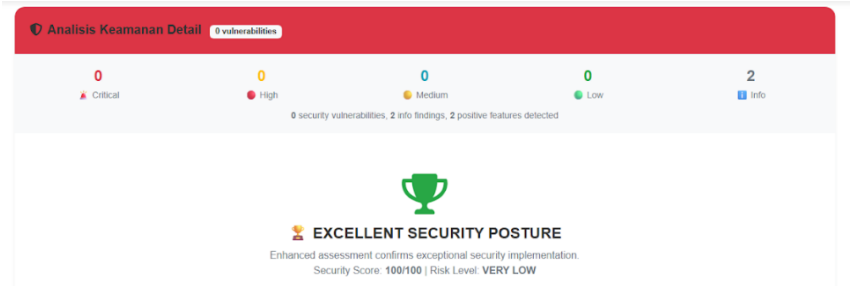
Gambar 5. 63 menampilkan rekomendasi perbaikan yang selaras dengan temuan kerentanan pada *website* target. Ketiadaan *header* keamanan X-Frame-Options direspons dengan saran untuk menambahkan X-Frame-

Options: DENY pada seluruh respons HTTP, sehingga dapat mencegah potensi serangan *clickjacking*. Temuan serupa pada tipe *Missing Security Headers* diikuti dengan rekomendasi untuk mengimplementasikan *security headers* yang hilang, guna memperkuat perlindungan aplikasi. Penggunaan server LiteSpeed diantisipasi melalui anjuran pembaruan dan audit keamanan, memastikan versi server telah mendapatkan *security patch* terbaru. Sementara itu, pengalihan dari halaman utama menuju https://sman1pyk.sch.id:443/ diatasi dengan saran memperbarui tujuan pengalihan agar langsung mengarah ke https://sman1pyk.sch.id , menjamin koneksi aman secara default melalui HTTPS.

b. Menggunakan Tools Nmap



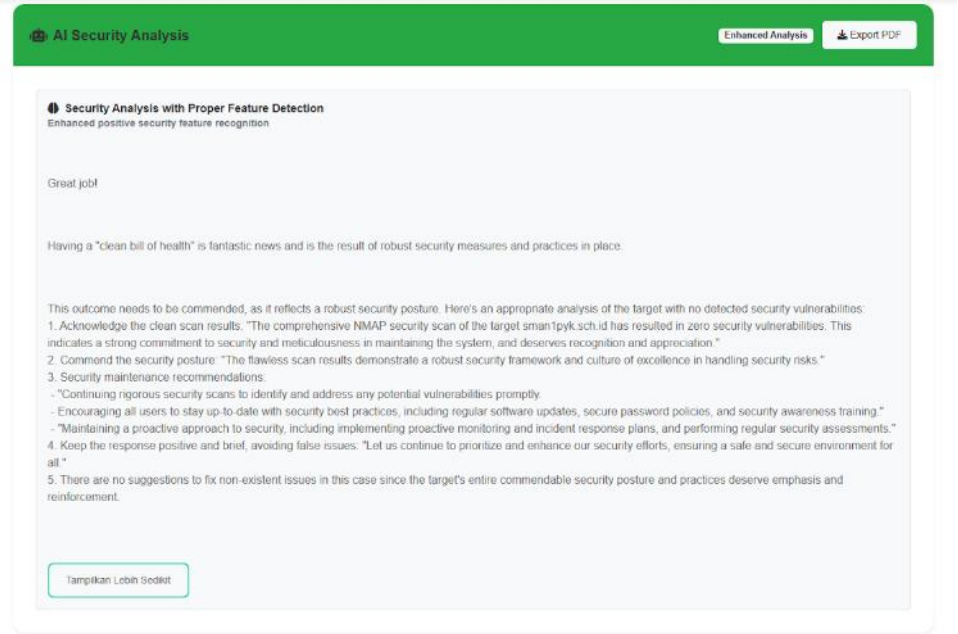
Gambar 5. 64 Skor Keamanan *Website* https://sman1pyk.sch.id/ Berdasarkan Hasil Pemindaian Nmap



Gambar 5. 65 Hasil Temuan Kerentanan *Website* Https://Sman1pyk.Sch.Id/ Berdasarkan Hasil Pemindaian Nmap

Gambar Gambar 5. 64 dan Gambar 5. 65 menampilkan hasil pengujian terhadap *website* target https://sman1pyk.sch.id/ menggunakan *tools* Nmap.

Pengujian ini tidak menemukan adanya kerentanan pada *website*. Berdasarkan hasil tersebut, *website* ini memperoleh nilai keamanan (*security score*) sebesar 100, yang menunjukkan bahwa *website* berada dalam kondisi sangat aman dan minim risiko serangan.

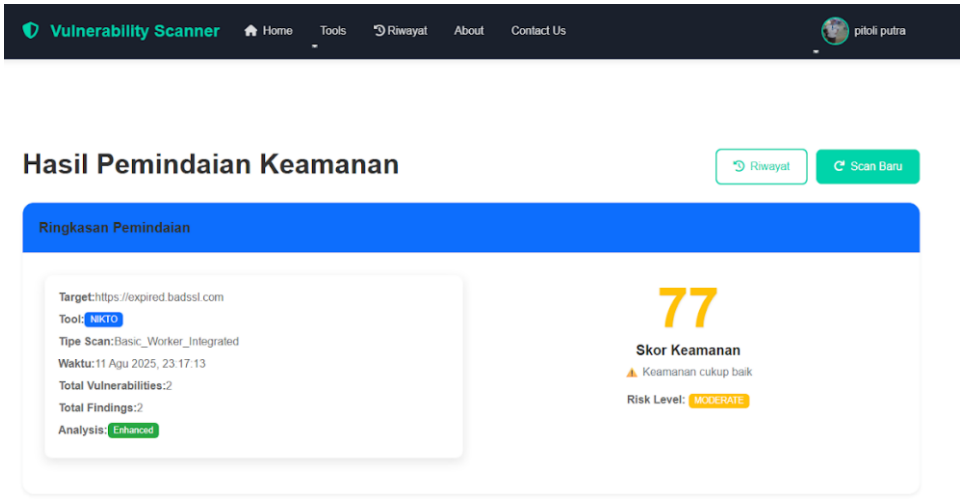


Gambar 5. 66 Hasil Rekomendasi AI *Website* https://sman1pyk.sch.id/ Berdasarkan Hasil Pemindaian Nmap

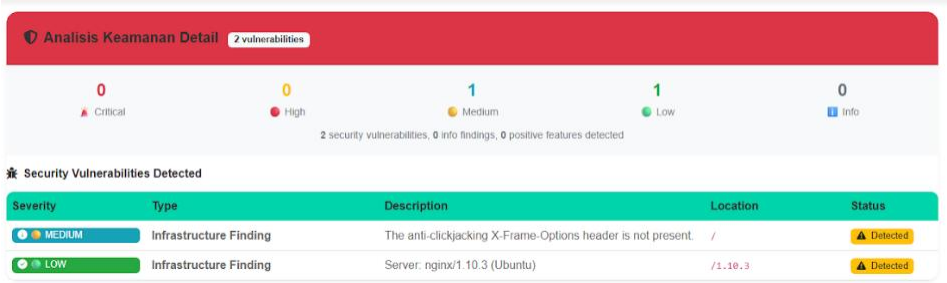
Gambar 5. 66 merupakan hasil rekomendasi dari AI untuk *website* target https://sman1pyk.sch.id/. Berdasarkan hasil pengujian, nilai kerentanan tercatat 0 atau tidak ditemukan kerentanan pada *website* ini. Meskipun demikian, rekomendasi yang diberikan tetap relevan untuk menjaga keamanan jangka panjang. Saran yang disampaikan meliputi pelaksanaan *security scan* secara ketat dan berkelanjutan untuk mengidentifikasi serta menangani potensi kerentanan secara cepat, memastikan seluruh pengguna selalu mengikuti praktik keamanan terbaik seperti pembaruan perangkat lunak secara rutin, kebijakan kata sandi yang kuat, dan pelatihan kesadaran keamanan, serta mempertahankan pendekatan keamanan proaktif melalui penerapan pemantauan berkelanjutan, rencana respons insiden, dan penilaian keamanan secara berkala.

5.2.3.2 Website https://expired.badssl.com

a. Menggunakan Tools Nikto

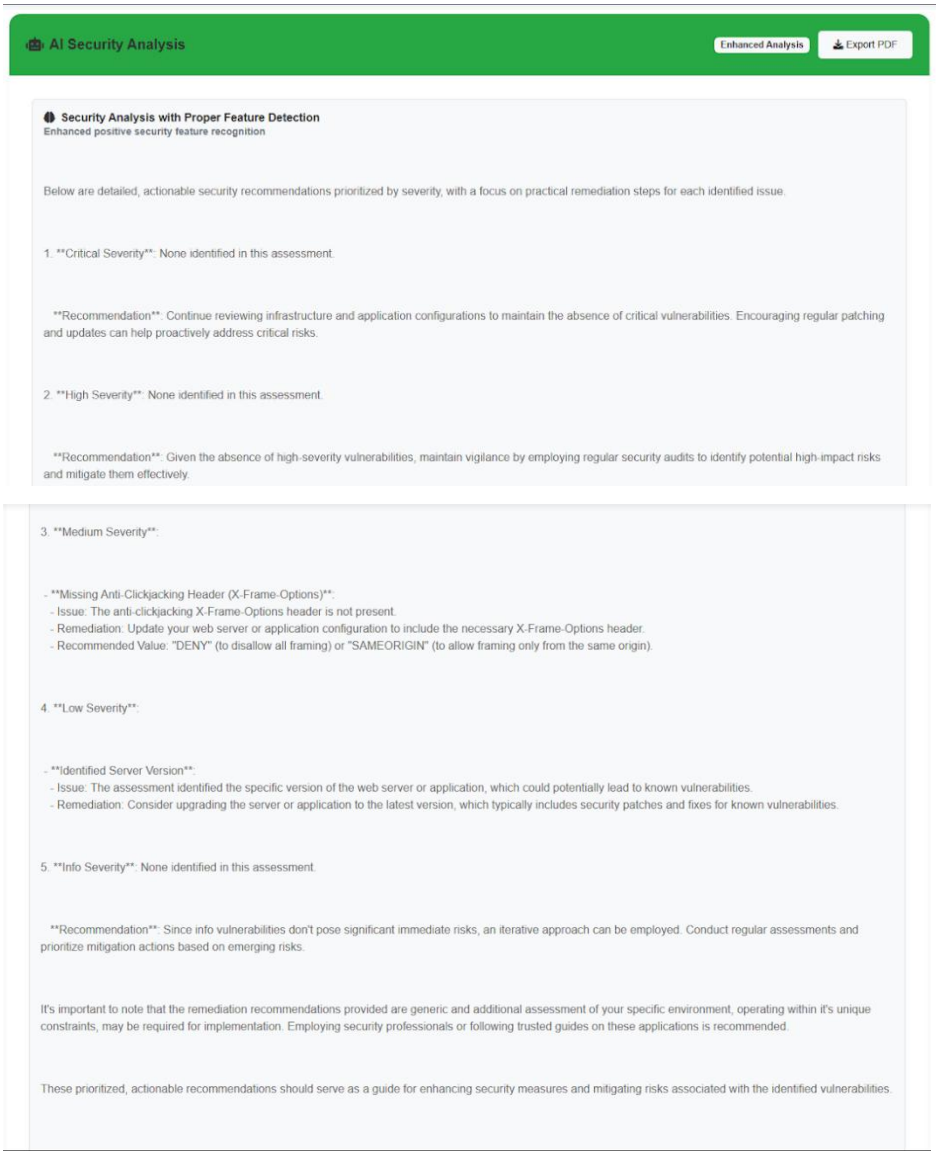


Gambar 5. 67 Skor Keamanan *Website* https://expired.badssl.com Berdasarkan Hasil Pemindaian Nikto



Gambar 5. 68 Hasil Temuan Kerentanan *Website* https://expired.badssl.com Berdasarkan Hasil Pemindaian Nikto

Gambar 5.67 dan Gambar 5.68 menampilkan hasil pengujian terhadap *website* target https://expired.badssl.com menggunakan *tools* Nikto. Pengujian ini menghasilkan dua temuan kerentanan. Temuan pertama memiliki kategori *medium* dan dikategorikan sebagai tipe *Infrastructure Finding*, yaitu tidak adanya header anti-clickjacking X-Frame-Options. Temuan kedua memiliki kategori *low* dengan tipe *Infrastructure Finding*, yang menunjukkan bahwa server menggunakan nginx/1.10.3 (Ubuntu). Berdasarkan keseluruhan temuan, *website* ini memperoleh nilai keamanan (*security score*) sebesar 77, yang menunjukkan tingkat risiko sedang (*moderate risk level*).

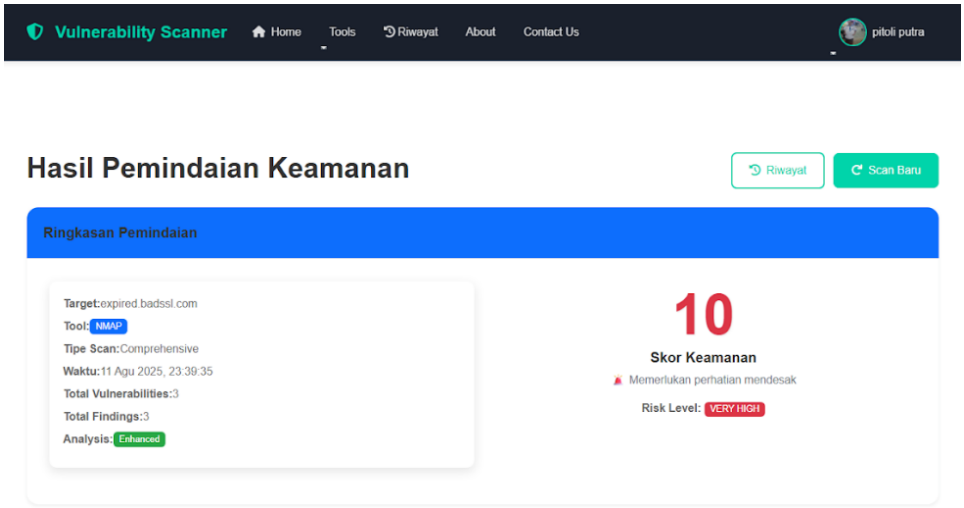


Gambar 5. 69 Hasil Rekomendasi AI *Website* Https://Expired.Badssl.Com Berdasarkan Hasil Pemindaian Nikto

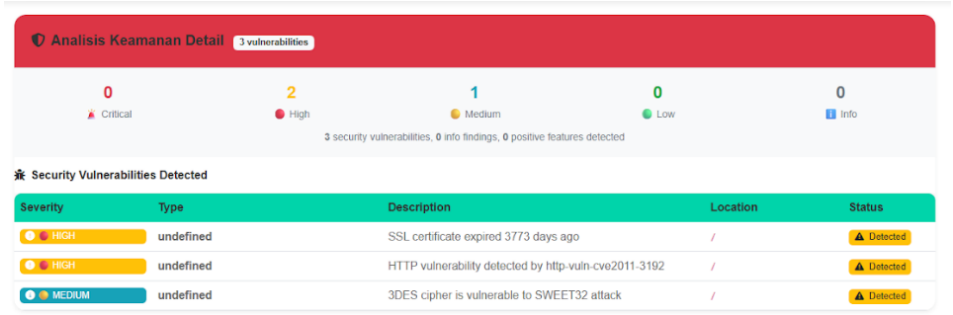
Gambar 5. 69 menampilkan hasil rekomendasi perbaikan yang sesuai dengan temuan kerentanan pada website target https://expired.badssl.com . Untuk kerentanan dengan tingkat medium, yaitu ketiadaan header anti-clickjacking X-Frame-Options, rekomendasi yang diberikan adalah memperbarui konfigurasi server atau aplikasi web agar menyertakan header X-Frame-Options dengan nilai yang disarankan, yaitu "DENY" untuk melarang seluruh framing atau "SAMEORIGIN" untuk mengizinkan framing hanya dari sumber yang sama. Untuk temuan dengan tingkat rendah yang berkaitan dengan versi server nginx/1.10.3 (Ubuntu), rekomendasi yang dianjurkan adalah melakukan pembaruan server atau

aplikasi ke versi terbaru guna memastikan penerapan patch keamanan dan perbaikan atas kerentanan yang sudah diketahui.

b. Menggunakan Tools Nmap



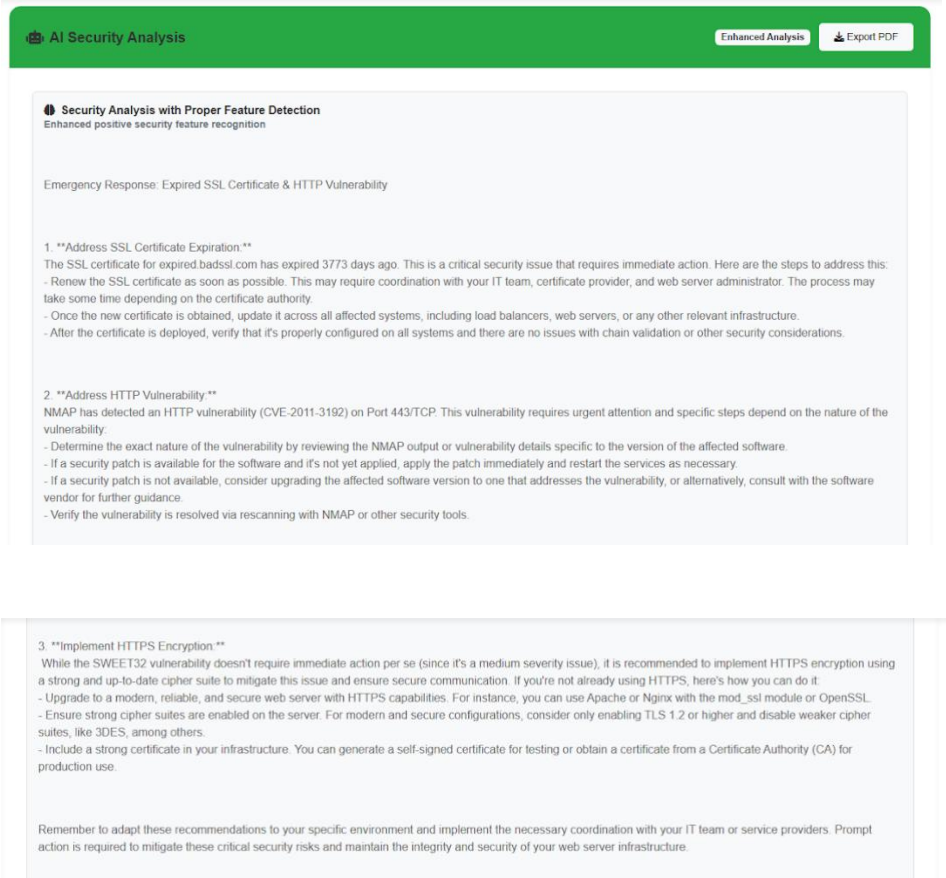
Gambar 5. 70 Skor Keamanan *Website* https://expired.badssl.com Berdasarkan Hasil Pemindaian Nmap



Gambar 5. 71 Hasil Temuan Kerentanan *Website* Https://Expired.Badssl.Com Berdasarkan Hasil Pemindaian Nmap

Gambar 5. 70 dan Gambar 5. 71 menampilkan hasil pengujian terhadap *website* target https://expired.badssl.com menggunakan *tools* Nmap. Pengujian ini mengidentifikasi tiga temuan kerentanan dengan kategori yang berbeda. Temuan pertama memiliki kategori *high* yaitu sertifikat SSL yang telah kedaluwarsa selama 3773 hari. Temuan kedua memiliki kategori *high* yaitu adanya kerentanan HTTP yang terdeteksi melalui CVE-2011-3192. Sementara itu, temuan ketiga memiliki kategori *medium* yaitu penggunaan cipher 3DES yang rentan terhadap serangan SWEET32.

Berdasarkan hasil keseluruhan, *website* ini memperoleh skor keamanan sebesar 10 dengan tingkat risiko sangat tinggi (*very high risk level*).



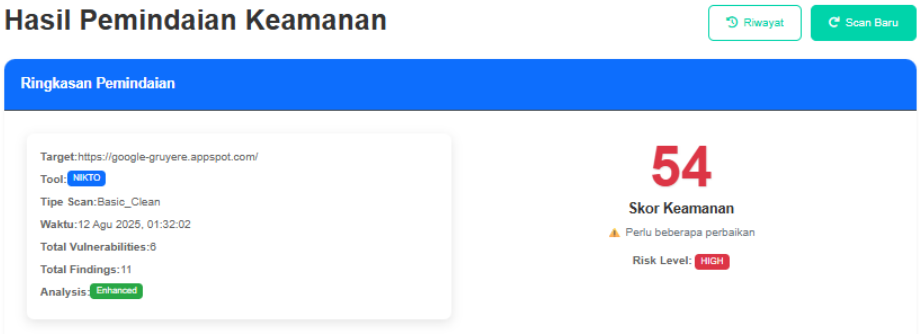
Gambar 5. 72 Hasil Rekomendasi AI *Website* Https://Expired.Badssl.Com Berdasarkan Hasil Pemindaian Nmap

Gambar 5. 72 menampilkan hasil rekomendasi perbaikan yang sesuai dengan temuan kerentanan pada *website* target https://expired.badssl.com . Untuk kerentanan terkait sertifikat SSL yang telah kedaluwarsa selama 3773 hari, rekomendasi yang diberikan meliputi pembaruan sertifikat SSL secara segera, koordinasi dengan tim TI dan penyedia sertifikat, serta verifikasi konfigurasi sertifikat setelah penerapan pada seluruh sistem terkait. Mengenai kerentanan HTTP yang terdeteksi melalui CVE-2011-3192, saran yang disampaikan adalah untuk mengidentifikasi sifat kerentanan secara tepat, menerapkan *patch* keamanan jika tersedia, melakukan pembaruan perangkat lunak jika diperlukan, dan melakukan pemindaian ulang guna memastikan kerentanan telah tertangani. Untuk isu penggunaan cipher 3DES yang rentan terhadap serangan SWEET32, dianjurkan penerapan enkripsi HTTPS dengan konfigurasi *cipher suite*

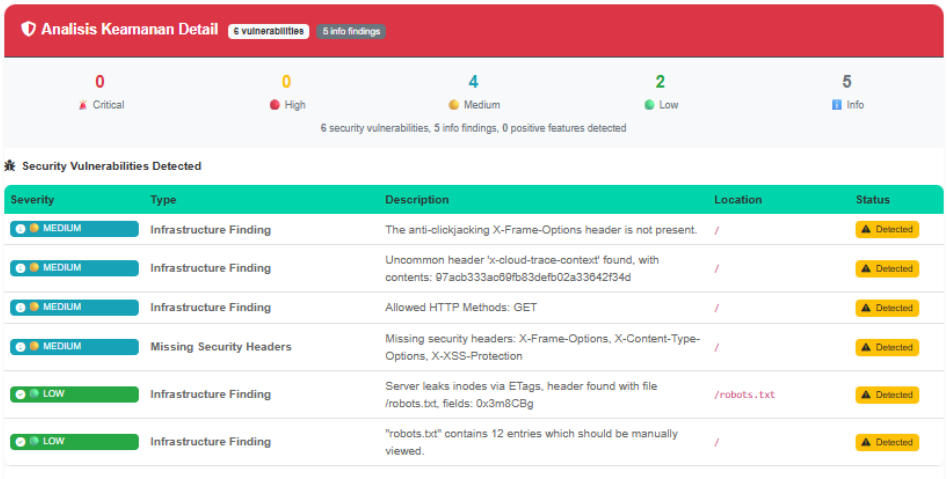
yang kuat dan mutakhir, seperti penggunaan TLS 1.2 ke atas serta menonaktifkan *cipher* yang lemah, termasuk 3DES, guna memastikan komunikasi yang aman.

5.2.3.3 Website https://google-gruyere.appspot.com/

a. Menggunakan Tools Nikto



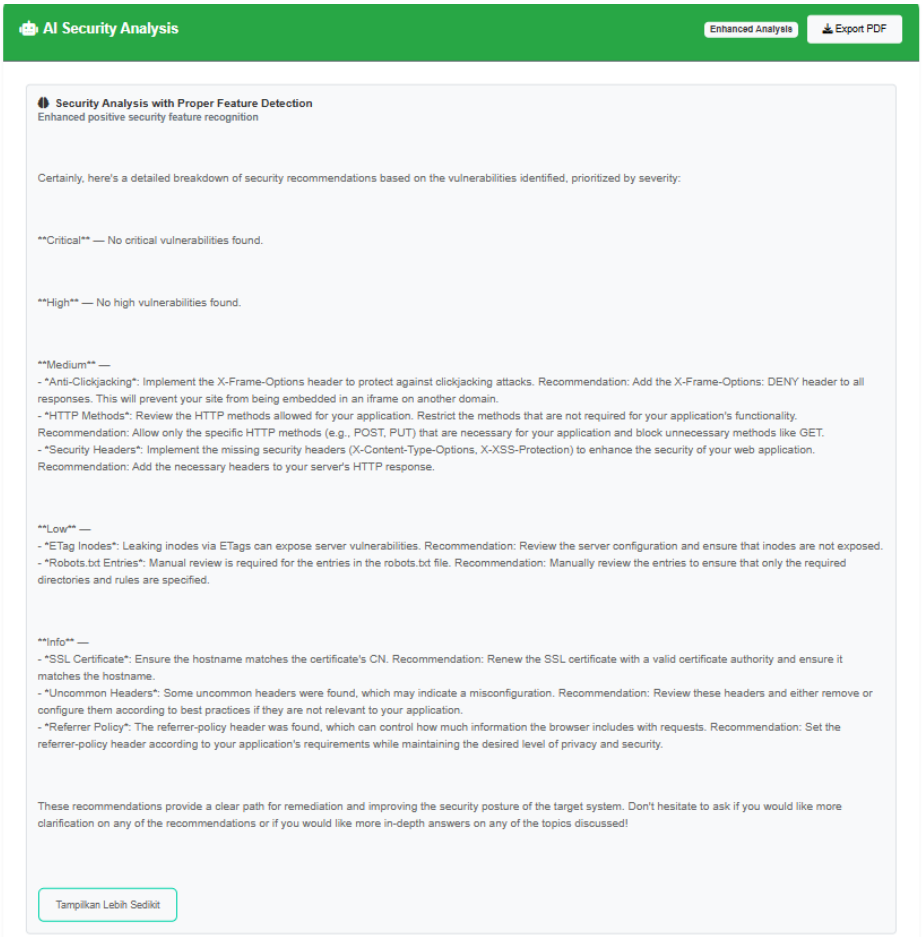
Gambar 5. 73 Skor Keamanan *Website* https://google-gruyere.appspot.com/ Berdasarkan Hasil Pemindaian Nikto



Gambar 5. 74 Hasil Temuan Kerentanan *Website* https://google-gruyere.appspot.com/ Berdasarkan Hasil Pemindaian Nikto

Gambar 5. 73 dan Gambar 5 .74 menampilkan hasil pengujian terhadap *website* target https://google-gruyere.appspot.com/ menggunakan *tools* Nikto. Pengujian ini menghasilkan enam temuan kerentanan dengan variasi kategori dan tingkat kerentanan. Terdapat empat temuan dengan kategori *medium*, yaitu ketiadaan *header anti-clickjacking X-Frame-Options*, keberadaan *header* tidak umum 'x-cloud-trace-context' dengan nilai tertentu, metode HTTP yang diizinkan hanya GET, serta ketiadaan

beberapa *header* keamanan penting seperti *X-Frame-Options*, *X-Content-Type-Options*, dan *X-XSS-Protection*. Selain itu, terdapat dua temuan dengan kategori low berupa kebocoran *inode* melalui *header* ETags pada file /robots.txt dan adanya 12 entri dalam file "robots.txt" yang perlu ditinjau secara manual. Berdasarkan keseluruhan temuan, *website* ini memperoleh nilai keamanan (*security score*) sebesar 54 dengan tingkat risiko tinggi (*high risk level*).



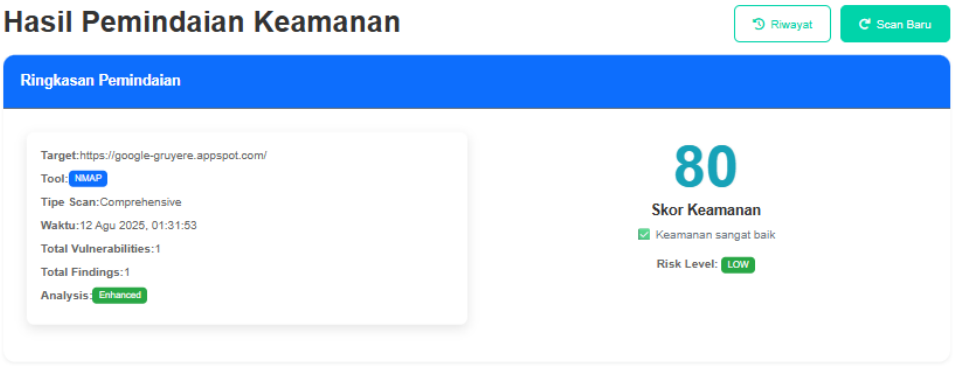
Gambar 5. 75 Hasil Rekomendasi AI *Website* https://google-gruyere.appspot.com/ Berdasarkan Hasil Pemindaian Nikto

Gambar 5. 75 menunjukkan hasil validasi rekomendasi AI terhadap temuan kerentanan pada *website* target https://google-gruyere.appspot.com/. Untuk kerentanan kategori *medium*, seperti ketiadaan *header anti-clickjacking X-Frame-Options*, disarankan penambahan *header X-Frame-Options: DENY* pada seluruh respons HTTP guna mencegah serangan *clickjacking*. Selain itu, metode HTTP

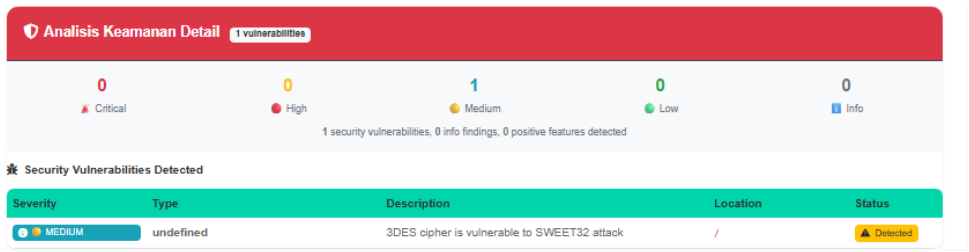
yang diizinkan (misalnya GET) perlu dibatasi hanya pada metode yang relevan dengan fungsi aplikasi, serta memblokir metode yang tidak diperlukan. Implementasi *header* keamanan tambahan seperti *X-Content-Type-Options dan X-XSS-Protection* juga direkomendasikan untuk memperkuat perlindungan aplikasi web.

Untuk temuan kategori rendah, seperti kebocoran *inode* melalui ETags, perlu dilakukan peninjauan konfigurasi server agar informasi *inodes* tidak terekspos. *Entry* pada file robots.txt juga harus ditinjau secara manual untuk memastikan hanya direktori dan aturan yang diperlukan yang dicantumkan. Dan temuan tambahan terkait *header* tidak umum dan kebijakan *referrer* disarankan untuk dikaji dan disesuaikan dengan praktik terbaik demi menjaga privasi dan keamanan aplikasi.

b. Menggunakan *Tools* Nmap



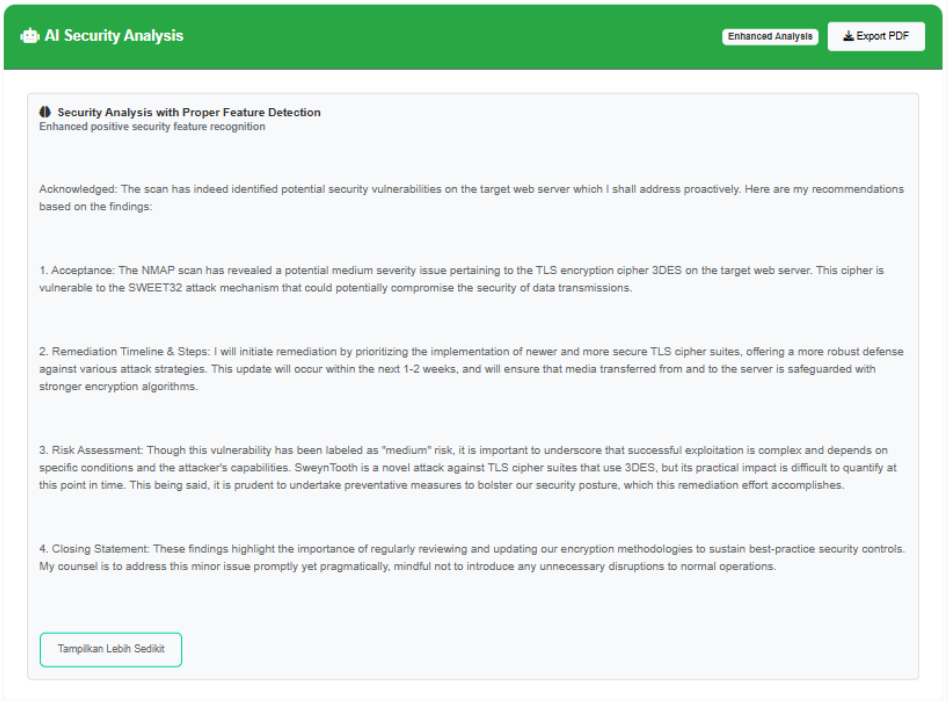
Gambar 5. 76 Skor Keamanan *Website* https://google-gruyere.appspot.com/ Berdasarkan Hasil Pemindaian Nmap



Gambar 5. 77 Hasil Temuan Kerentanan *Website* https://google-gruyere.appspot.com/ Berdasarkan Hasil Pemindaian Nmap

Gambar 5. 76 dan Gambar 5. 77 menampilkan hasil pengujian terhadap *website* target https://google-gruyere.appspot.com/ menggunakan *tools* Nmap. Pengujian ini mengidentifikasi satu temuan kerentanan dengan

kategori *medium*, yaitu penggunaan *cipher* 3DES yang rentan terhadap serangan SWEET32. Berdasarkan hasil keseluruhan, *website* ini memperoleh skor keamanan sebesar 80 dengan tingkat risiko rendah (*low risk level*).



Gambar 5. 78 Hasil Rekomendasi AI *Website* https://google-gruyere.appspot.com/ Berdasarkan Hasil Pemindaian Nmap

Gambar 5. 78 menampilkan hasil validasi rekomendasi AI yang sesuai dengan temuan kerentanan pada *website* target https://google-gruyere.appspot.com/. Temuan kategori *medium* tentang penggunaan *cipher* 3DES yang rentan terhadap serangan SWEET32 merupakan masalah keamanan yang perlu ditangani. Rekomendasi yang diberikan menekankan penerapan *cipher* TLS yang lebih baru dan kuat sebagai langkah mitigasi, dengan rencana perbaikan dalam jangka waktu 1-2 minggu guna memastikan data yang ditransfer ke dan dari server terlindungi dengan algoritma enkripsi yang lebih aman.

5.2.3.4 Hasil Analisis Pengujian Komparasi Nikto dan Nmap

Tabel 5. 8 Rekapitulasi Hasil Pengujian Komparasi Nikto dan Nmap

Website	***Security Score***				**Validasi Rekomendasi AI**
	Nikto	***Risk Level***	**Nmap**	***Risk Level***	
https://sman1pyk.sch.id/	60	*Moderate*	100	*Very Low*	Sesuai dengan temuan
https://expired.badssl.com	77	*Moderate*	10	*Very High*	Sesuai dengan temuan
https://google-gruyere.appspot.com/	54	*High*	80	*Low*	Sesuai dengan temuan

Tabel 5.8 menunujukan hasil pengujian yang dilakukan terhadap tiga *website* menggunakan dua *tools* berbeda, yaitu *Nmap* dan *Nikto*, ditemukan bahwa pengujian yang dilakukan pada target yang sama, skor keamanan (*security score*) dan tingkat risiko (*risk level*) yang dihasilkan menunjukkan perbedaan yang cukup signifikan. Misalnya, pada *website* https://expired.badssl.com, *Nikto* memberikan skor 77 dengan tingkat risiko *Moderate*, sedangkan *Nmap* memberikan skor 10 dengan tingkat risiko *Very High*. Perbedaan ini juga terlihat pada website lainnya, di mana skor dan tingkat risiko yang diberikan oleh masing-masing tools berbeda.

Perbedaan ini secara teknis dapat dijelaskan melalui karakteristik dan fokus utama dari kedua *tools* tersebut. *Nmap* berfungsi sebagai *network scanner* yang berfokus pada pemetaan *port*, layanan, dan konfigurasi jaringan. Temuan *Nmap* berfokus pada kerentanan yang terkait dengan *network exposure*, konfigurasi *port*, dan protokol yang digunakan. Sementara itu, *Nikto* adalah *web vulnerability scanner* yang secara spesifik menilai kerentanan pada sisi aplikasi *website*, seperti konfigurasi *server HTTP*, file berbahaya, potensi injeksi, dan masalah keamanan pada halaman *web*. Perbedaan ruang lingkup inilah yang menyebabkan hasil pengujian terhadap target yang sama dapat berbeda secara signifikan baik dari sisi skor maupun tingkat risiko.

Berdasarkan analisis lebih lanjut, hasil temuan kerentanan kemudian dikategorikan sesuai dengan tingkat keparahan berdasarkan *CVSS (Common Vulnerability Scoring System)* yang membagi kerentanan menjadi empat

kategori, yaitu *Critical*, *High*, *Low*, dan *Info*. Penilaian skor keamanan pada tabel ini dihitung menggunakan metode *OWASP Risk Rating Methodology*, yaitu dengan mempertimbangkan seberapa besar peluang celah keamanan tersebut dapat dimanfaatkan dan seberapa besar dampaknya jika terjadi serangan. Perbedaan fokus deteksi kedua *tools* membuat hasil skornya berbeda meskipun diuji pada target yang sama.

Dari ketiga pengujian yang telah dilakukan, rekomendasi perbaikan yang dihasilkan oleh AI terbukti sesuai dan relevan dengan kerentanan yang ditemukan oleh kedua tools. Pada https://sman1pyk.sch.id, rekomendasi AI menanggapi temuan seperti ketiadaan *security header* dan pengalihan yang tidak optimal dengan saran implementasi konfigurasi yang tepat, pembaruan *server*, serta praktik keamanan proaktif. Pada *https://expired.badssl.com*, AI secara akurat memberikan langkah mitigasi untuk *SSL certificate* yang kedaluwarsa, kerentanan *HTTP*, dan penggunaan *cipher* lemah, termasuk pembaruan *certificate*, penerapan *security patch*, dan penguatan konfigurasi enkripsi. Sementara itu, pada *https://google-gruyere.appspot.com/*, AI menyesuaikan rekomendasi dengan kelemahan seperti *security header* yang hilang, konfigurasi metode *HTTP* yang berlebihan, kebocoran *inode* melalui *ETags*, serta masalah pada *robots.txt* dan kebijakan *referrer*, dengan memberikan langkah teknis yang tepat untuk mitigasi. Keseluruhan hasil ini menunjukkan bahwa rekomendasi tersebut sesuai dengan temuan yang telah diidentifikasi pada masing-masing *website*.

5.2.4 Pengujian Reliability Menggunakan ApacheJmeter

Pengujian menggunakan apache JMeter pada Platform Pengujian Keamanan Server Berbasis *Website* bertujuan untuk melihat performa platform ini dalam menjalankan tugasnya. Pengujian ini dilakukan sebanyak 6 kali dengan jumlah *request* user yang berbeda beda. Terdapat beberapa hal yang diuji pada pengujian menggunakan apache JMeter ini diantaranya :

- *Stability* = Kemampuan sistem untuk tetap berjalan/online tanpa crash, hang, atau shutdown
- *Reliability* = Kemampuan sistem untuk memberikan hasil yang benar/sukses secara konsisten

- *Scalability* = Kemampuan sistem untuk handle beban yang meningkat dengan menambah kapasitas secara proporsional
- *Performance* = Kecepatan sistem dalam memproses dan merespons

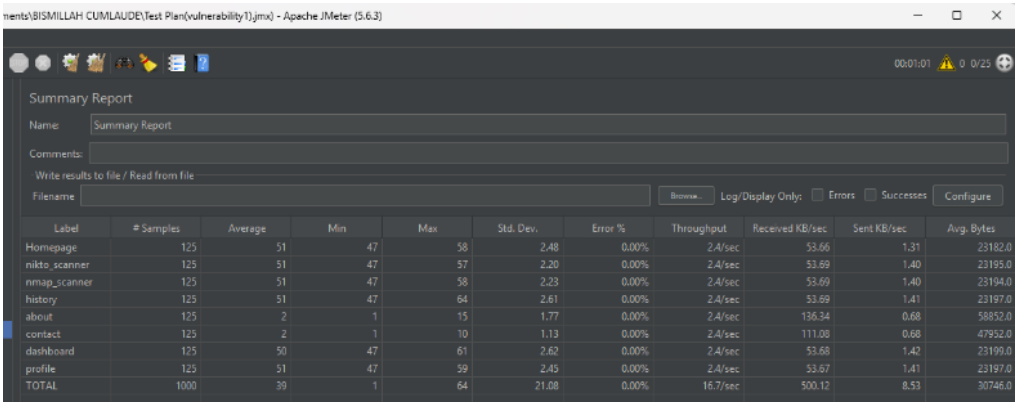
Tabel 5.9 merupakan pengujian menggunakan apache Jmeter yang dilakukan sebanyak 6 kali pengujian dengan data skenario seperti pada tabel.

Tabel 5.9 Rekap Hasil Pengujian Performa Website

Pengujian	Jumlah User	Waktu	Loops
1	25 users	10 detik	5 loops
2	50 users	30 detik	5 loops
3	100 user	60 detik	5 loops
4	200 users	120 detik	5 loops
5	800 users	60 detik	5 loops
6	1200 users	60 detik	5 loops

a. Pengujian 1

pengujian 1 menggunakan 25 *user*, 10 detik dan 5 loops untuk melihat performa platform pengujian keamanan server berbasis *Website* ini.



Gambar 5. 79 Pengujian 1

Gambar 5. 79 merupakan hasil pengujian berupa Total *request* sebanyak 1.000 *Request*, Total error rate sebesar 0.00%, Average Respon Time selama 39 ms dan Throughput sebanyak 16.7 *request*/detik.

b. Pengujian 2

Pengujian 2 menggunakan 50 *user*, 30 detik dan 5 loops untuk melihat performa platform pengujian keamanan server berbasis *Website* ini. Berdasarkan pengujian tersebut didapatkan hasil pengujian berupa Total *request*

sebanyak 2.000 *Request*, Total error rate sebesar 0.00%, Average Respon Time selama 40 ms dan Throughput sebanyak 28.4 *request*/detik.

c. Pengujian 3

Pengujian 3 menggunakan 100 *user*, 60 detik dan 5 loops untuk melihat performa platform pengujian keamanan server berbasis *Website* ini. Berdasarkan pengujian tersebut didapatkan hasil pengujian berupa Total *request* sebanyak 4.000 *Request*, Total error rate sebesar 0.00%, Average Respon Time selama 40 ms dan Throughput sebanyak 39.9 *request*/detik.

d. Pengujian 4

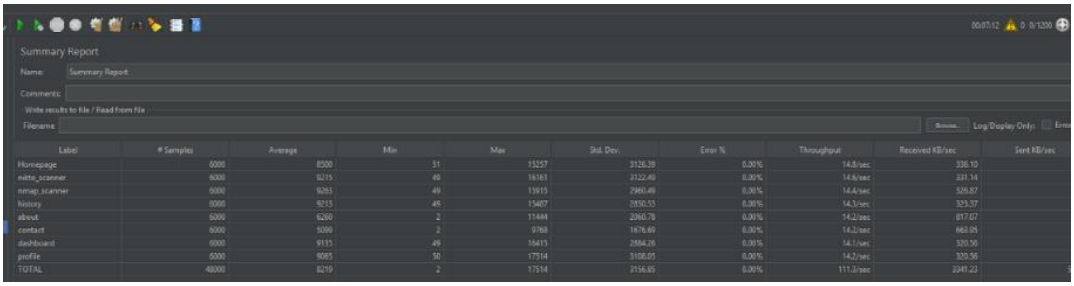
Pengujian 4 menggunakan 200 *user*, 120 detik dan 5 loops untuk melihat performa platform pengujian keamanan server berbasis *Website* ini. Berdasarkan pengujian tersebut didapatkan hasil pengujian berupa Total *request* sebanyak 8.000 *Request*, Total error rate sebesar 0.00%, Average Respon Time selama 42 ms dan Throughput sebanyak 49.9 *request*/detik.

e. Pengujian 5

Pengujian 5 menggunakan 800 *user*, 60 detik dan 5 loops untuk melihat performa platform pengujian keamanan server berbasis *Website* ini. Berdasarkan pengujian tersebut didapatkan hasil pengujian berupa Total *request* sebanyak 32.000 *Request*, Total error rate sebesar 0.00%, Average Respon Time selama 4539 ms dan Throughput sebanyak 114.2 *request*/detik.

f. Pengujian 6

Pengujian 6 menggunakan 1200 *user*, 60 detik dan 5 loops untuk melihat performa platform pengujian keamanan server berbasis *Website* ini.



Gambar 5.80 Pengujian 6

Gambar 5.80 merupakan hasil pengujian berupa Total *request* sebanyak 48.000 *Request*, Total *error rate* sebesar 0.00%, *Average Respon Time* selama 8219 ms dan *Throughput* sebanyak 111.3 *request*/detik.

Berdasarkan 6 pengujian menggunakan apache JMeter di atas, didapatkan data hasil analisis sebagai berikut :

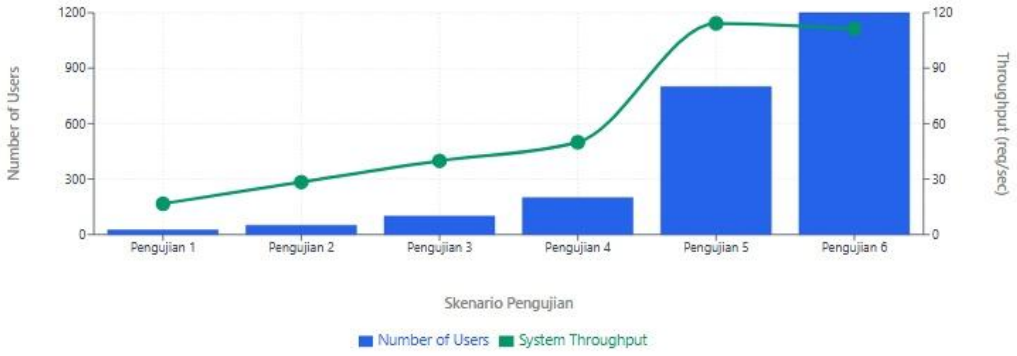
Tabel 5.10 Rekap Hasil Pengujian Mengunakan Apache JMeter

Pengujian	**Hasil Pengujian**			
	Total *Request*	**Total *Error Rate***	***Average Respon Time***	**Throughput**
1	1.000 *Request*	0.00%	39 ms	16.7 *request*/detik
2	2.000 *Request*	0.00%	40 ms	28.4 *request*/detik.
3	4.000 *Request*	0.00%	40 ms	39.9 *request*/detik
4	8.000 *Request*	0.00%	42 ms	49.9 *request*/detik
5	32.000 *Request*	0.00%	4539 ms	114.2 *request*/detik
6	48.000 Request	0.00%	8219 ms	111.3 *request*/detik

Tabel 5.11 Hasil Analisis pengujian menggunakan Apache JMeter

STABILITY	***SCALABILITY***	***RELIABILITY***	***PERFORMANCE***
Stabil *(no crashes)*	*Good (linear throughput)*	*Perfect* (0% *error*)	S*low* response ketika diberi *request* yang sangat tinggi di 32.000 – 48.000

Gambar 5. 81 Grafik *Average Response Time*



Gambar 5.82 Grafik *System Throughput*

Tabel 5.10, Tabel 5.11, Gambar 5.81, dan Gambar 5.82 menunjukan bahwa *Platform* Pengujian Keamanan Server Berbasis *Website* ini reliability yang baik karena ketika diuji dengan berbagai input berdasarkan data di atas, *Website* mampu merespons semua permintaan dengan sukses, ditunjukkan oleh tingkat error 0% dan tidak adanya crash, sehingga *Website* berjalan stabil. Selain itu, ketika jumlah *request* ditingkatkan, nilai throughput juga ikut meningkat, yang menandakan kemampuan scalability *Website* cukup bagus. Namun, terdapat sedikit penurunan throughput ketika diberikan 120 *request* dalam 60 detik. Namun, Saat jumlah *request* yang diterima sangat besar, performa *Website* mengalami penurunan yaitu respon menjadi lebih lambat.

5.2.5 Pengujian Backend

Pengujian performa Backend pada *platform* Pengujian Keamanan server Berbasis Website ini dilakukan dengan tujuan memastikan bahwa Backend dari *platform* yang telah dirancang dapat diakses dengan baik sehingga mampu menampung pengguna yang mengakses website ini dengan jumlah banyak secara bersamaan, dan pengujian ini juga dilakukan untuk menganalisa batas kemampuan website ketika di akses pengguna yang banyak. Dalam proses ini menggunakan ApacheJMeter sebagai alat uji peforma (*load testing*) untuk melakukan simulasi aktivitas pengguna yang mengakases website pada waktu yang bersamaan.

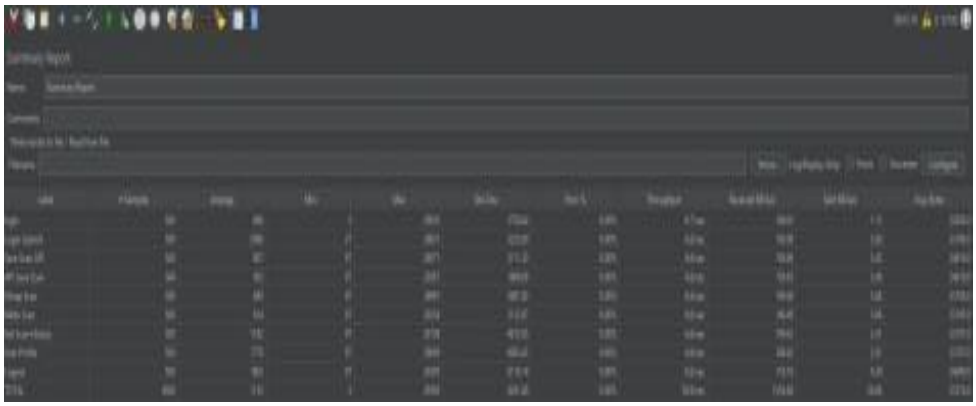
Skenario pengujian dimulai dengan melakukan konfigurasi ApacheJMeter, yaitu menentukan jumlah pengguna yang mengakses website per-30 detik, konfigurasi seperti ini bertujuan untuk melihat bagaimana performa Backend website pada banyak pengguna yang di konfigurasi per-30 detik dan nantinya jumlah pengguna akan ditambah secara terus menerus sampai menemukan titik batas kemampuan Backend website (*breaking point*).

5.2.5.1 Hasil Pengujian *Load Testing*

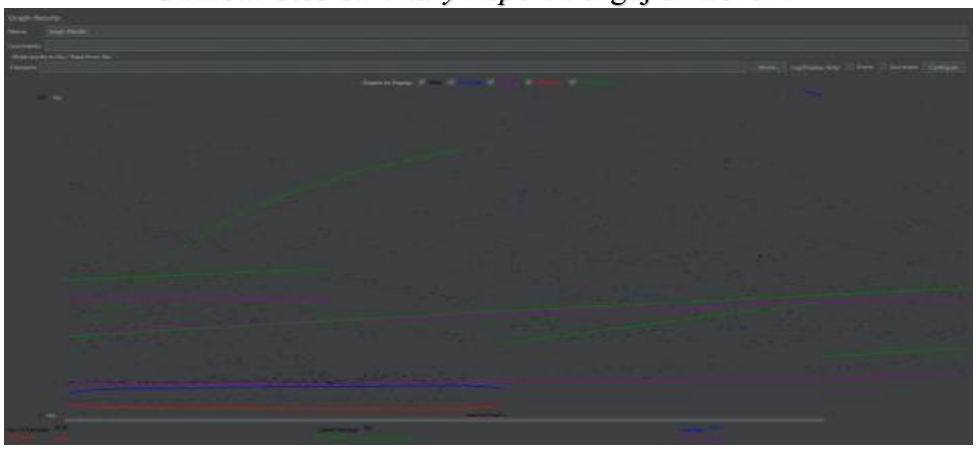
Pada hasil pengujian performa *backend* (*load testing*) ini,pengujian dilakukan menggunakan *software* ApacheJMeter. Pengujian dilakukan sebanyak 5 kali dengan level yang berbeda.

1. Pengujian 1

   pengujian 1 ini menggunakan 100 pengguna dalam waktu 30 detik dan 5 kali pengulangan untuk melihat peforma backend pada platform pengujian keamanan server berbasis website ini.



Gambar 5.83 *Summary Report* Pengujian Level 1



Gambar 5.84 *Graph Result* Pengujian Level 1

2. Pengujian 2

   pengujian 2 menggunakan 250 pengguna dalam waktu 30 detik dan 5 kali pengulangan untuk melihat peforma backend pada platform pengujian keamanan server berbasis website ini.

3. Pengujian 3

pengujian 3 menggunakan 500 pengguna dalam waktu 30 detik dan 5 kali pengulangan untuk melihat peforma backend pada *platform* pengujian keamanan server berbasis website ini.

4. Pengujian 4

pengujian 4 menggunakan 1000 pengguna dalam waktu 30 detik dan 5 kali pengulangan untuk melihat peforma backend pada *platform* pengujian keamanan server berbasis website ini.

5. Pengujian 5

pengujian 5 menggunakan 2000 pengguna dalam waktu 30 detik dan 5 kali pengulangan untuk melihat peforma *backend* pada platform pengujian keamanan *server* berbasis website ini.

Berdasarkan 5 pengujian di atas didapatkan hasil seperti pada tabel berikut :

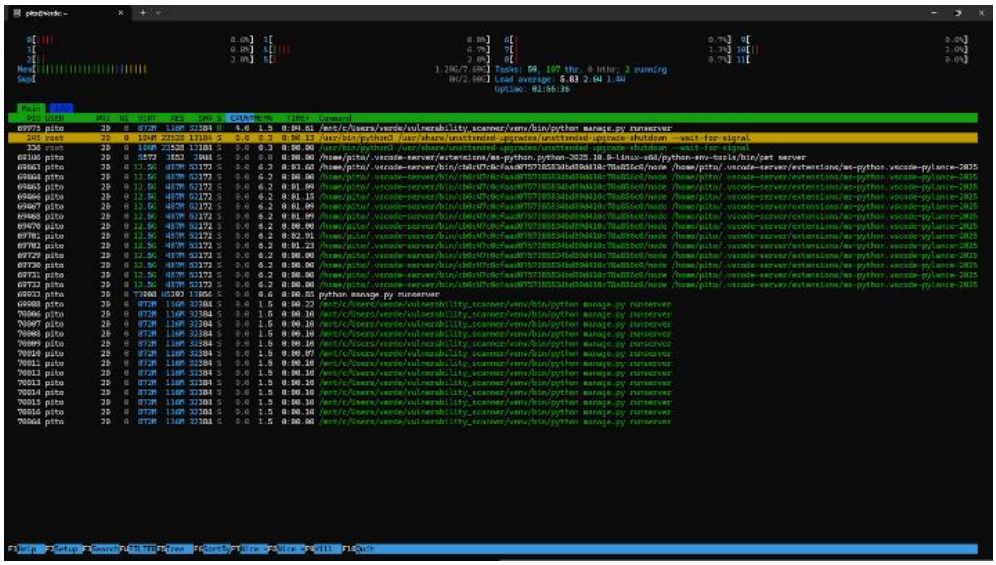
Tabel 5.12 data pengujian *load testing* sistem *backend*

Skenario	Users	Response Time (ms)	Throughput (req/s)	ErrorRate
Pengujian 1	100	1152	58.9	0.00%
Pengujian 2	250	934	154.1	0.00%
Pengujian 3	500	2561	153.3	0.00%
Pengujian 4	1000	6449	140.2	0.00%
Pengujian 5	2000	12716	147.4	13.42%

Tabel 5.12 merupakan hasil pengujian ApacheJMeter, performa backend pada *platform* pengujian keamanan server berbasis website yang sangat baik dan stabil hingga 1000 *concurrent users* dengan *response time* berkisar 696-7022 ms dan *error rate* 0.00%, menandakan sistem dapat menangani beban kerja normal hingga *stress test* tanpa kegagalan. *Throughput* konsisten pada level 140- 154 req/s untuk beban medium hingga *stress test*, menunjukkan sistem memiliki kapasitas optimal sekitar 1000 *users concurrent*. Namun, pada *Extreme Load* (2000 users), terjadi degradasi signifikan dengan *response time* melonjak drastis ke 9184-14,716 ms dan *error rate* mencapai 8.8-19.3%, mengindikasikan *breaking point* sistem berada di sekitar 1500-2000 *users concurrent*. Kesimpulannya, website dapat beroperasi dengan sangat baik untuk penggunaan normal

hingga 1000 *users*, namun memerlukan optimasi infrastruktur atau *load balancing* untuk menangani *traffic* ekstrem di atas 1500 *users concurrent.*

5.2.5.2 Monitoring Penggunaan CPU dan Memori pada Backend



Gambar 5.85 Monitoring Penggunaan CPU dan Memori

Gambar 5.85 merupakan monitoring penggunaan CPU dan memori pada sistem backend ini menggunakan *tools* HTOP pada *command prompt*, untuk mendapatkan hasil yang maksimal, penulis menggunakan metode pengambilan data dengan cara mengamati data HTOP selama 5 menit dan mencatat data sebanyak 5 kali termasuk data penggunaan CPU dan memori minimum sampai data penggunaan CPU dan memori maksimum. Untuk *monitoring* ini dilakukan di 3 kondisi yang berbeda yaitu: website dalam keadaan normal (tidak ada proses yang berjalan), website dalam keadaan proses *scanning* dengan *tools* nikto dan terakhir website dalam keadaan proses *scanning tools* nmap.

Tabel 5.13 Data Penggunaan CPU

Kondisi	CPU Min	CPU Max	CPU AVG
Normal	4.0%	12.0%	7.44%
Scan Nikto	7.3%	11.4%	9.22%
Scan Nmap	6.7%	12.0%	9.84%

Tabel 5.14 Data Pengunaan Memori

Kondisi	Memori Min	Memori Max	Memori AVG
Normal	1.5%	1.5%	1.50%
Scan Nikto	1.6%	1.7%	1.62%
Scan Nmap	1.7%	1.8%	1.76%

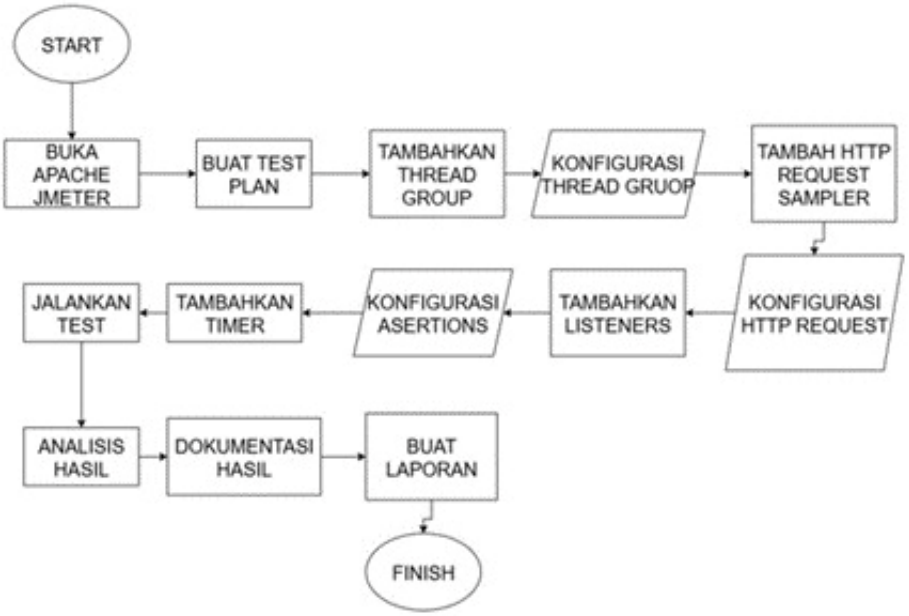
Tabel 5.13 dan Tabel 5.14 merupakan monitoring penggunaan CPU dan memori. Maka, dapat disimpulkan bahwa tentunya penggunaan CPU dan memori paling rendah berada pada kondisi website dalam keadaan normal karena website tidak sedang dalam proses apapun, tetapi jika dibandingkan dalam kondisi proses *scanning*, *scan* nmap lebih banyak menggunakan CPU dan memori dibandingkan *scan* nikto. Penggunaan CPU dan memori *scan* nmap sedikit lebih tinggi karena nmap melakukan pekerjaan yang lebih kompleks di level jaringan yang lebih rendah, sementara nikto fokus pada level aplikasi yang lebih sederhana dan efisien.

5.2.6 Pengujian Frontend

Pengujian antarmuka *website* dilakukan dengan tujuan memastikan bahwa *platform* pengujian keamanan server yang telah dikembangkan dapat diakses dengan baik dan mampu menangani sejumlah permintaan secara bersamaan. Dalam proses ini, digunakan ApacheJMeter sebagai alat uji performa (*load testing*) untuk melakukan simulasi aktivitas pengguna yang mengakses *website* pada waktu yang bersamaan. Berikut merupakan skenario pengujian dan *flowchart* pengujian yang tertera pada Tabel 5.14 dan Gambar 5.68 *Flowchart* alur pengujian *Frontend.*

Tabel 5. 15 Skenario Pengujian *Interface*

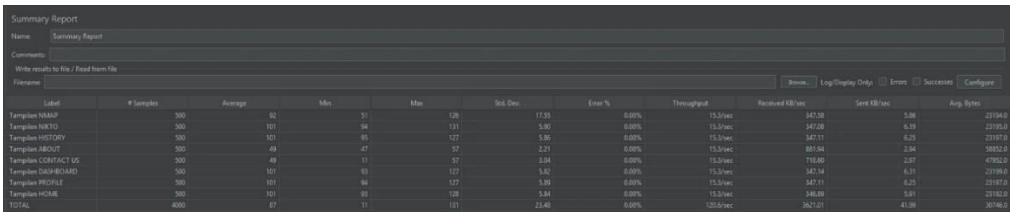
Pengujian	Jumlah User	Waktu	Loops
1	100 users	30 detik	5 loops
2	250 users	30 detik	5 loops
3	500 users	30 detik	5 loops
4	1000 users	30 detik	5 loops
5	2000 users	30 detik	5 loops



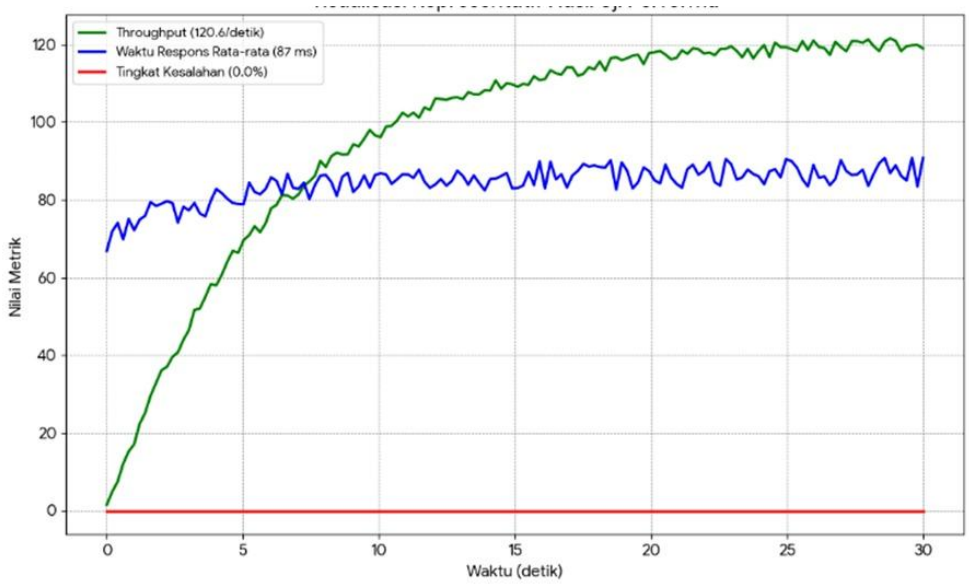
Gambar 5.86 *Flowchart* Alur Pengujian *Frontend*

1. Pengujian 1

pengujian 1 menggunakan 100 *user*, 30 *second* dan 5 *loops* untuk melihat performa *platform* pengujian keamanan server berbasis website ini. Gambar 5.87 dan Gambar 5.88 menunjukan bahwa dari pengujian ini didapatkan hasil *average respon time* selama 87 ms, *throughput* sebesar 120.6/sec, *error rate* sebesar 0.00%.



Gambar 5.87 Hasil Pengujian 1



Gambar 5.88 Grafik Hasil Pengujian 1

2. Pengujian 2

Pada pengujian 2 menggunakan 250 *user*, 30 *second* dan 5 *loops* untuk melihat performa *platform* pengujian keamanan server berbasis website ini. Dari pengujian tersebut didapatkan hasil *average* respon time selama 564 ms, *throughput* sebesar 187.6/sec, *error rate* sebesar 0.00%.

3. Pengujian 3

Pada pengujian 3 ini menggunakan 500 *user*, 30 *second* dan 5 *loops* untuk melihat performa *platform* pengujian keamanan server berbasis website ini. Dari pengujian tersebut didapatkan hasil *average respon time* selama 2113 ms, *throughput* sebesar 171.5/sec, *error rate* sebesar 0.00%.

4. Pengujian 4

Pada pengujian 4 ini menggunakan 1000 *user*, 30 *second* dan 5 *loops* untuk melihat performa *platform* pengujian keamanan server berbasis website ini. Dari pengujian tersebut didapatkan hasil *average respon time* selama 5584 ms, *throughput* sebesar 156.3/sec, *error rate* sebesar 0.00%.

5. Pengujian 5

Pada pengujian 3 ini menggunakan 2000 *user*, 30 *second* dan 5 *loops* untuk melihat performa *platform* pengujian keamanan server berbasis website ini. Dari pengujian tersebut didapatkan hasil berupa *average* respon time selama 12644 ms, *throughput* sebesar 146.6/sec, *error rate* sebesar 3.51%.

Berdasarkan Hasil pengujian diatas, didapatkan data tabel rekap sebagai berikut:

Tabel 5.16 Rekapitulasi Data Hasil Pengujian Frontend

users	*Avg Response Time*	*Throughput*	*Errorrate*	status
100	87 ms	120.6/sec	0.00%	sempurna
250	564 ms	187.6/sec	0.00%	bagus
500	2113 ms	171.5/sec	0.00%	layak
1000	5584 ms	156.3/sec	0.00%	Sangat lambat
2000	12644 ms	146.6/sec	3.51%	tidak layak

Tabel 5.15 Rekapitulasi Data Hasil Pengujian Frontend menunjukan bahwa hasil pengujian performa menunjukkan bahwa waktu respons aplikasi sangat optimal pada 100 pengguna dengan rata-rata 87 milidetik, tetap berada dalam batas wajar pada 250 pengguna dengan 564 milidetik, dan mulai melambat menjadi 2,1 detik pada 500 pengguna. Pada 1000 pengguna, respons mencapai 5,6 detik, sedangkan pada 2000 pengguna menjadi sangat lambat yaitu 12,6 detik, sehingga tidak layak digunakan dalam kondisi produksi. Dari sisi *throughput*, nilai tertinggi tercatat sebesar 187,6 permintaan per detik pada 250 pengguna, kemudian mengalami penurunan seiring bertambahnya jumlah pengguna, dengan tingkat efisiensi terbaik ditemukan pada rentang 250 hingga 500 pengguna. Analisis tingkat kesalahan menunjukkan tidak terdapat *error* hingga pengujian 1000 pengguna, yang menandakan stabilitas aplikasi masih sangat baik, sedangkan pada 2000 pengguna muncul *error rate* sebesar 3,51% akibat beban yang melebihi kapasitas sistem.

Tabel 5. 17 Rekapitulasi Performa Setiap *Endpoint*

endpoint	*avg response*	*error rate*	*severity*
NMAP	14162ms	3.06%	*critical*
NIKTO	14079ms	3.65%	*critical*
HISTORY	13914ms	3.36%	*critical*
HOME	13818ms	4.44%	*critical*
DASHBOARD	14045ms	4.25%	*critical*
PROFILE	13818ms	5.16%	*critical*
ABOUT	9901ms	2.31%	*warning*
CONTACT	8414ms	1.83%	*warning*

Tabel 5.16 menunjukan hasil analisis performa setiap *endpoint* pada *platform* pengujian keamanan server saat diuji dengan beban 2000 pengguna secara bersamaan. Pengujian difokuskan untuk mengidentifikasi *bottleneck* yang memengaruhi kecepatan respon dan stabilitas sistem. Hasil menunjukkan *endpoint* PROFILE dan *HOME* menjadi yang paling bermasalah dengan *error rate* di atas 5%, sedangkan ABOUT dan CONTACT paling stabil karena menampilkan konten statis. Halaman pemindaian NMAP dan NIKTO mengalami *latency* tinggi. Secara keseluruhan, aplikasi berjalan stabil tanpa *error* hingga 1000 pengguna dan menunjukkan performa sangat baik pada beban normal 100–250 pengguna.

5.2.7 Pengujian Docker

Pengujian ini dilakukan menggunakan Docker *Dekstop*. Pengujian dilakukan dengan tujuan memastikan bahwa setiap layanan *microservice* yang berjalan di dalam docker dapat bekerja dengan baik dan tidak menggunakan sumber daya server secara berlebihan. Dalam proses ini, Docker *Desktop* digunakan untuk memantau penggunaan CPU, memori, jaringan, dan aktivitas disk dari masing-masing *container* secara langsung. Skenario pengujian dimulai dengan menjalankan semua *container* menggunakan Docker *Dekstop*, sehingga seluruh layanan aktif bersamaan. Skenario pengujian dilakukan dengan menjalankan seluruh *container* secara manual melalui Docker *Desktop*, sehingga semua layanan aktif secara bersamaan. Pengujian dibagi menjadi dua kondisi, yaitu:

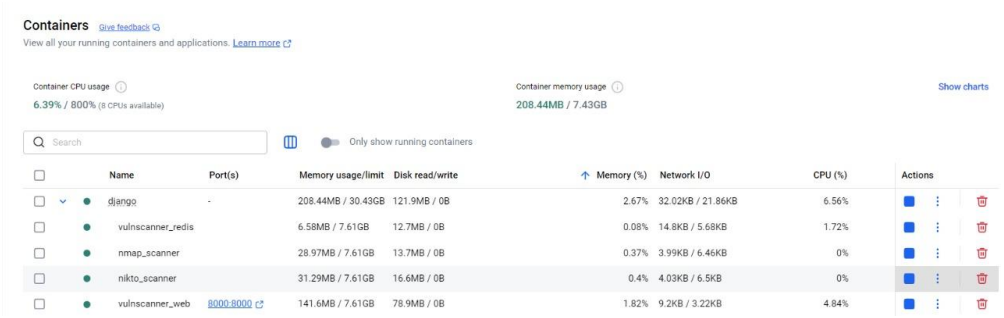
- Docker dalam keadaan aktif tanpa menjalankan proses pemindaian.
- Docker aktif dengan proses pemindaian berjalan.

Pemindaian dilakukan menggunakan tiga metode yaitu Nikto, Nmap, dan kombinasi keduanya secara bersamaan. Data yang dikumpulkan mencakup persentase penggunaan CPU, total penggunaan memori, lalu lintas data jaringan (data terkirim dan diterima), aktivitas baca dan tulis pada disk, serta jumlah proses yang berjalan pada masing-masing *container*.

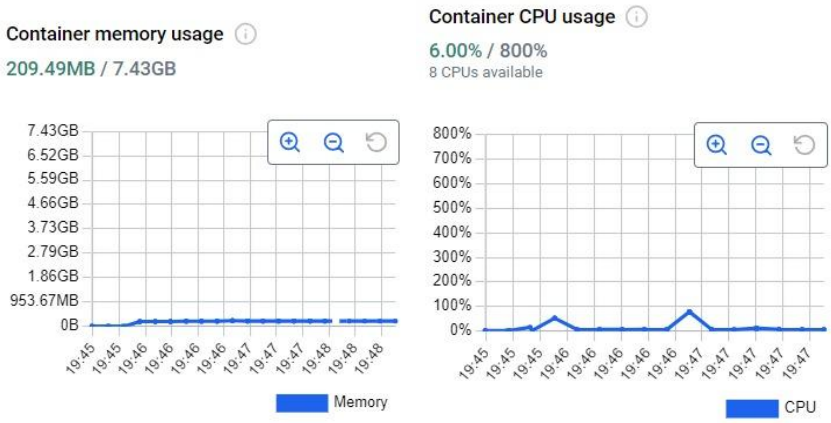
5.2.7.1 Hasil Pengujian Docker

a. Docker Aktif Tanpa Proses Pemindaian

Berikut merupakan hasil pengujian docker aktif tanpa proses pemindaian.



Gambar 5.89 Hasil Pengujian Tanpa Proses Pemindaian 1



Gambar 5.90 Grafik CPU dan Memory

Tabel 5.18 Hasil Pengujian Tanpa Proses Pemindaian 1

Container	CPU	Memory	Network		Disk	
			data sent	data received	data read	data write
vulscanner_web	0 % - 4.61 %	0 B – 114.4 MB	0 B – 3.22 KB	0 B – 9.15 KB	0 B	0 B
nikto_scanner	0%	0 B – 20.36 MB	0 B – 15.3 KB	0 B – 8.09 KB	0 B	0 B
nmap_scanner	0%	0 B – 16.64 MB	0 B – 21 KB	0 B – 10.1 KB	0 B	0 B
vulnscanner_redis	0 % - 2.01 %	0 B – 3.75 MB	0 B – 23.5 KB	0 B – 78.3 KB	0 B – 49.2 KB	0 B

Tabel 5.17 merupakan Hasil pengujian pada kondisi docker aktif tanpa proses pemindaian yang menunjukkan total penggunaan CPU sebesar 0% hingga 19,05% dan total penggunaan memory sebesar 0 B hingga 155,13 MB. Seluruh kontainer menunjukkan efisiensi penggunaan sumber daya dengan konsumsi CPU dan memory yang relatif rendah. Kontainer vulscanner_web memiliki penggunaan CPU tertinggi sebesar 4,61% dan memory sebesar 114,4 MB. Kontainer lainnya seperti nikto_scanner, nmap_scanner, dan

vulscanner_redis mencatat penggunaan CPU di bawah 2,1% dan memory di bawah 21 MB. Aktivitas network dan disk juga sangat minim. Seluruh kontainer mencatat penggunaan data sent tidak lebih dari 23,5 KB dan data received tidak melebihi 78,3 KB. Hanya kontainer vulscanner_redis yang mencatat aktivitas data read pada disk sebesar 49,2 KB, sementara kontainer lainnya tidak menunjukkan aktivitas read maupun write pada disk.

b. Docker Aktif dengan Proses Pemindaian

Pada bagian ini terdapat 3 jenis pemindaian, yaitu pemindaian menggunakan Nikto, pemindaian menggunakan Nmap dan pemindaian menggunakan Nikto dan Nmap secara bersamaan.

1. Nikto

Tabel 5.19 Hasil 1 (Pengujian Docker Aktif dengan Pemindaian Menggunakan Nikto)

Container	CPU	Memory	Network		Disk	
			data sent	data received	data read	data write
vulscanner_web	4.79 % - 63.76 %	122.6 MB – 130.7 MB	276 KB – 510 KB	161 KB – 596 KB	8.65 MB – 13.5 MB	0 B
nikto_scanner	43.61 % - 67.26 %	16.8 MB – 16.81 MB	39.6 KB – 70.4 KB	16.6 KB – 26.9 KB	0 B	0 B
nmap_scanner	0%	16.64 MB – 16.72 MB	39.3 KB – 70.1 KB	16.5 KB – 31 KB	0 B	0 B
vulnscanner_redis	1.4 % - 13.01 %	3.48 MB – 3.73 MB	35 KB – 65.8 KB	106 KB – 196 KB	81.9 KB	4.1 KB

Tabel 5.18 Hasil pengujian pada kondisi docker aktif dengan proses pemindaian menggunakan Nikto menunjukkan total penggunaan CPU berada pada rentang 6,23% hingga 125,40%, dan total penggunaan memory sebesar 167,63 MB hingga 176,8 MB. Kontainer dengan penggunaan CPU tertinggi adalah nikto_scanner dengan nilai mencapai 67,26%, diikuti oleh vulscanner_web sebesar 63,76%. Sementara itu, kontainer nmap_scanner tidak mencatat aktivitas CPU, dan vulscanner_redis menggunakan CPU hingga 13,01%. Dari sisi *memory*, kontainer vulscanner_web menunjukkan penggunaan tertinggi, yaitu antara 122,6 MB hingga 130,7 MB, sedangkan kontainer lain berada di bawah 17 MB.

Aktivitas network tercatat cukup signifikan pada kontainer vulscanner_web dan vulscanner_redis. Data sent tertinggi dicatat oleh vulscanner_web sebesar 510 KB, dan data received

mencapai 596 KB. Kontainer vulscanner_redis juga menunjukkan aktivitas network yang cukup tinggi dengan data received hingga 196 KB. Pada aktivitas disk, data read tertinggi tercatat pada vulscanner_web sebesar 13,5 MB, dan vulscanner_redis sebesar 81,9 KB. Satu-satunya aktivitas data write tercatat pada kontainer vulscanner_redis sebesar 4,1 KB, sedangkan kontainer lainnya tidak mencatat aktivitas write pada disk.

2. Nmap

Tabel 5.20 Hasil 1 (Pengujian Docker aktif dengan Pemindaian Menggunakan Nmap)

Container	CPU	Memory	Network		Disk	
			data sent	data received	data read	data write
vulscanner_web	5.93 % - 16.72 %	133.9 MB – 152.3 MB	513 KB – 990 KB	599 KB – 782 KB	13.5 MB – 19.3 MB	0 B
nikto_scanner	0%	16.81 MB	89.4 KB – 106 KB	34.6 KB – 44.7 KB	0 B	0 B
nmap_scanner	18.08 % - 94.02 %	16.64 MB – 16.8 MB	87 KB – 124 KB	35 KB – 47.5 KB	0 B	0 B
vulnscanner_redis	1.41 % - 12.6 %	3.48 MB – 3.77 MB	70.2 KB – 104 KB	203 KB – 314 KB	8.19 KB – 12.3 KB	81.9 KB

Tabel 5.19 Hasil pengujian pada kondisi docker aktif dengan proses pemindaian menggunakan Nmap menunjukkan total penggunaan CPU berada pada rentang 3,76% hingga 129,79%, dan total penggunaan memory sebesar 37,1 MB hingga 189,42 MB. Kontainer dengan penggunaan CPU tertinggi adalah nmap_scanner yang mencapai 94,02%, diikuti oleh vulscanner_web sebesar 16,72%. Kontainer nikto_scanner tidak mencatat aktivitas CPU, dan vulscanner_redis menggunakan CPU hingga 12,6%. Dari sisi memory, vulscanner_web menjadi kontainer dengan penggunaan tertinggi yaitu hingga 152,3 MB, sementara kontainer lainnya tercatat menggunakan memory di bawah 17 MB.

Aktivitas network juga meningkat terutama pada vulscanner_web dan vulscanner_redis. Data sent tertinggi dicatat oleh vulscanner_web sebesar 990 KB dan *data received* mencapai 782 KB. Kontainer vulscanner_redis juga mencatat

data received yang cukup tinggi hingga 314 KB. Untuk aktivitas disk, vulscanner_web mencatat data read tertinggi sebesar 19,3 MB, disusul oleh vulscanner_redis dengan *data read* sebesar 12,3 KB dan satu-satunya aktivitas *data write* sebesar 81,9 KB. Kontainer lainnya tidak mencatat aktivitas read maupun write pada disk.

3. Nikto dan Nmap Secara Bersamaan

Tabel 5.21 Hasil 2 (Pengujian Docker Aktif Dengan Pemindaian Menggunakan Nmap Dan Nikto Bersamaan)

Container	CPU	Memory	Network		Disk	
			data sent	data received	data read	data write
vulscanner_web	12.41 % - 93.74 %	144.7 MB – 162.5 MB	1.02 MB – 1.78 MB	141 KB – 1.71 MB	5.58 MB	28.7 KB
nikto_scanner	46.24 % - 139.52 %	16.78 MB – 31.89 MB	10.1 KB – 48.9 KB	4.93 KB – 18 KB	0 B	0 B
nmap_scanner	25.41 % - 68.22 %	16.77 MB – 16.79 MB	9.9 KB – 51.6 KB	4.83 KB – 20.8 KB	0 B	0 B
vulnscanner_redis	1.72 % - 16.5 %	3.75 MB – 4.02 MB	36.2 KB – 125 KB	52.3 KB – 155 KB	0 B	0 B – 4.1 KB

Tabel 5.20 Hasil pengujian pada kondisi docker aktif dengan proses pemindaian secara bersamaan menggunakan Nikto dan Nmap menunjukkan total penggunaan CPU berada pada rentang 6,25% hingga 306,79%, serta total penggunaan memory sebesar 183,53 MB hingga 208,68 MB. Kontainer dengan penggunaan CPU tertinggi adalah nikto_scanner sebesar 139,52%, diikuti oleh vulscanner_web sebesar 93,74% dan nmap_scanner sebesar 68,22%. Kontainer vulscanner_redis mencatat penggunaan CPU paling rendah sebesar 16,5%. Dari sisi memory, vulscanner_web memiliki penggunaan tertinggi hingga 162,5 MB, sementara kontainer lainnya menggunakan memory di bawah 32 MB. Aktivitas network juga cukup tinggi pada pengujian ini. Data sent tertinggi tercatat pada kontainer vulscanner_web sebesar 1,78 MB, disusul oleh nikto_scanner dan nmap_scanner masing-masing sebesar 48,9 KB dan 51,6 KB.

Data received tertinggi juga berasal dari vulscanner_web dengan nilai hingga 1,71 MB, serta vulscanner_redis sebesar 155 KB. Untuk aktivitas disk, vulscanner_web mencatat data read sebesar 5,58 MB dan data write sebesar 28,7 KB. Selain itu, vulscanner_redis menjadi satu-satunya kontainer lain yang mencatat aktivitas write pada disk sebesar 4,1 KB. Kontainer lainnya tidak menunjukkan aktivitas read maupun write pada disk.

5.2.8 Rangkuman Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan yaitu pengujian keamanan menggunakan 2 *tools* dengan masing – masing *tools* memiliki 3 kategori *website* yaitu localhost 1 *device*, localhost dari *device* lain, dan *website* dari internet. Hasil pengujian menunjukan berbagai jenis kategori kerentanan pada setiap *website*. Hasil pemindaian keamanan menunjukan hasil *reporting* yang konsisten setelah dilakukan pengujian selama 3 kali. *Platform* ini juga memberikan rekomendasi perbaikan berbasis *generative* AI kepada *user* sesuai dengan hasil *reporting* yang sudah diberikan sebelumnya. Pengujian performa *platform* ini menggunakan ApacheJmeter juga menunjukan bahwa *platform* ini memiliki tingkat skalabilitas dan reliabilitas yang bagus, terbukti dari *error rate* sebesar 0% dan tidak terjadi *crash* pada seluruh skenario pengujian. *Platform* juga menunjukkan *scalability* yang baik, dengan *throughput* meningkat seiring kenaikan jumlah *request*. Namun, pada skenario beban sangat tinggi, waktu respon mengalami kenaikan signifikan sehingga kinerja menjadi lebih lambat. Secara keseluruhan, *platform* pengujian keamanan server berbasis website ini berhsail melakukan *scanning*, *reporting* dan rekomendasi berbasis AI yang dapat diunduh oleh pengguna dalam format PDF.

Tabel 5. 22 Perbandingan Spesifikasi dan Realisasi

No	Spesifikasi	Realisasi	Tercapai/ Tidak Tercapai
1	*Scanning*	*Platfom* pengujian keamanan ini dapat melakukan *scanning* keamanan dengan menggunakan dua *tools* utama yaitu Nikto dan Nmap yang diintegrasikan dalam sebuah *Website* dengan fitur *scanning* diantaranya *OS Detection*, *Port scan*, *Vurnerability*, *comperhensive*, *Basic scan*, *Full scan*, SSL/TSL *scan*, dan *custom scan* (menyesuaikan parameter lanjutan).	Tercapai
2	*Reporting*	*Platform* ini dapat menyediakan laporan hasil temuan kerentanan pada server berupa ringkasan pemindaian, skor keamanan, *chart*, analisis keamanan detail dengan kategori tingkat kerentanan *Critical*, *High*, *Medium*, *Low*, Info. dan menyediakan pelaporan data hasil *security* scan.	Tercapai
3	Rekomendasi perbaikan	*Platform* menyediakan rekomendasi perbaikan berbasis *generative AI* berdasarkan hasil temuan kerentanan dan dapat diunduh oleh *user* dalam format PDF.	Tercapai

BAB 6
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, pengembangan *platform* pengujian keamanan server berbasis *website* ini terbukti mampu menjadi solusi yang efektif bagi kompleksitas masalah dalam keamanan server yaitu serangan *cyber* seperti pencurian data, peretasan, serangan kerentanan terhadap *malware* dan berbagai serangan berkelanjutan lainnya. Analisis pengujian menggunakan dua tools utama dengan masing-masing memiliki 3 skenario pengujian membuktikan bahwa *platform* pengujian keamanan server berbasis *website* ini mampu melakukan *scanning* keamanan server dengan berbagai macam metode yang tersedia pada *platform* sesuai kebutuhan pengguna, *platform* ini juga berhasil menyediakan hasil temuan kerentanan secara detail.

Dari hasil rekapitulasi pengujian, *platform* diuji pada berbagai skenario, yaitu pada lingkungan *Localhost* 1 *device*, *Localhost* beda *device*, dan internet. Untuk kategori *Localhost* 1 *device*, durasi rata-rata pemindaian menggunakan nikto berkisar antara 0.99 hingga 1.77 detik, sedangkan nmap menunjukkan durasi rata-rata 174.10 hingga 512.23 detik, dengan hasil konsistensi temuan yang tetap terjaga. Pada kategori *Localhost* beda *device*, durasi rata-rata pemindaian nikto meningkat antara 4.51 hingga 10.12 detik, sementara nmap mencapai 439.81 hingga 989.25 detik dengan hasil yang konsisten. Sementara itu, pengujian terhadap *website* berbasis internet seperti www.pln.co.id, httpbin.org, dan www.komdigi.go.id menunjukkan waktu pemindaian nikto antara 20.18 hingga 82.76 detik, dan durasi nmap antara 358.18 hingga 847.14 detik, yang semuanya tetap menunjukkan konsistensi dalam hasil temuan kerentanan.

Platform ini juga telah diuji *reliability*-nya menggunakan ApacheJMeter yang membuktikan bahwa *platform* "Pengujian Keamanan Server Berbasis *Website*" ini memiliki *reliability* yang baik karena ketika diuji dengan *request* terbanyak sebesar 48.000, *Website* mampu merespons semua permintaan dengan sukses, ditunjukkan oleh tingkat *error* 0% dan tidak adanya *crash*, sehingga *website* berjalan stabil. Meskipun pada skenario beban sangat tinggi, waktu respon mengalami kenaikan signifikan sehingga kinerja *platform* ini menjadi lebih lambat.

Pengujian performa *backend* menunjukkan bahwa penggunaan CPU dan memori meningkat secara bertahap pada saat proses *scanning* berlangsung. Penggunaan CPU rata-rata meningkat dari 7.44% pada kondisi normal menjadi 9.22% saat menggunakan nikto dan 9.84% saat menggunakan nmap. Begitu pula penggunaan memori, yang naik dari 1.50% dalam

kondisi normal menjadi 1.62% saat nikto aktif, dan 1.76% saat nmap berjalan. Hal ini menandakan adanya beban komputasi yang wajar dan proporsional selama proses *scanning* berlangsung.

Sementara itu, hasil pengujian performa *frontend* menunjukkan bahwa sistem masih memberikan respon yang baik hingga 500 pengguna, namun mulai mengalami perlambatan di atas 1000 pengguna, dengan *response time* mencapai 12.644 ms dan *error rate* sebesar 3.51% pada 2000 pengguna. Selain itu, performa tiap *endpoint* menunjukkan beberapa *endpoint* kritikal seperti NMAP, NIKTO, dan DASHBOARD memiliki rata-rata waktu respons di atas 13 detik dengan tingkat *error* di atas 3%, yang menandakan perlunya optimisasi lebih lanjut untuk kestabilan di skenario beban tinggi.

Selain itu, pengujian performa *container* menggunakan Docker Desktop menunjukkan bahwa penggunaan CPU dan memori meningkat sesuai dengan intensitas proses *scanning*. Dari seluruh pengujian, penggunaan CPU total tertinggi tercatat sebesar 306,79% saat dilakukan pemindaian gabungan menggunakan nikto dan nmap, sedangkan penggunaan CPU total terendah terjadi pada kondisi docker aktif tanpa proses pemindaian, yaitu sebesar 19,05%. Untuk penggunaan *memory* total, nilai tertinggi dicapai pada pengujian gabungan tersebut sebesar 208,68 MB, sedangkan nilai terendah tercatat sebesar 155,13 MB pada kondisi tanpa pemindaian. Aktivitas jaringan dan *disk* juga meningkat signifikan terutama saat proses pemindaian dijalankan secara bersamaan dengan *tools* nikto dan nmap, yang menunjukkan bahwa beban sistem terdistribusi sesuai fungsi masing-masing *container*.

Dari hasil pengujian keamanan yang diperoleh, *platform* ini menunjukan hasil temuan kerentanan yang konsisten pada setiap target yang diuji dan semua data *reporting* kerentanan diperoleh secara *real-time* tidak ada data *dummy* (data tiruan atau data palsu) dengan waktu pengujian yang cukup singkat. *Platform* ini juga memberikan rekomendasi perbaikan berbasis *generative AI* yang relevan dengan hasil temuan kerentanan dan dapat diunduh oleh pengguna dalam format PDF sehingga memberikan gambaran langkah yang harus dilakukan ke depannya oleh pengguna.

Namun, meskipun pengembangan *platform* pengujian keamanan ini efektif untuk menjadi solusi bagi kompleksitas masalah serangan *cyber*, dalam penerapannya masih perlu dilakukan lagi pengembangan secara berkala seperti tambahan integrasi *tools* pengembangan *open source* yang lebih beragam agar dapat mendeteksi keseluruhan OWASP Top 10 daftar risiko keamanan aplikasi web yang paling kritis.

6.2 Saran

Berdasarkan pengujian *platform* “Pengujian Keamanan Server Berbasis *Website*” ini diperlukan beberapa saran untuk pengembangan lebih lanjut. Berikut merupakan saran yang diperlukan untuk pengembangan *platform* “Pengujian Keamanan Server Berbasis *Website*” ini :

1. Tambahan integrasi *tools* pengujian keamanan yang bersifat *open source* yang lebih beragam agar dapat mendeteksi keseluruhan OWASP Top 10 daftar risiko keamanan aplikasi *web* yang paling kritis.
2. Memperbanyak pengujian keamanan terhadap *website* yang ada di internet. Hal ini dikarenakan setiap *website* memiliki deskripsi kerentanan yang berbeda-beda. *Platform* pengujian keamanan ini memerlukan proses pendefinisian secara spesifik terhadap deskripsi setiap kerentanan tersebut agar hasil pemindaian menjadi lebih akurat dan relevan. Hal ini terjadi karena setiap *website* memiliki karakteristik dan deskripsi kerentanan yang beragam. Oleh karena itu, *platform* pengujian keamanan ini perlu melakukan pendefinisian secara spesifik terhadap tiap kerentanan tersebut, sehingga hasil pemindaian dapat menjadi lebih akurat
3. Mengembangkan fitur admin dan memperkecil kemungkinan adanya bug.

Dari saran-saran penulis di atas, diharapkan Tugas Akhir Capstone Design *platform* ”Pengujian Keamanan Server Berbasis *Website*” ini dapat memberikan manfaat yang baik untuk pengembangan selanjutnya.

DAFTAR PUSTAKA

1. Alqausar, D. (2024, Desember 27). *proxsisgroup*. Retrieved from biztech.proxsisgroup.com: https://infrasec.proxsisgroup.com/membandingkan-metasploit-nessus-dan-acunetix-untuk-kebutuhan-penetration-testing-perusahaan-anda/#%3A~%3Atext%3DNessus-%2CNessus%20adalah%20alat%20pemindaian%20keamanan%20yang%20digunakan%20untuk%20mengidentifikasi%20kere
2. Ari Marta Tania, Didik Setiyadi, & Fata Nidaul Khasanah. (2018). Keamanan *Website* Menggunakan Vulnerability Assessment. *INFORMATIC FOR EDUCATORS AND PROFESSIONALS*, 171-180.
3. Bai, J. (2021). A Study on Software Vulnerability Prioritization and Management Methods. *Bai, J. (2021). A Study on Software Vulnerability PrioritizInternational Conference on Computer, Information and Telecommunication Systems (CITS)*, 1-5.
4. *Bizplus*. (2025). Retrieved from bizplus.id: https://bizplus.id/pengertian-dan-metode-*security*-testi/.
5. Chongqing kang, Daniel Kirschen, & Timothy C.Green. (2023). The Evolution of Smart Grids. *Proceeding of the IEEE*, 691-693.
6. Cloudeka. (2023). *8 Tahapan Penetration Testing yang Penting dan Akurat.* jakarta: Lintasarta Cloudeka.
7. *CLOUDFLARE*. (2025). Retrieved from cloudflare.com: https://www.cloudflare.com/learning/*security*/threats/owasp-top-10/
8. Dasci, M. (2024). Exploring Nmap Tool: A Comprehensive Analysis. 50.
9. Django. (2025, 6 18). *Django.* Retrieved from docs.djangoproject.com: https://docs.djangoproject.com/en/5.2/
10. *Docker.* (2025). Retrieved from docs.docker.com: Diakses pada 7 juli 2025 https://docs.docker.com
11. Esra Abdullatif Altulaihan, Abrar Alismail, & Mounir Frikha. (4 march 2023). A Survey on *Web* Application Penetration Testing. *electronics*, 12.
12. F.Fachri. (2023). "OPTIMASI KEAMANAN *WEB* SERVER TERHADAP SERANGAN . 51-58.

13. Foundation, D. S. (2025, Juli 7). *Django.* Retrieved from docs.djangoproject.com: https://docs.djangoproject.com/en/stable/

14. Hogy Albani Bardian, & Imam Sutanto. (2025). PENGEMBANGAN APLIKASI VULNERABILITY *SCANNER* UNTUK MENDETEKSI CELAH KEAMANAN SIBER PADA *WEBSITE*. *Jurnal Mahasiswa Teknik Informatika*, 4044-4411.

15. Hubli, S. C., & Jaiswal, R. C. (2023). Efficient *Backend* Development with Spring Boot: A Comprehensive Overview. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 11.

16. Kane, S. P., & Matthias, K. (2022). Docker: Up & Running (3rd ed.). *O'Reilly Media*.

17. Lyon, G. (2024). *NMAP.ORG*. Retrieved from nmap.org: nmap.org

18. Ridho, M. R. (2024, march 18). *Telkomuniversity.* Retrieved from Telkomuniversity.ac.id: https://bee.telkomuniversity.ac.id/panduan-lengkap-macam-macam-jenis-server- serta-fungsinya/

19. Ruli Dimas Prakoso, & Asmunin. (2018). IMPLEMENTASI DAN PERBANDINGAN PERFORMA PROXMOX DALAM VIRTUALISASI . *JURNAL MANAJEMEN INFORMATIKA*, 79 - 85.

20. Schwartzburg, D. (2023). *FIRST.* Retrieved from first,org: https://www.first.org/cvss/v4-0/specification-document

21. Sofyan Mufti Prasetiyo, Muhammad Ivan Prayogi Nugroho, Riris Lima Putri, & Opa Fauzi. (2022). Pembahasan Mengenai Front-end *Web* Developer dalam Ruang Lingkup *Web* Development. *Jurnal Multidisiplin Ilmu*, 1015-1020.

22. SULIMAN ALAZMI, & DANIEL CONTE DE LEON. (2022). A Systematic Literature Review on the Caharacteristics and Effectiveness of *Web* Application Vulnerability *Scanner*s. *Digital Object Identifier*, 33200-33219.

23. Sullo, C. (2024). *CIRT.net*. Retrieved from cirt.net: https://www.cirt.net/sullo

24. VIncent, W. (2022). Django for Professionals: Production *Website*s with Python & Django. *Django Riffs*.

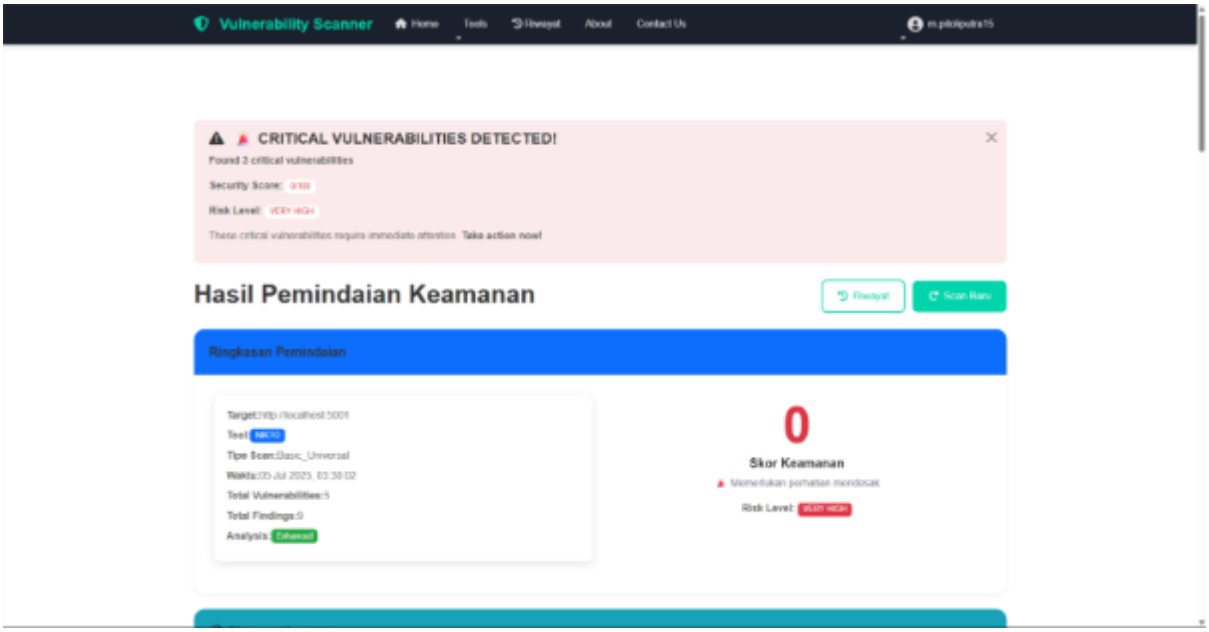
25. Yusuf Muhyidin, M.Hafid Totohendarto, Erina Undayamayanti, & Salsabilla C. N. (2021). Perbandingan Tingkat Keamanan *Website* Menggunakan Nmap Dan Nikto Dengan Metode Ethical Hacking. *Jurnal Teknologika*, 1-10.

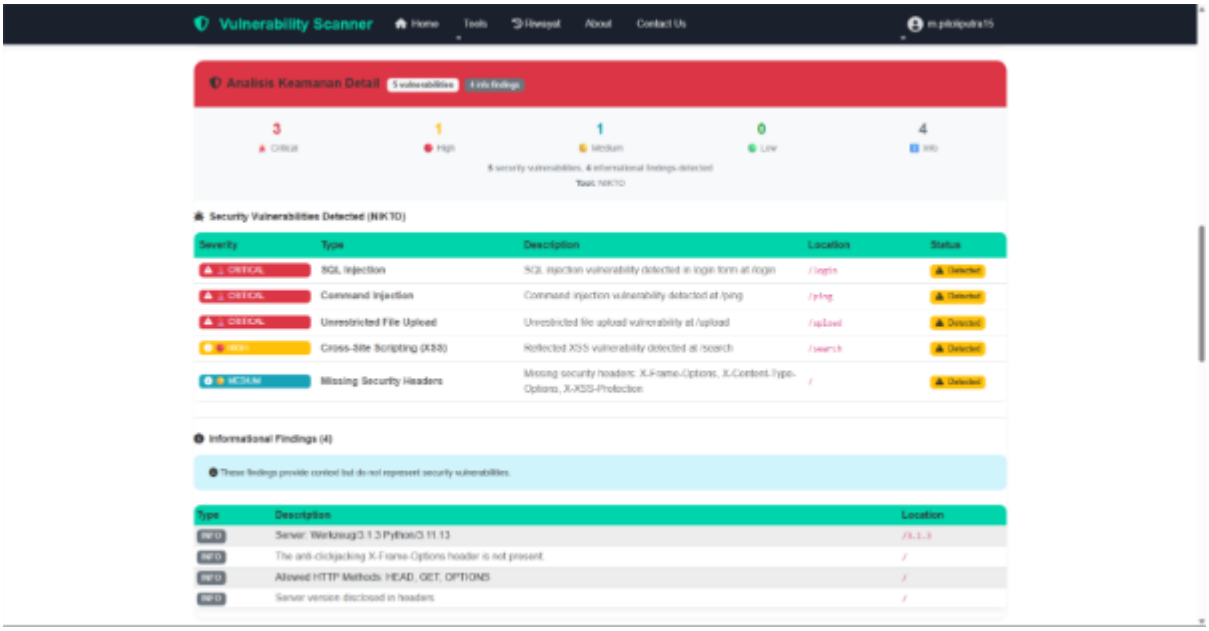
LAMPIRAN

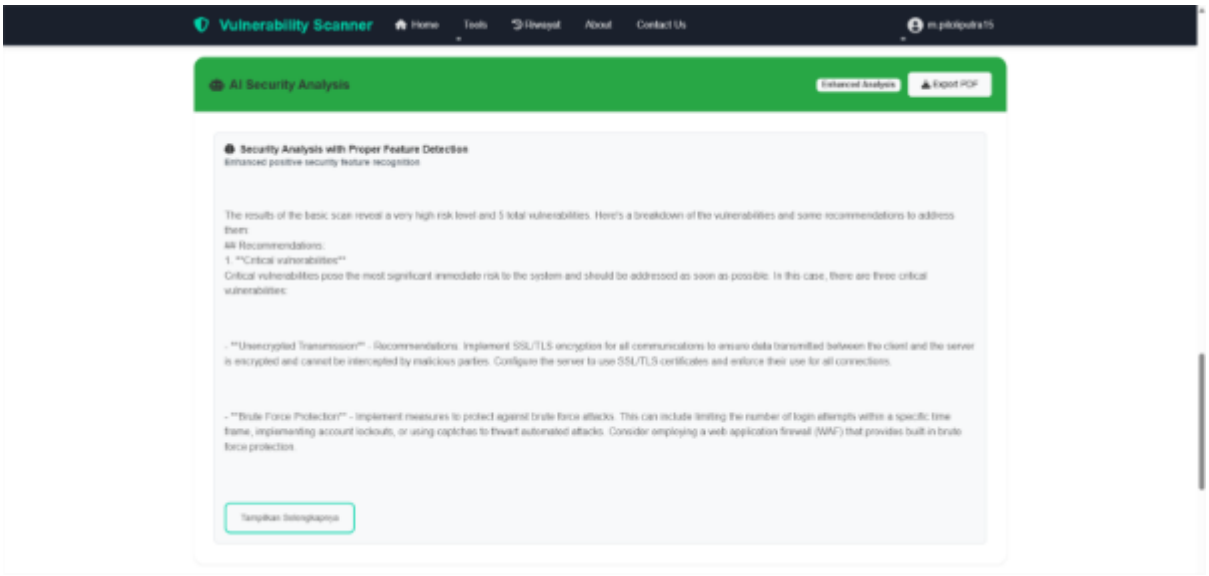
- Link repository Github : https://github.com/ronald212121/django_final.git

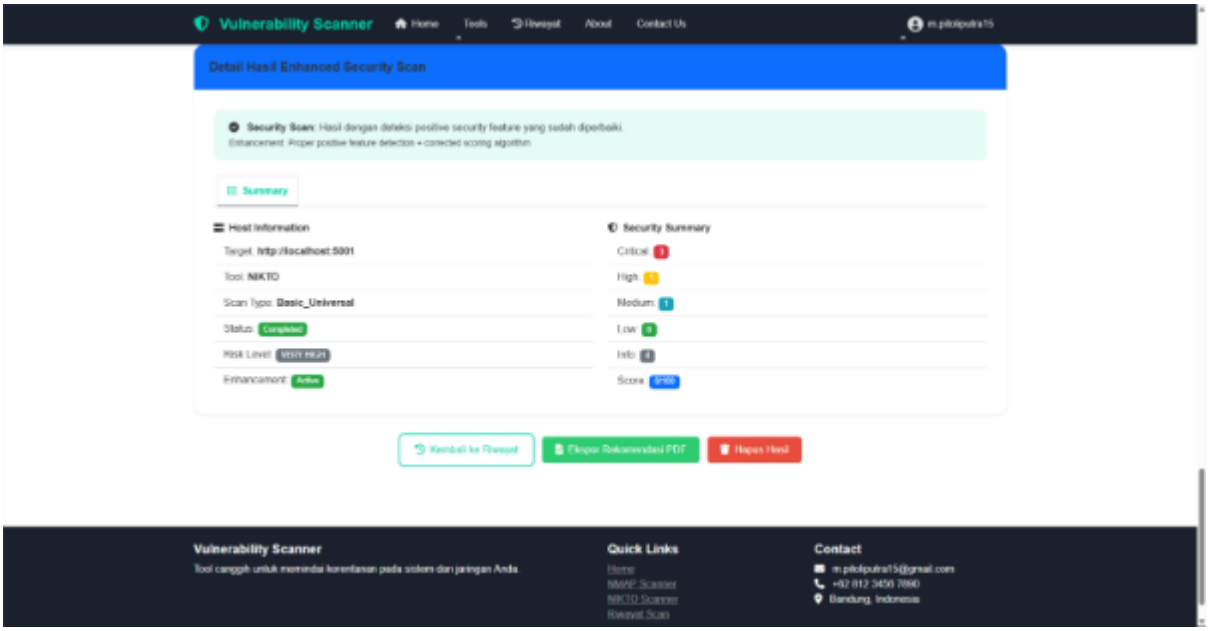
a. Pengujian (*tools* nikto) *Website* localhost 1 *device*

- *Website* 1 (localhost:5001)

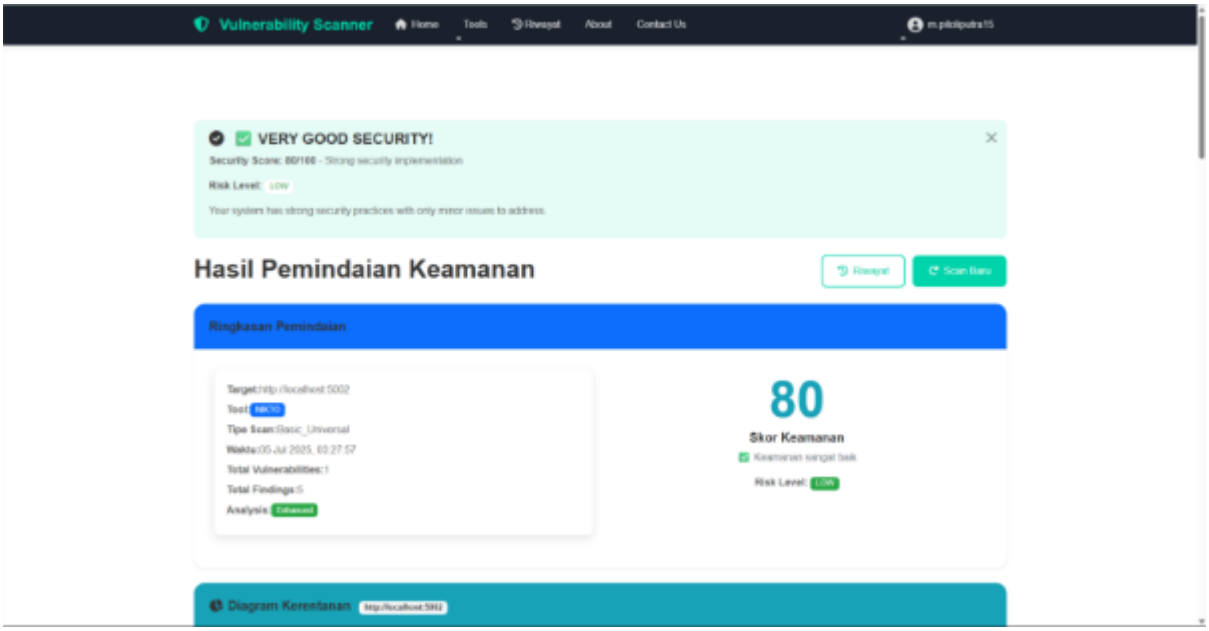


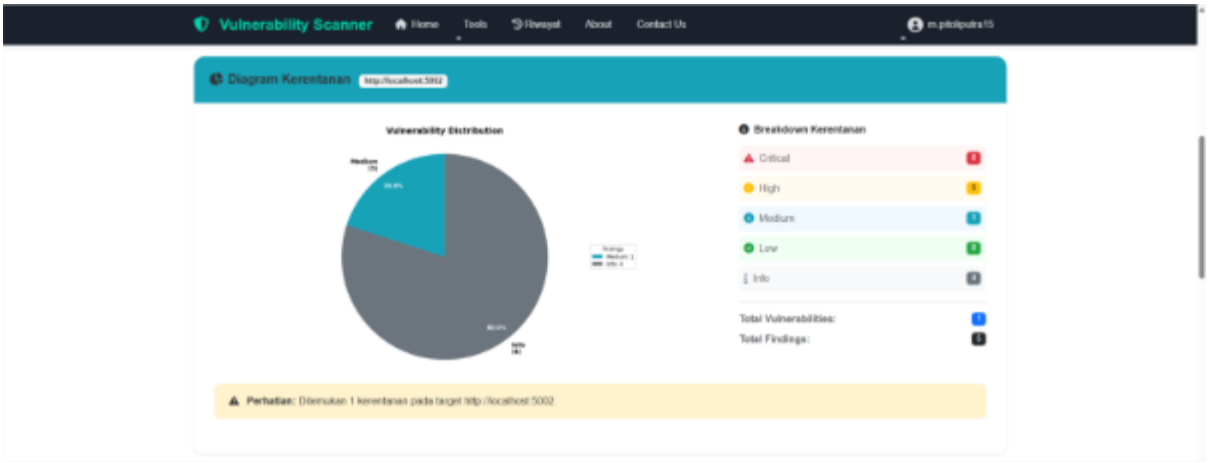

Vulnerability Scanner
Home
Tools
About
Contact Us
Analisis Keamanan Detail
3
1
1
0
4
Security Vulnerabilities Detected (NIKTO)
Severity
Type
Description
Location
Status
SQL Injection
Command Injection
Unrestricted File Upload
Cross-Site Scripting (XSS)
Missing Security Headers
Informational Findings (4)
Type
Description
Location

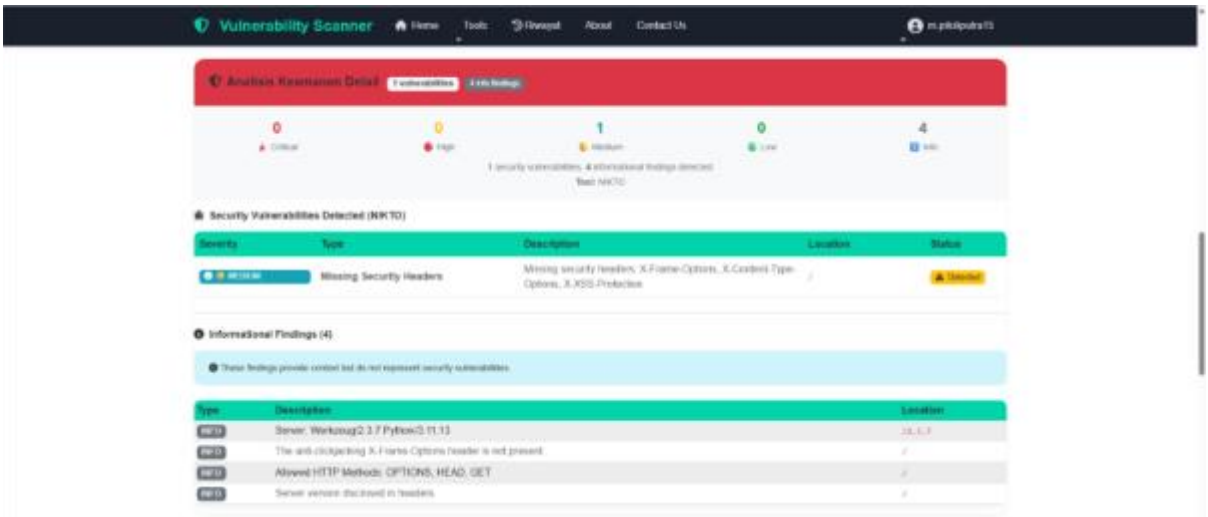

Vulnerability Scanner
Home
Tools
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

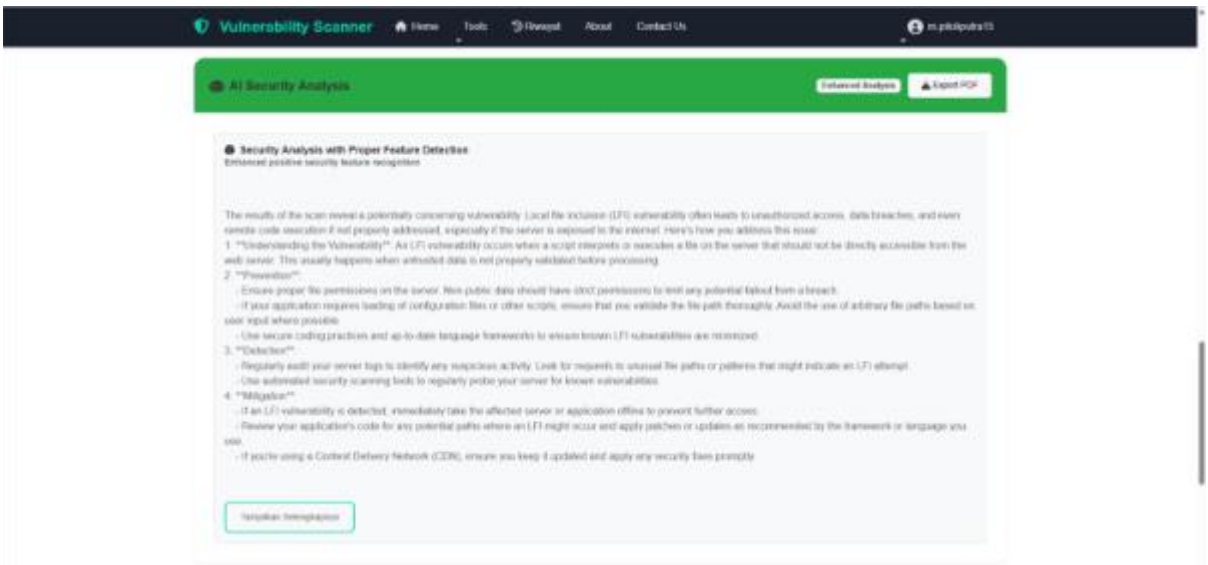


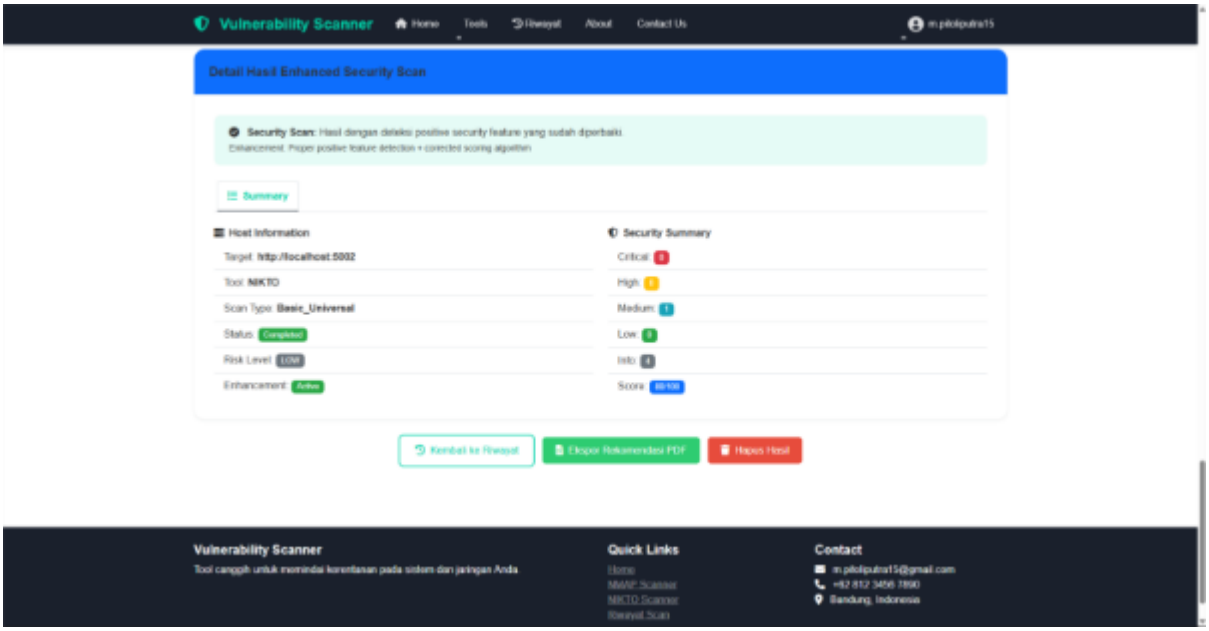
- *Website* 2 (localhost:5002)



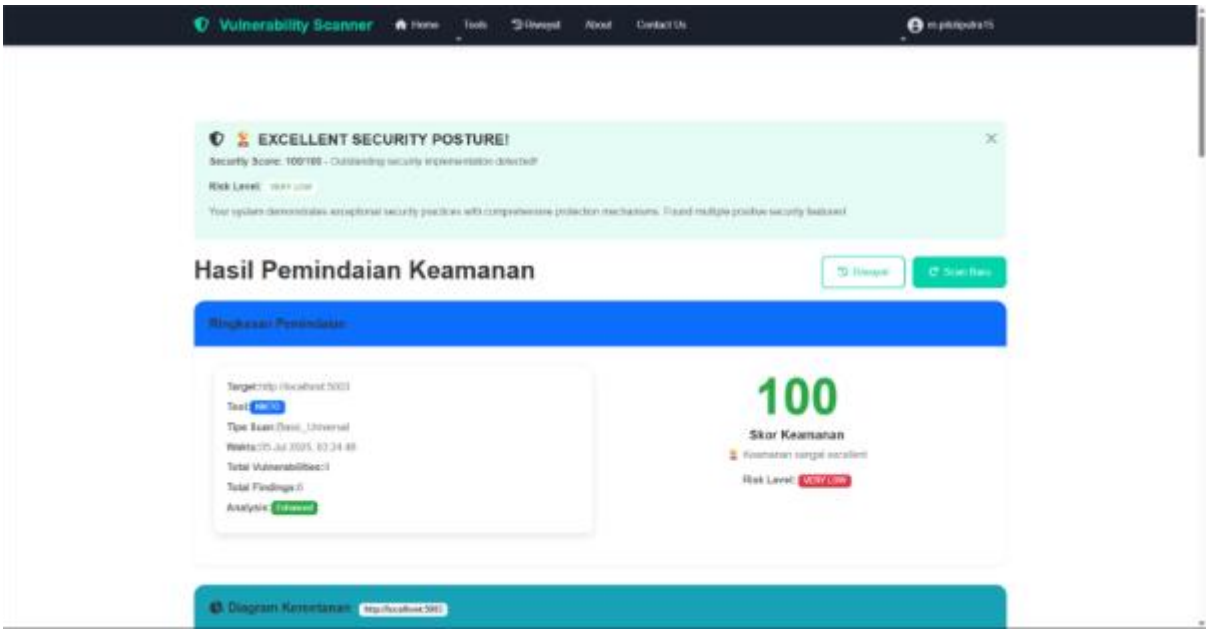

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Diagram Kerentanan
Vulnerability Distribution
Breakdown Kerentanan
Critical
High
Medium
Low
Info
Total Vulnerabilities:
Total Findings:
Perhatian: Ditemukan 1 kerentanan pada target http://localhost:5002

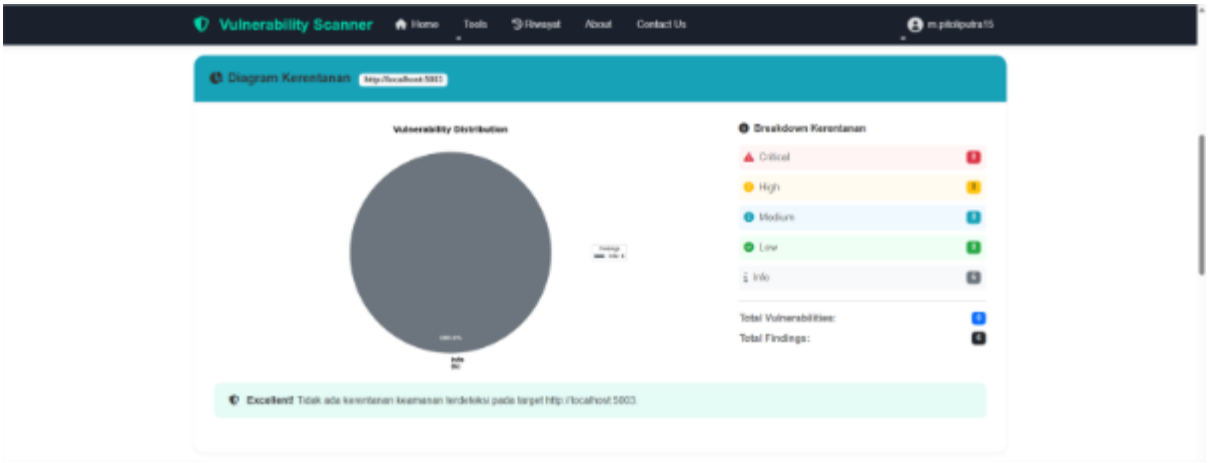

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Analisis Kerentanan Detail
0
0
1
0
4
Security Vulnerabilities Detected (NIKTO)
Severity
Type
Description
Location
Status
Missing Security Headers
Informational Findings (4)
Type
Description
Location
Server: Werkzeug/2.3.7 Python/3.11.13
Allowed HTTP Methods: OPTIONS, HEAD, GET

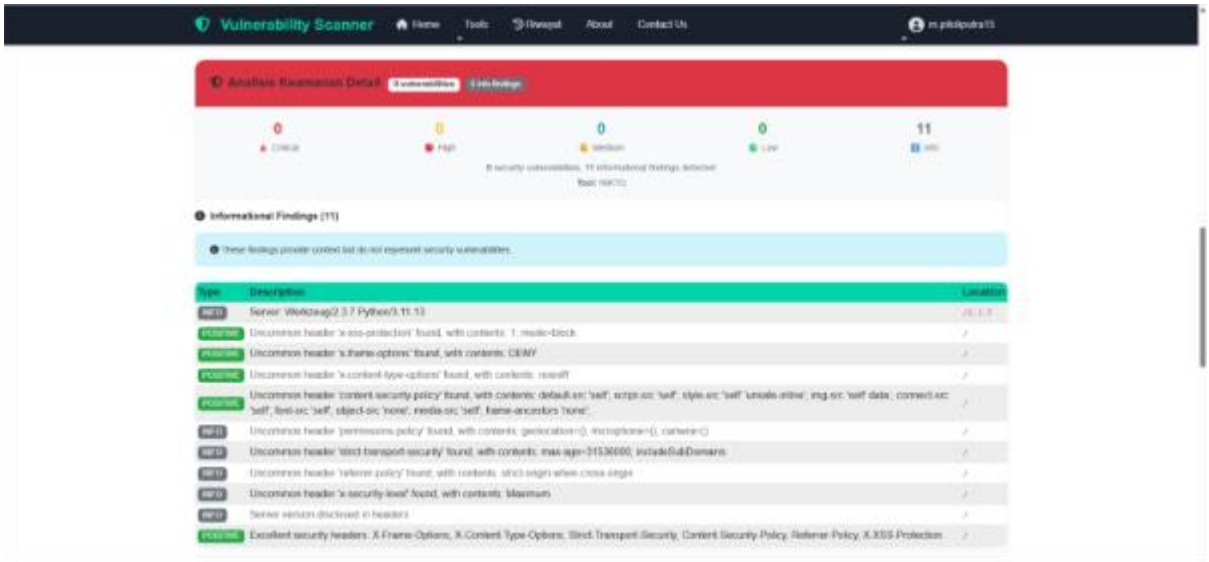

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

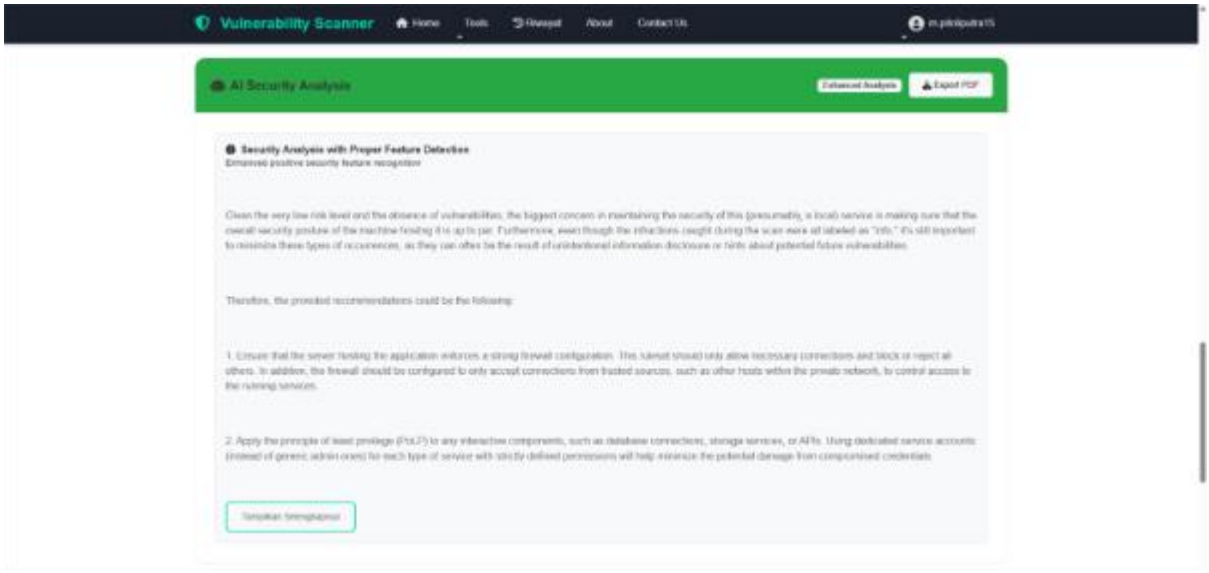


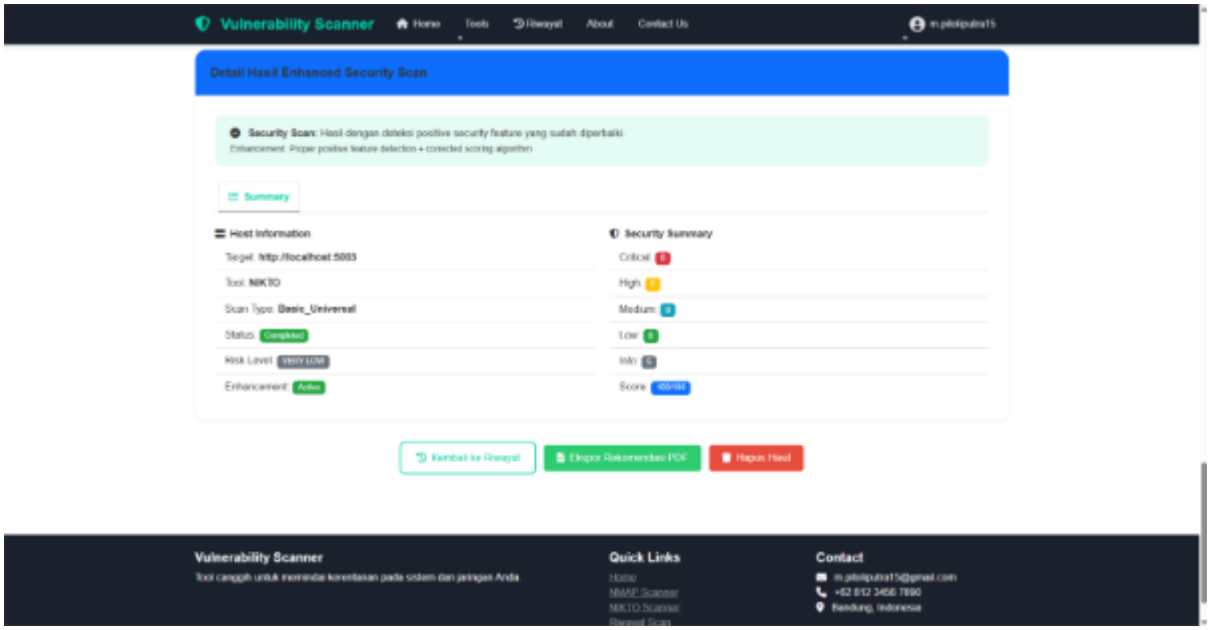
- *Website* 3 (localhost:5003)




Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Diagram Kerentanan
Vulnerability Distribution
Breakdown Kerentanan
Critical
High
Medium
Low
Info
Total Vulnerabilities:
Total Findings:
Excellent! Tidak ada kerentanan keamanan terdeteksi pada target http://localhost:5003.

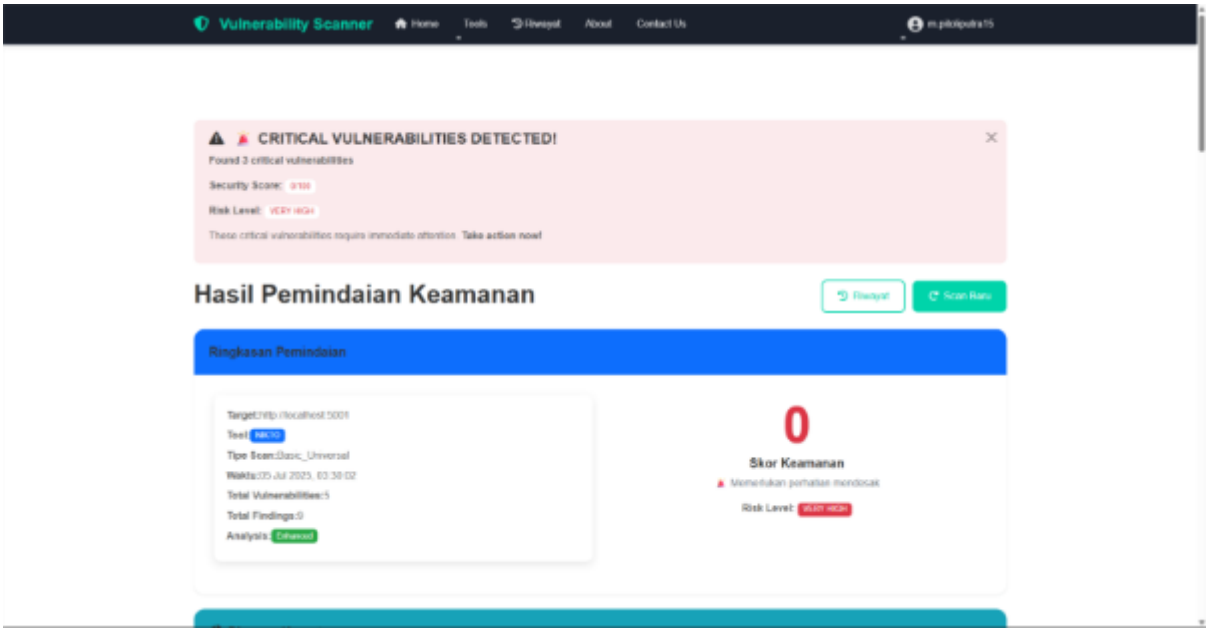

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
0
0
0
0
11
Informational Findings (11)
These findings provide context but do not represent security vulnerabilities.
Type
Description
Location
INFO
Server: Werkzeug/2.3.7 Python/3.11.13
POSITIVE
Uncommon header 'x-xss-protection' found, with contents: 1; mode=block
POSITIVE
Uncommon header 'x-frame-options' found, with contents: DENY
POSITIVE
Uncommon header 'x-content-type-options' found, with contents: nosniff
POSITIVE
Uncommon header 'content-security-policy' found, with contents: default-src 'self'; script-src 'self'; style-src 'self' 'unsafe-inline'; img-src 'self' data:; connect-src 'self'; font-src 'self'; object-src 'none'; media-src 'self'; frame-ancestors 'none';
INFO
Uncommon header 'permissions-policy' found, with contents: geolocation=(), microphone=(), camera=()
INFO
Uncommon header 'strict-transport-security' found, with contents: max-age=31536000; includeSubDomains
INFO
Uncommon header 'referrer-policy' found, with contents: strict-origin-when-cross-origin
INFO
Uncommon header 'x-security-level' found, with contents: Maximum
INFO
Server version disclosed in headers
POSITIVE
Excellent security headers: X-Frame-Options, X-Content-Type-Options, Strict-Transport-Security, Content-Security-Policy, Referrer-Policy, X-XSS-Protection

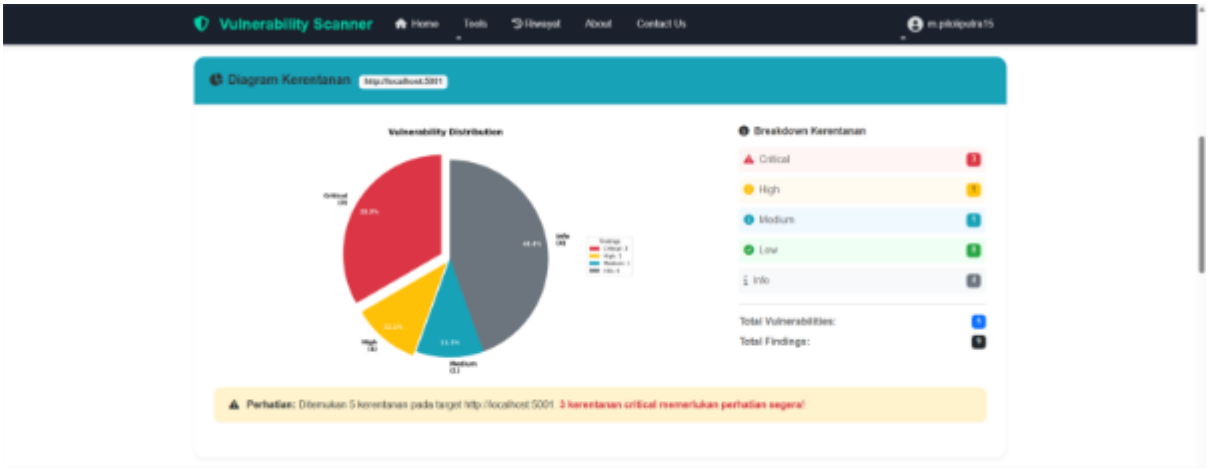

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

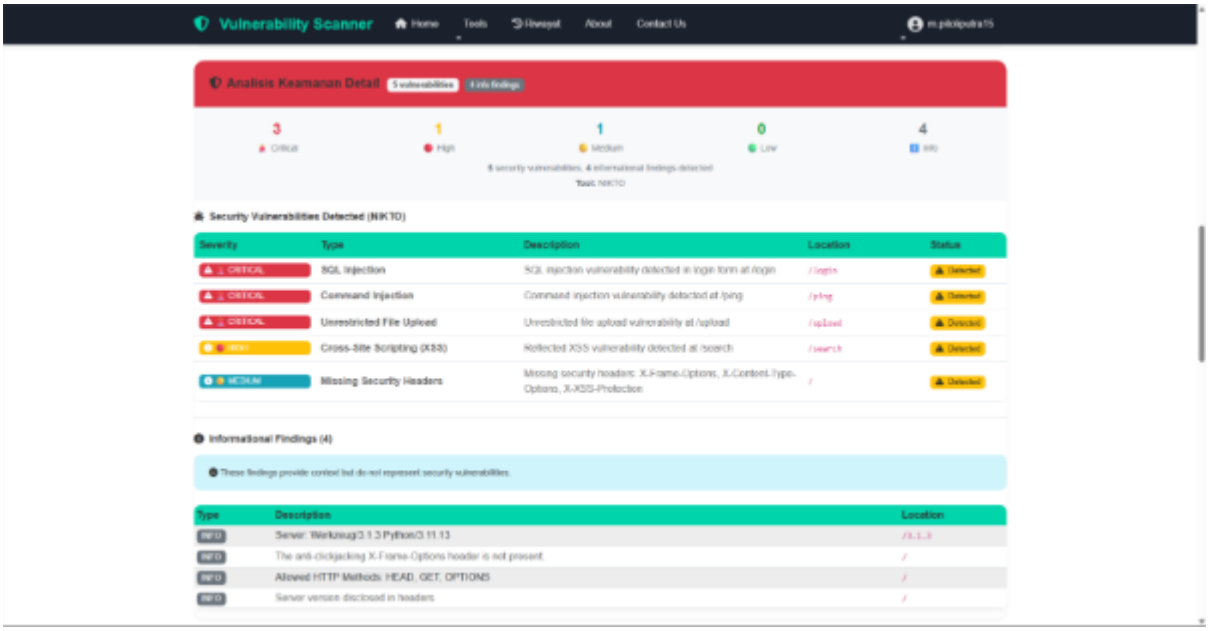

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Security Summary
Vulnerability Scanner
Quick Links
Contact

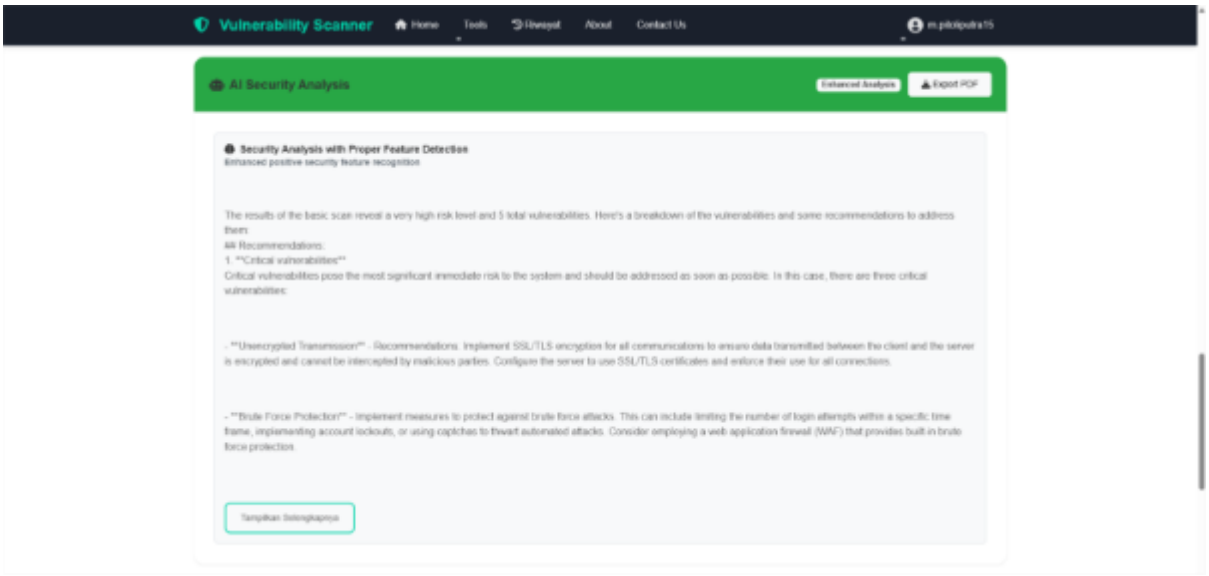
b. Pengujian *Website* (*tools* nikto) localhost beda *device*

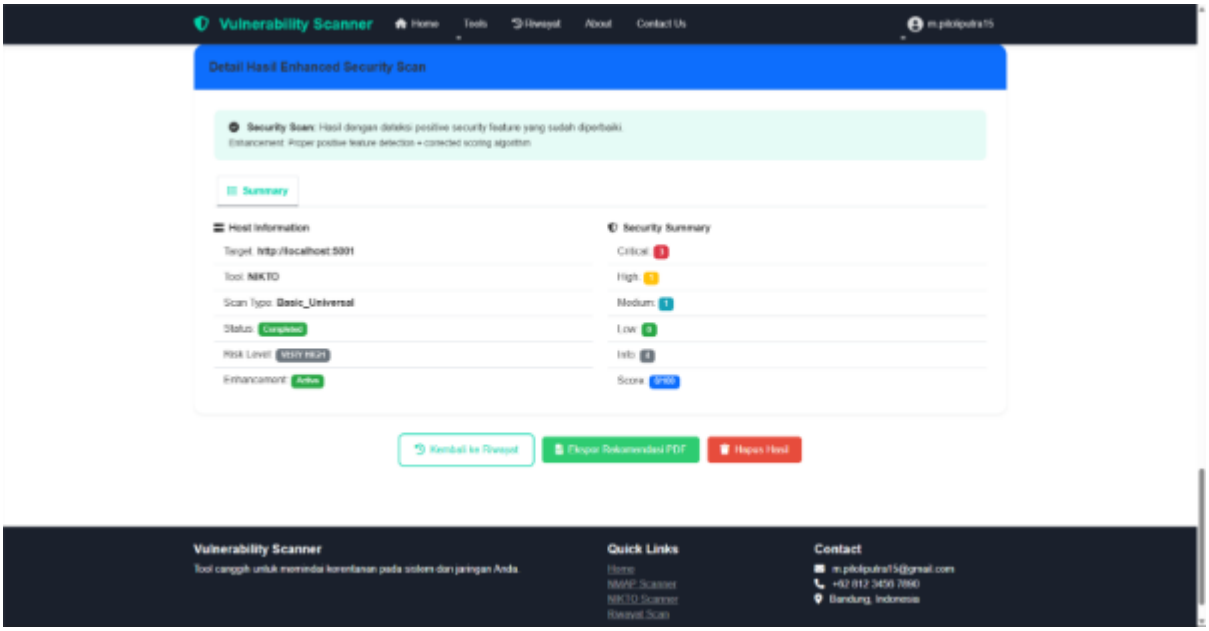
- *Website* 1 (localhost:5001) :



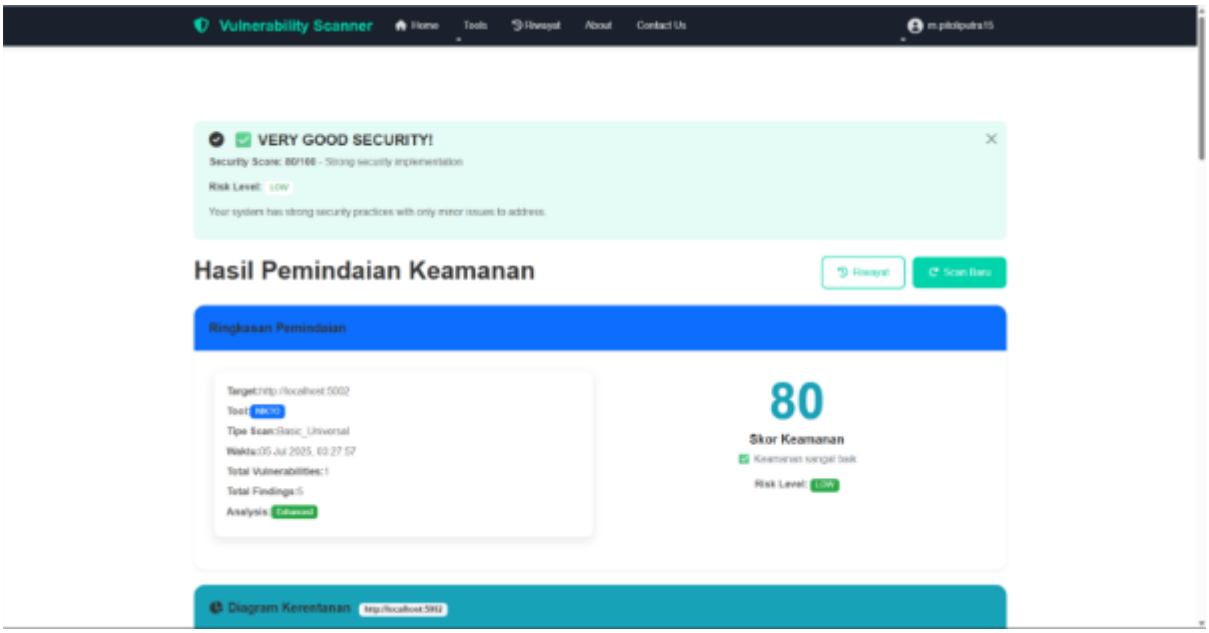


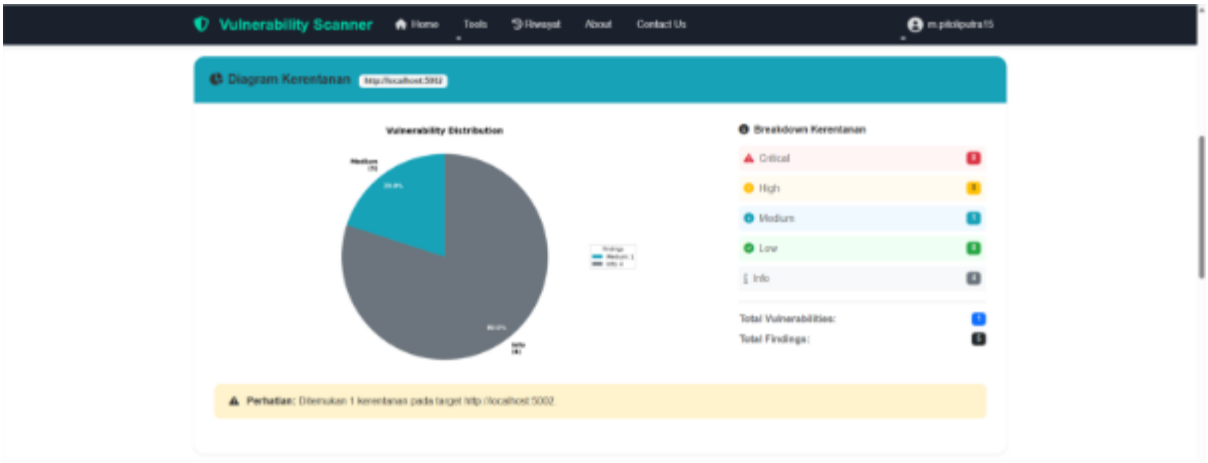

Vulnerability Scanner
Home
Tools
About
Contact Us
Analisis Keamanan Detail
3
1
1
0
4
Security Vulnerabilities Detected (NIKTO)
Severity
Type
Description
Location
Status
SQL Injection
Command Injection
Unrestricted File Upload
Cross-Site Scripting (XSS)
Missing Security Headers
Informational Findings (4)
Type
Description
Location

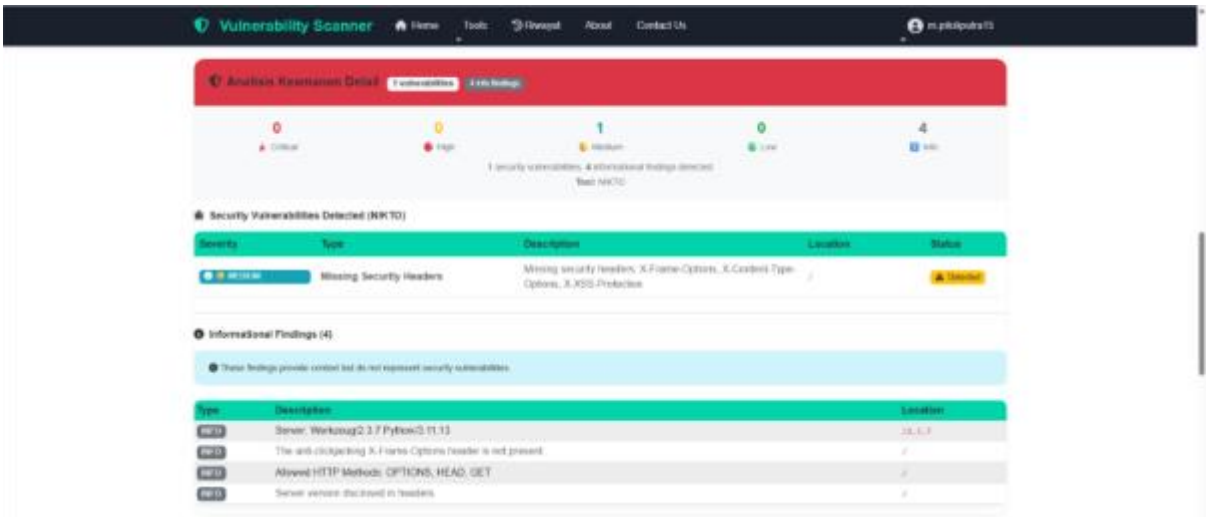

Vulnerability Scanner
Home
Tools
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

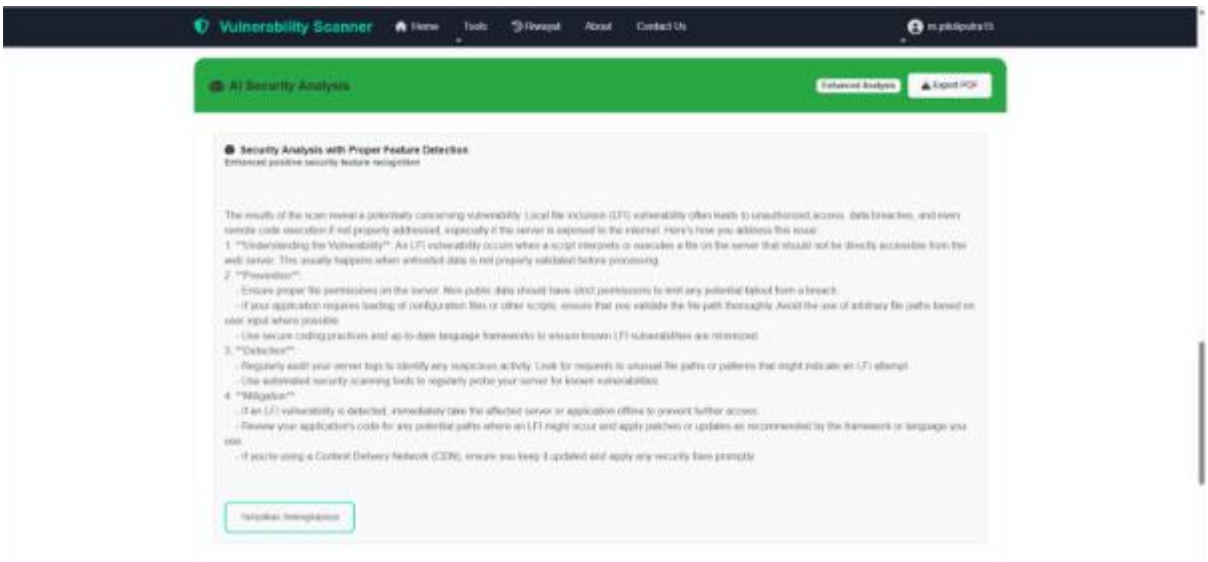


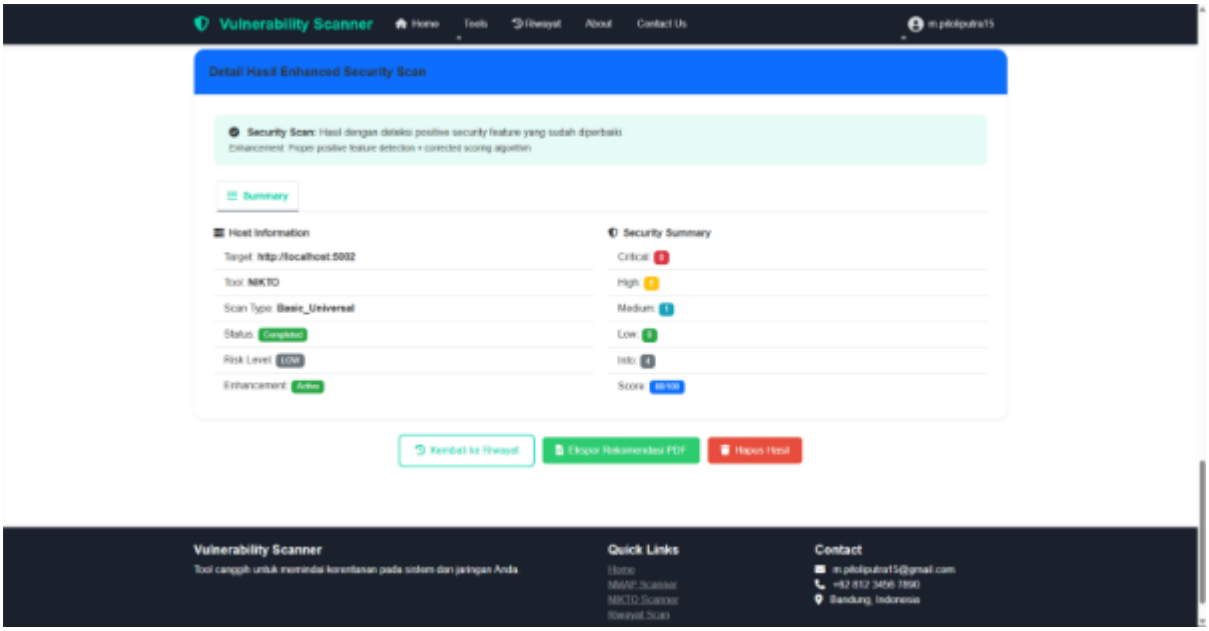
- *Website* 2 (localhost:5002)



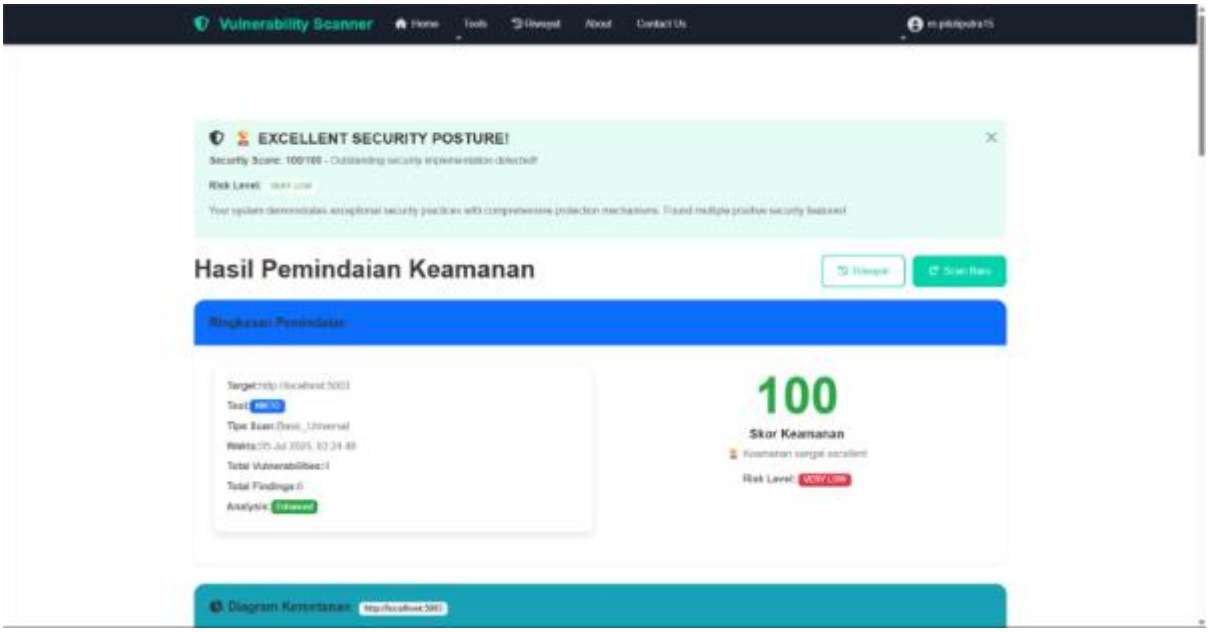

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
m.pikiputra15
Diagram Kerentanan
http://localhost:5002
Vulnerability Distribution
Medium (1)
Info (4)
Breakdown Kerentanan
Critical 0
High 0
Medium 1
Low 0
Info 4
Total Vulnerabilities: 1
Total Findings: 5
Perhatian: Ditemukan 1 kerentanan pada target http://localhost:5002

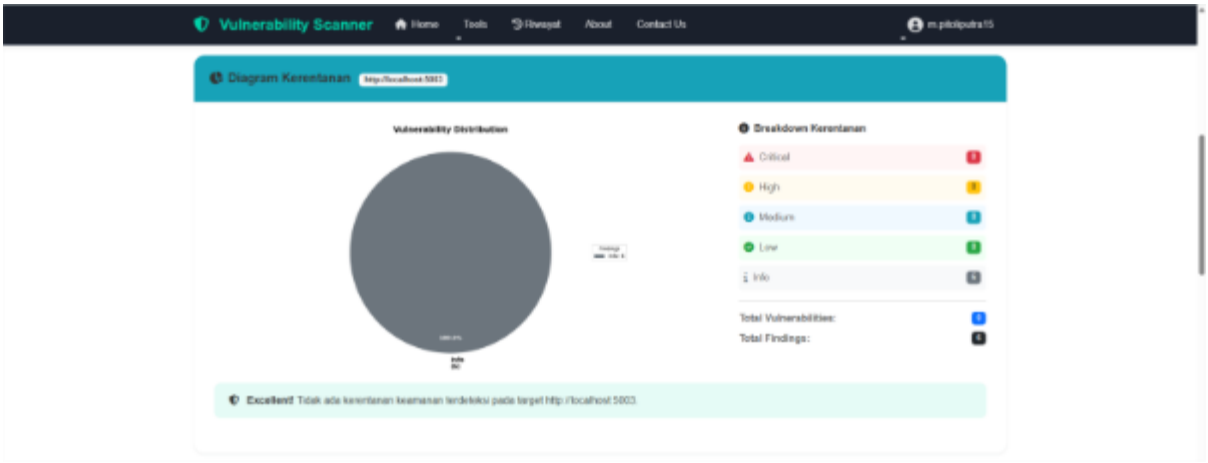

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
m.pikiputra15
Analisis Kerentanan Detail
1 vulnerabilities
4 info findings
0
Critical
0
High
1
Medium
0
Low
4
Info
1 security vulnerabilities, 4 informational findings detected
Tool: NIKTO
Security Vulnerabilities Detected (NIKTO)
Severity
Type
Description
Location
Status
MEDIUM
Missing Security Headers
Missing security headers: X-Frame-Options, X-Content-Type-Options, X-XSS-Protection
/
Detected
Informational Findings (4)
These findings provide context but do not represent security vulnerabilities
Type
Description
Location
INFO
Server: Werkzeug/2.3.7 Python/3.11.13
/
INFO
The anti-clickjacking X-Frame-Options header is not present.
/
INFO
Allowed HTTP Methods: OPTIONS, HEAD, GET
/
INFO
Server version disclosed in headers
/

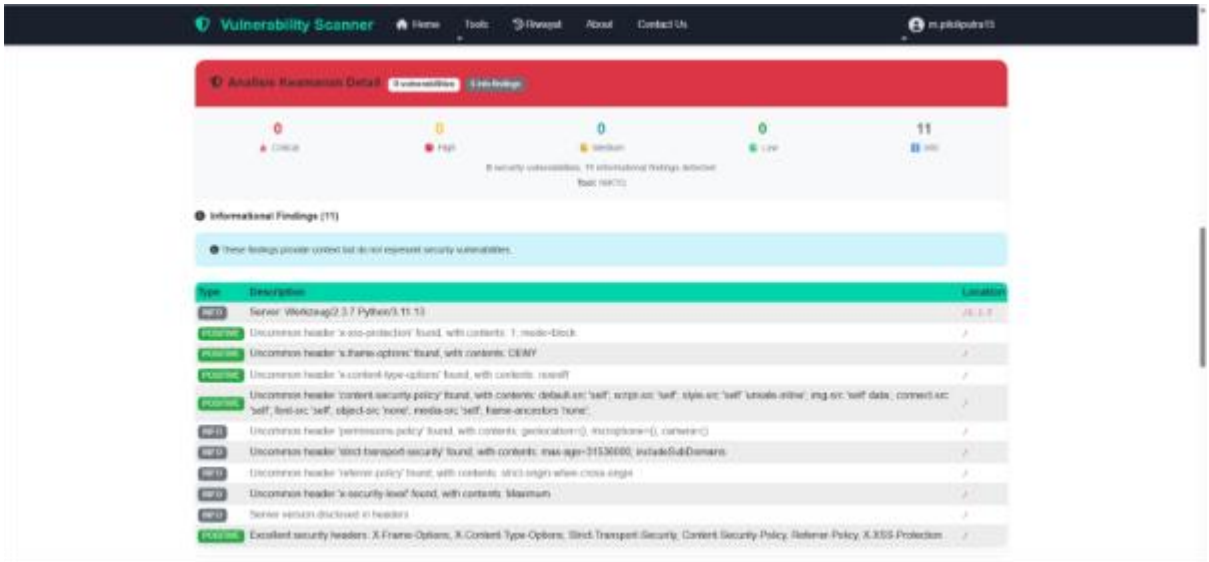

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
m.pikiputra15
AI Security Analysis
Enhanced Analysis
Export PDF
Security Analysis with Proper Feature Detection
Enhanced positive security feature recognition
The results of the scan reveal a potentially concerning vulnerability. Local file inclusion (LFI) vulnerability often leads to unauthorized access, data breaches, and even remote code execution if not properly addressed, especially if the server is exposed to the internet. Here's how you address this issue:
1. **Understanding the Vulnerability**: An LFI vulnerability occurs when a script interprets or executes a file on the server that should not be directly accessible from the web server. This usually happens when untrusted data is not properly validated before processing.
2. **Prevention**:
- Ensure proper file permissions on the server. Non-public data should have strict permissions to limit any potential fallout from a breach.
- If your application requires loading of configuration files or other scripts, ensure that you validate the file path thoroughly. Avoid the use of arbitrary file paths based on user input where possible.
- Use secure coding practices and up-to-date language frameworks to ensure known LFI vulnerabilities are minimized.
3. **Detection**:
- Regularly audit your server logs to identify any suspicious activity. Look for requests to unusual file paths or patterns that might indicate an LFI attempt.
- Use automated security scanning tools to regularly probe your server for known vulnerabilities.
4. **Mitigation**:
- If an LFI vulnerability is detected, immediately take the affected server or application offline to prevent further access.
- Review your application's code for any potential paths where an LFI might occur and apply patches or updates as recommended by the framework or language you use.
- If you're using a Content Delivery Network (CDN), ensure you keep it updated and apply any security fixes promptly.
Sembunyikan Interpretasi

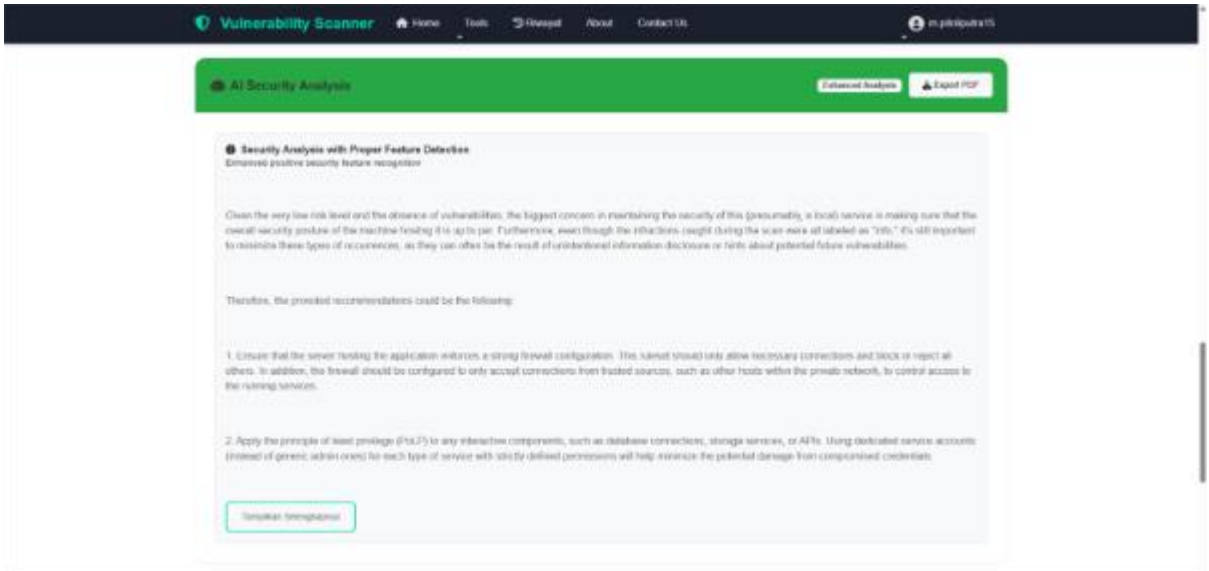


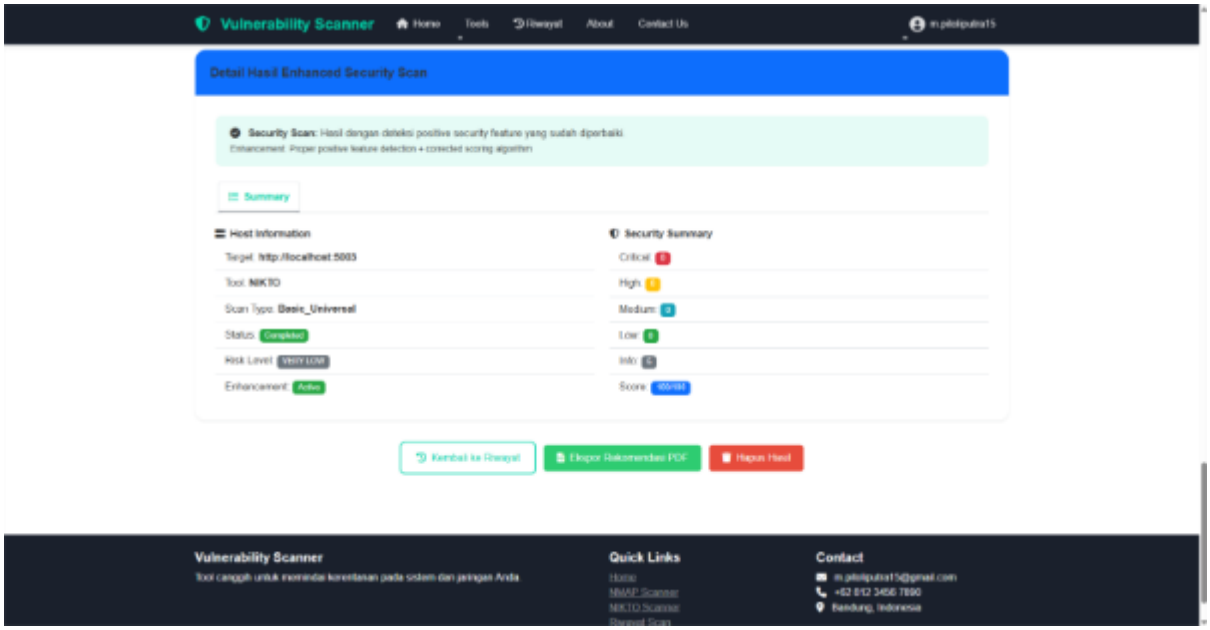
- *Website* 3 (localhost:5003)




Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Diagram Kerentanan
Vulnerability Distribution
Breakdown Kerentanan
Critical
High
Medium
Low
Info
Total Vulnerabilities:
Total Findings:
Excellent! Tidak ada kerentanan keamanan terdeteksi pada target http://localhost:5003.

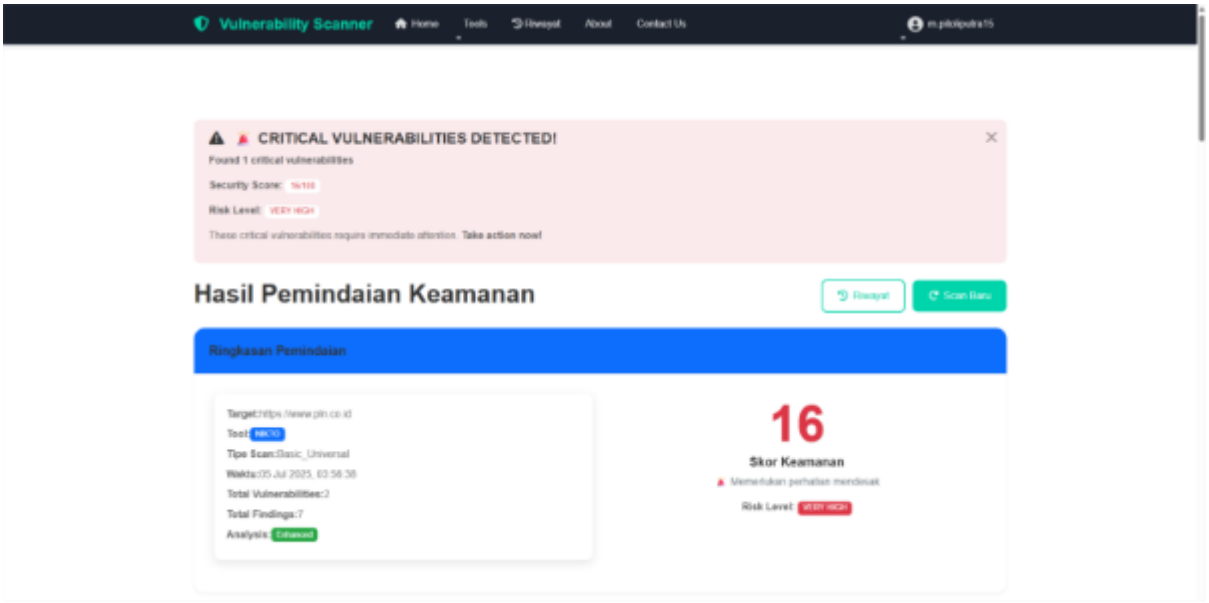

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
0
0
0
0
11
Informational Findings (11)
These findings provide context but do not represent security vulnerabilities.
Type
Description
Location

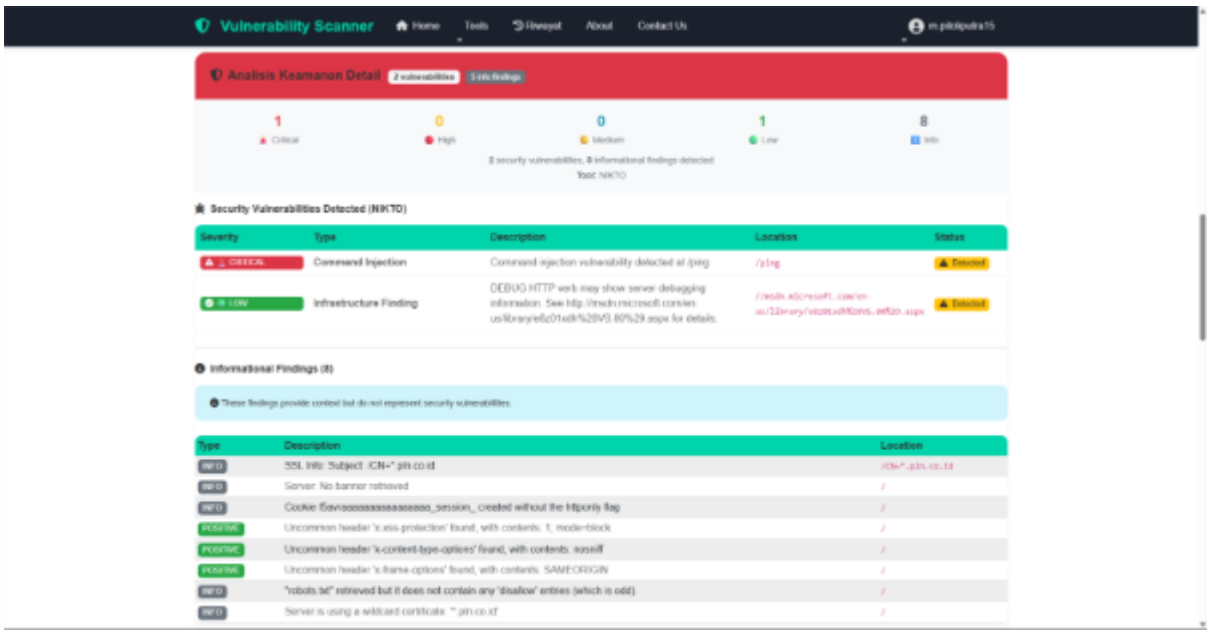

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Export PDF
Security Analysis with Proper Feature Detection

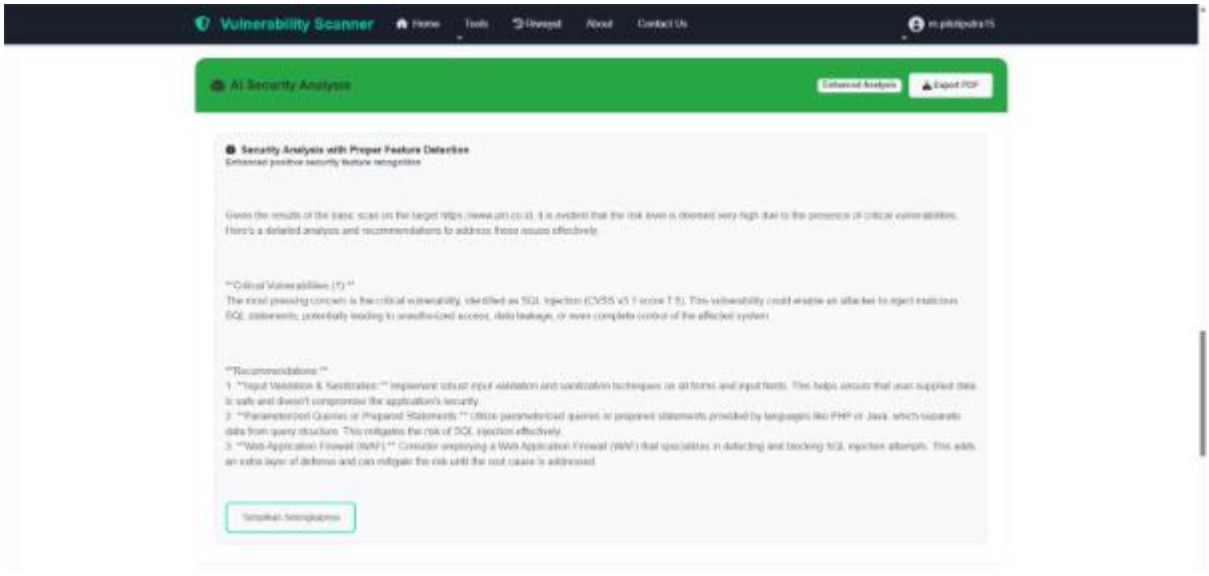

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Security Summary
Kembali ke Riwayat
Ekspor Rekomendasi PDF
Hapus Hasil
Vulnerability Scanner
Quick Links
Contact

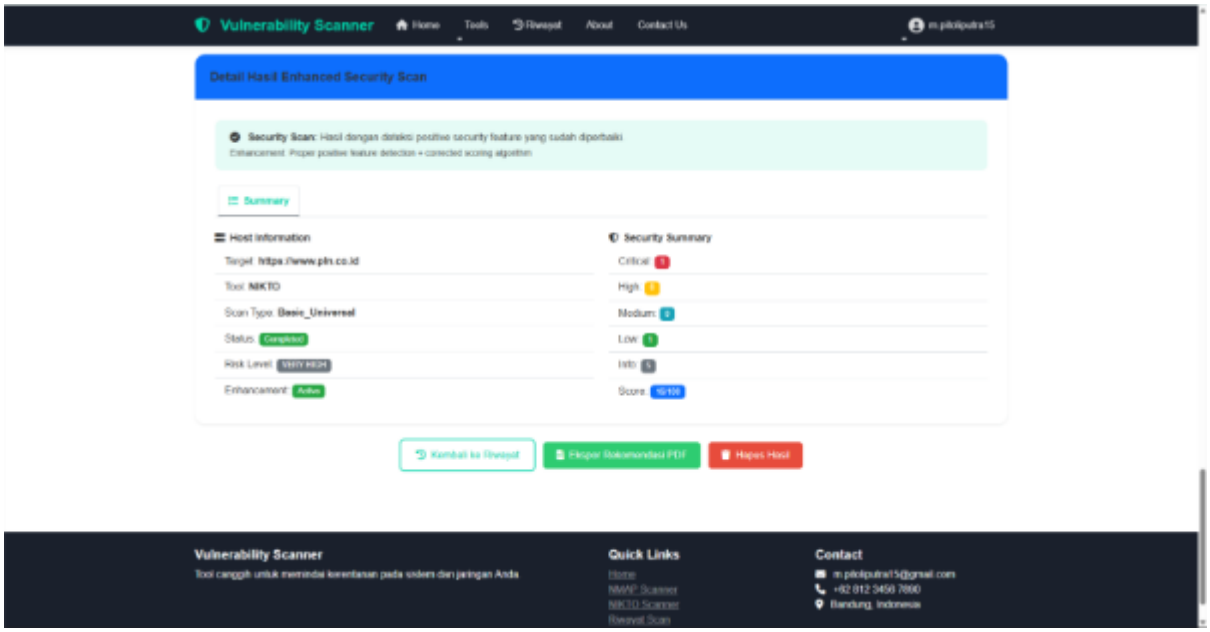
c. Pengujian (*tools* nikto) *Website* internet :

- *Website* 1 (https://www.pln.co.id)

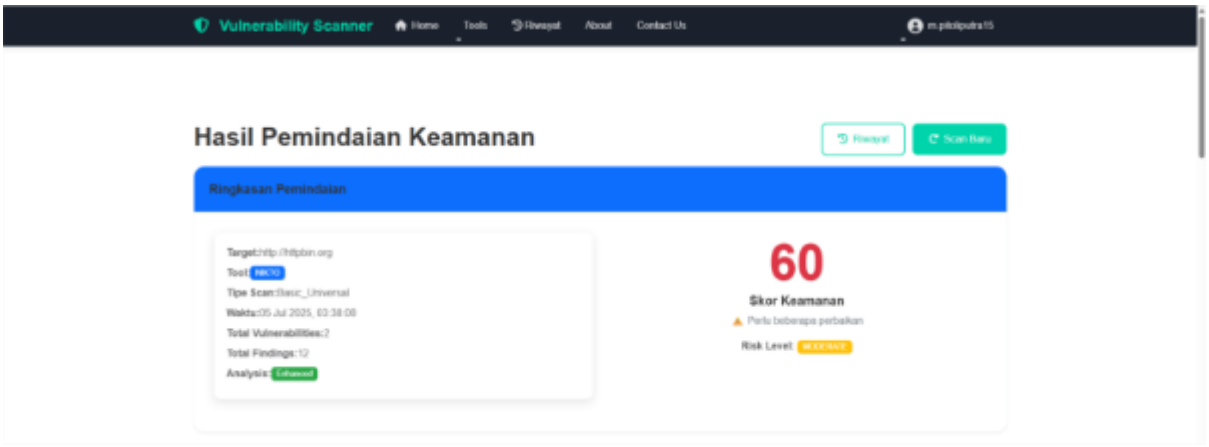


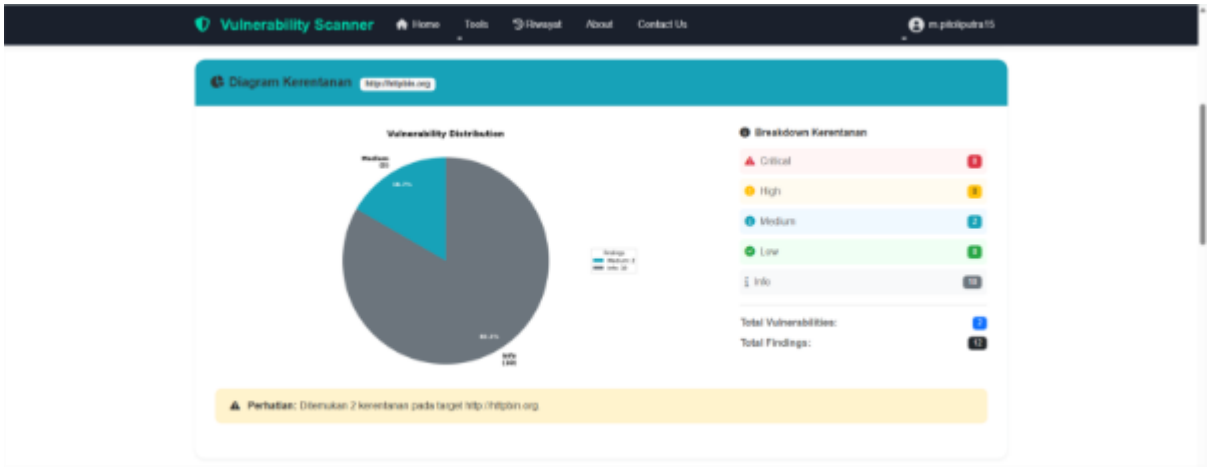


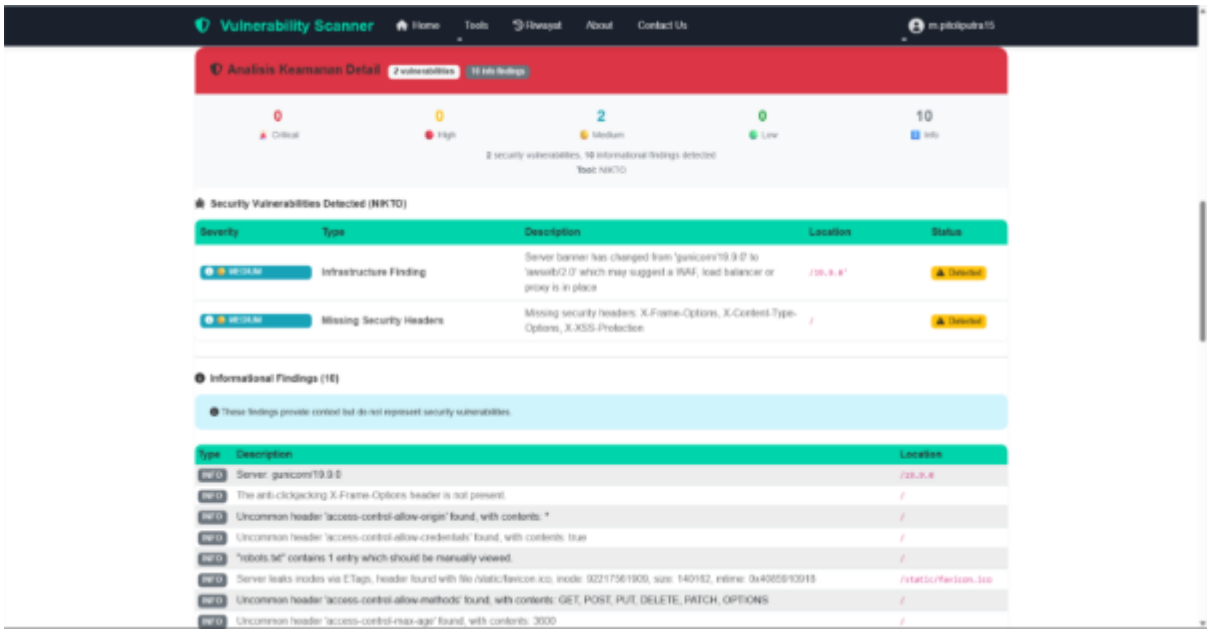

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

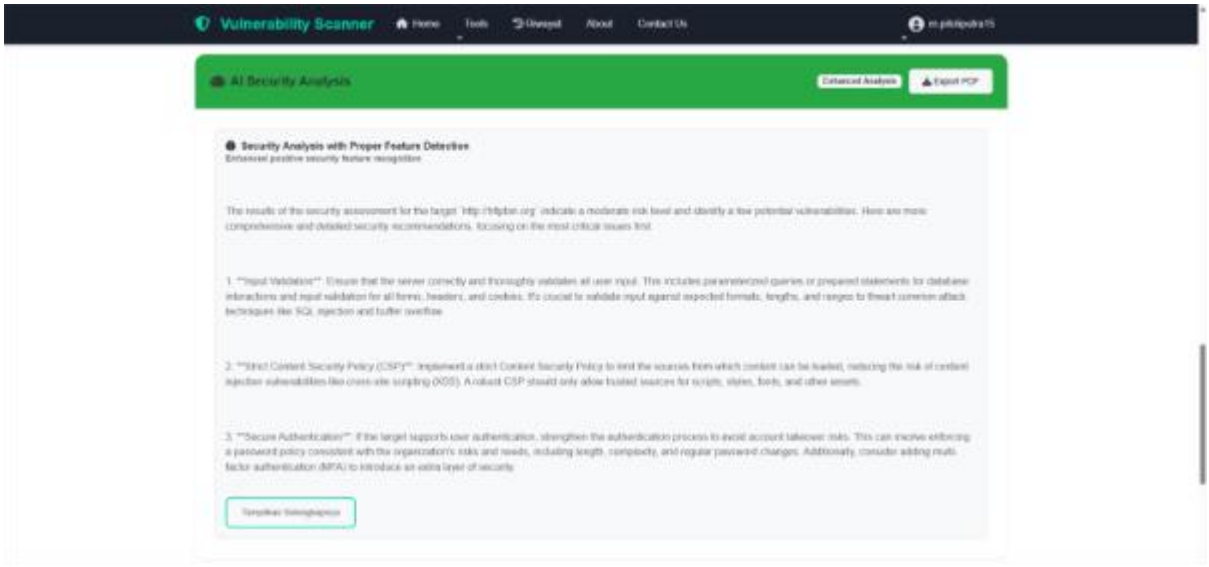

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Security Summary
Vulnerability Scanner
Quick Links
Contact

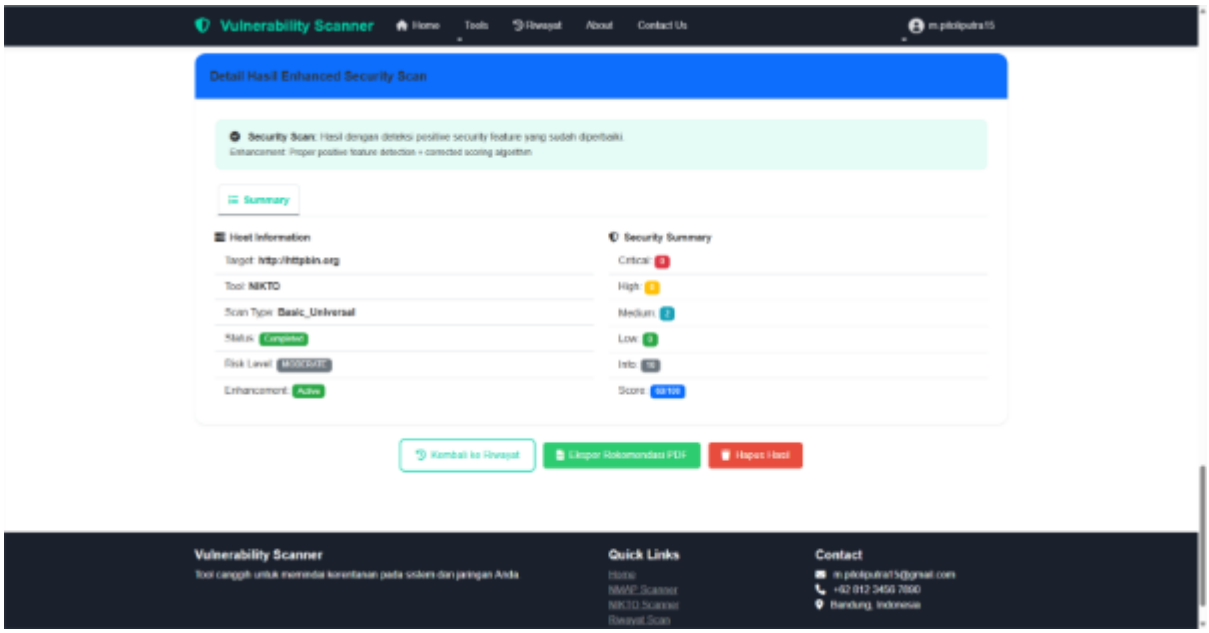
- *Website* 2 (http://httpbin.org)



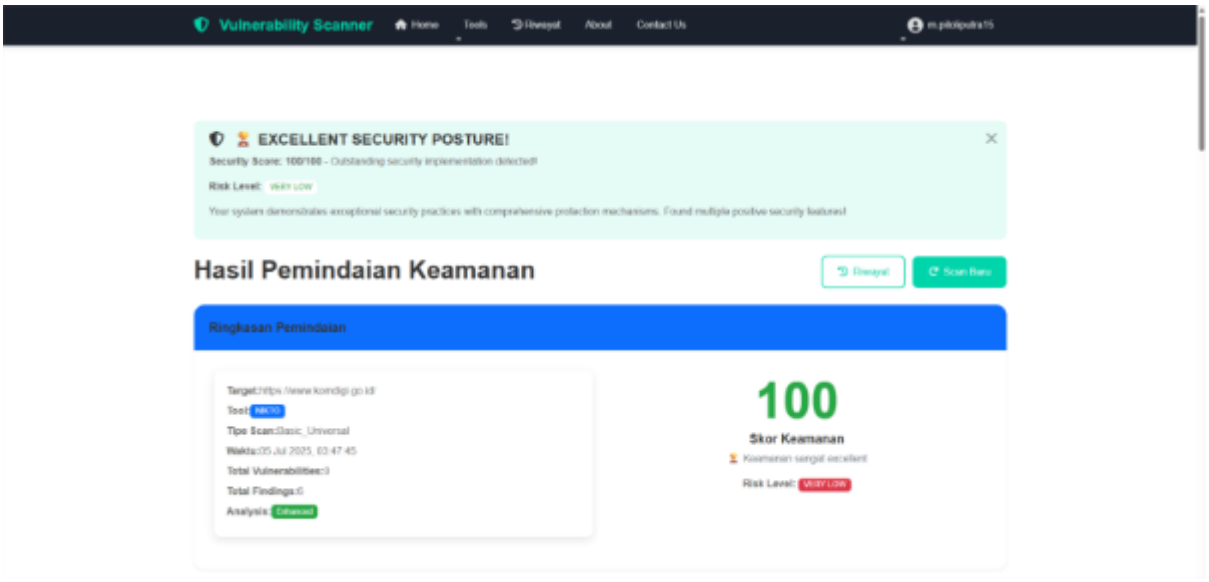


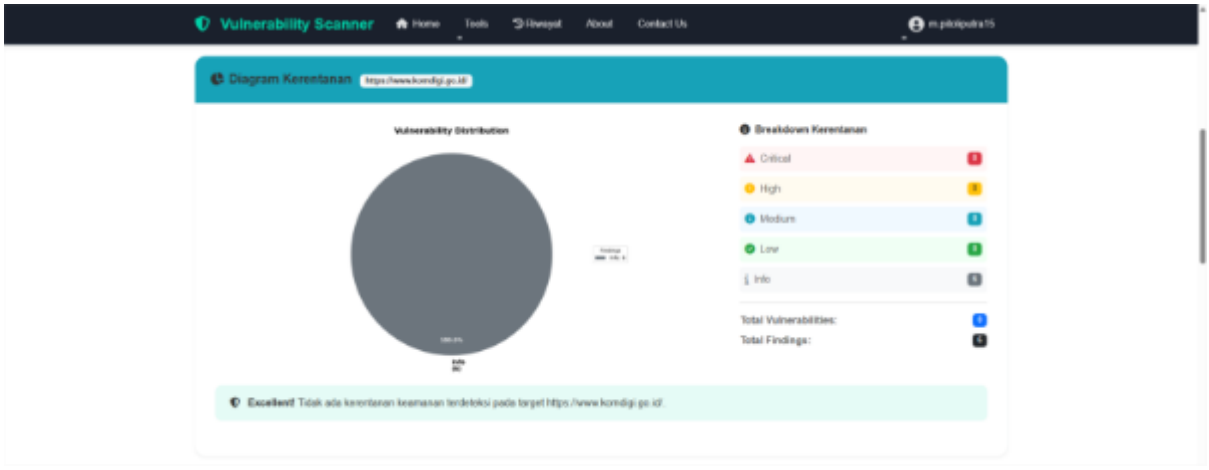


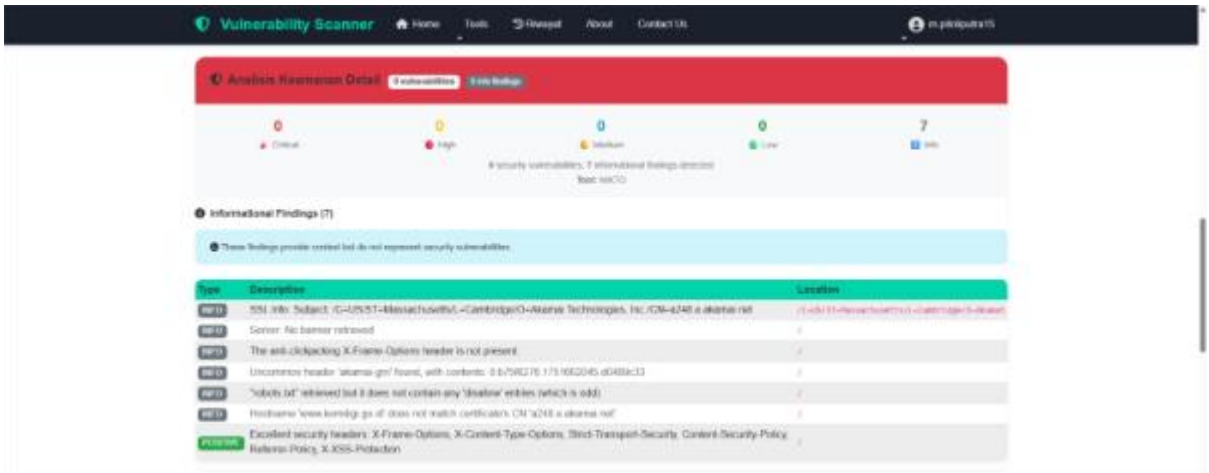

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

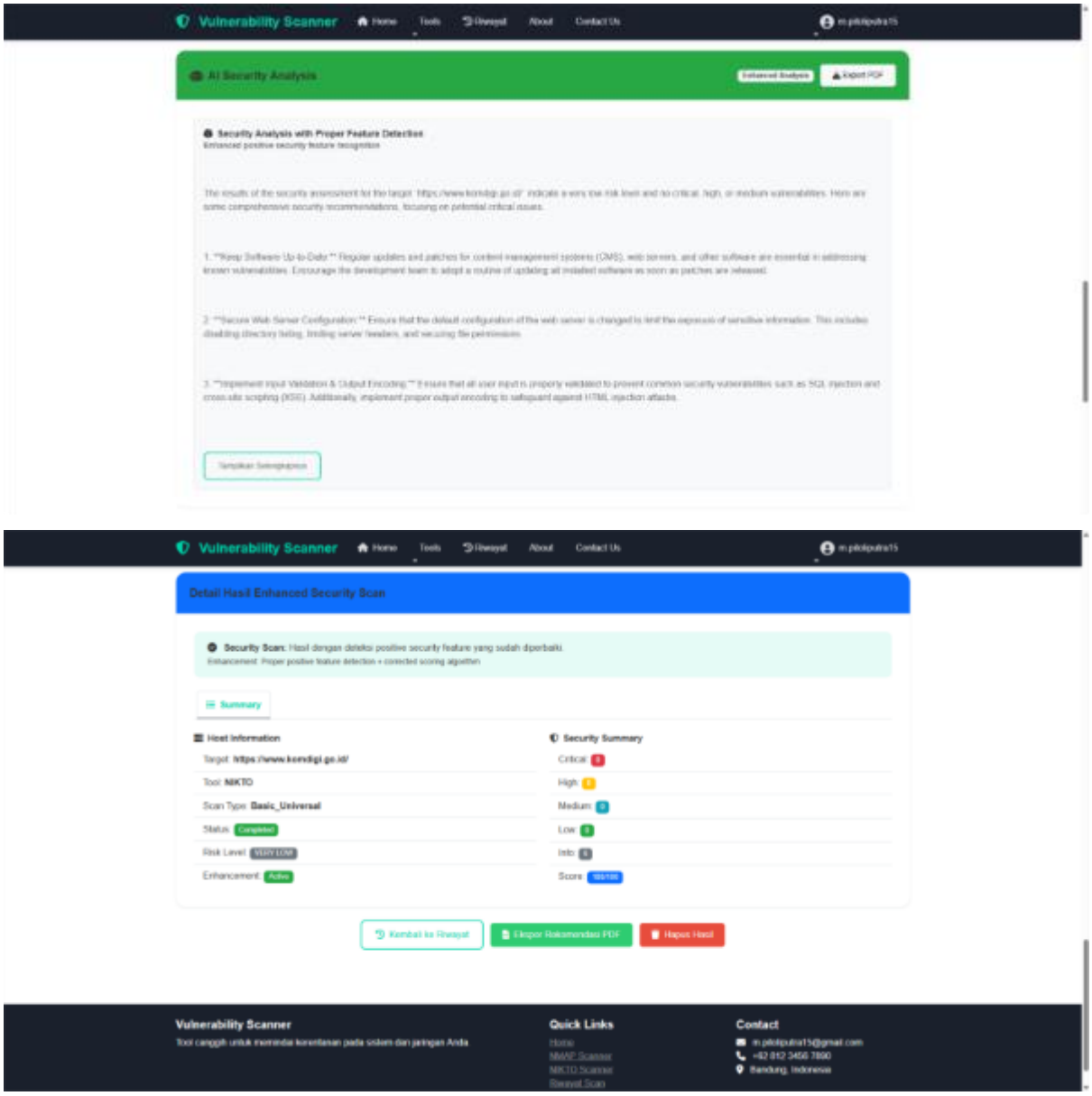

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Security Summary
Target: http://httpbin.org
Tool: NIKTO
Scan Type: Basic_Universal
Status: Completed
Enhancement: Active
Critical: 0
High: 1
Medium: 2
Low: 0
Info: 10
Score: 60/100
Kembali ke Riwayat
Ekspor Rekomendasi PDF
Hapus Hasil
Vulnerability Scanner
Tool canggih untuk memindai kerentanan pada sistem dan jaringan Anda
Quick Links
Home
MWE Scanner
NIKTO Scanner
Riwayat Scan
Contact
Bandung, Indonesia

- *Website* 3 (https://www.komdigi.go.id/)





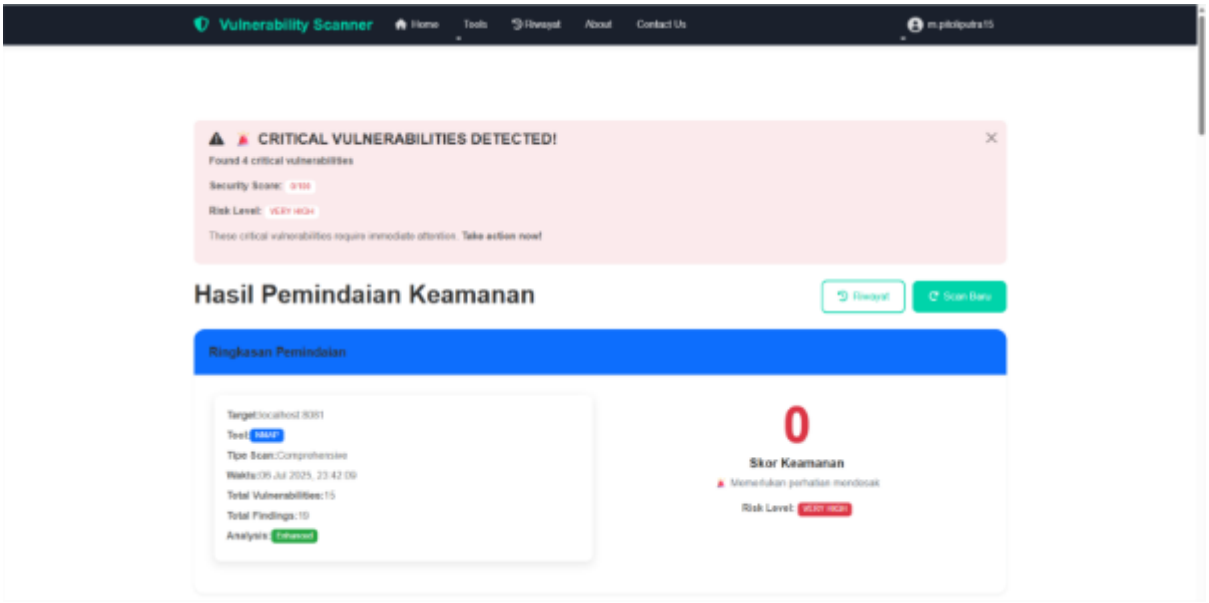


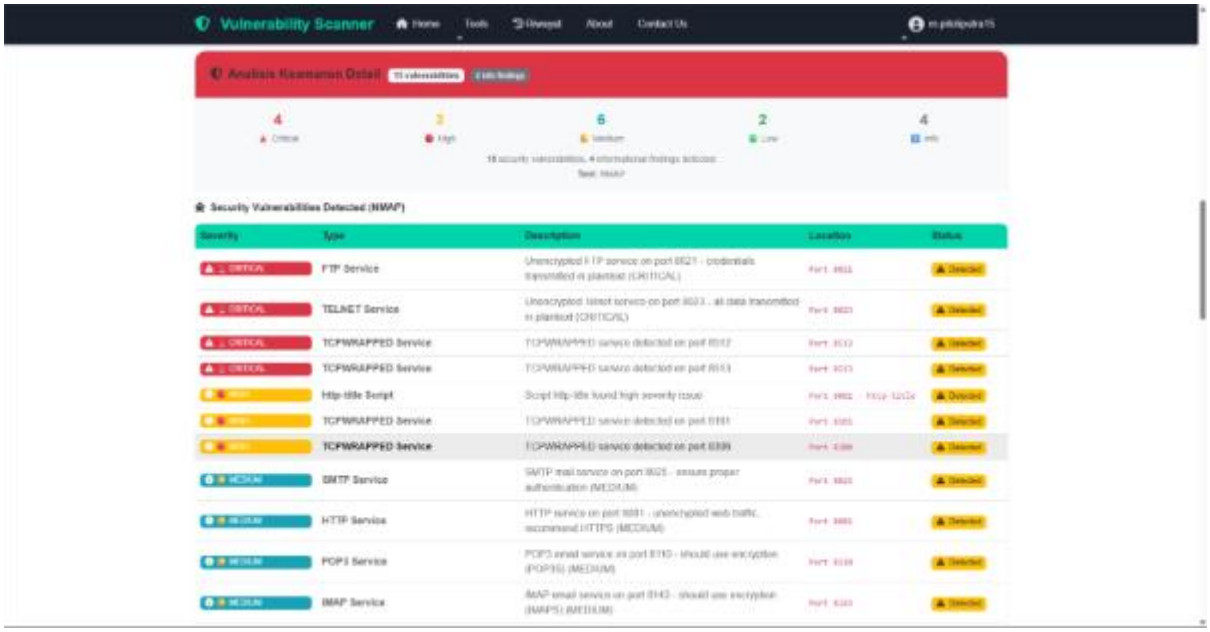

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Export PDF
Security Analysis with Proper Feature Detection

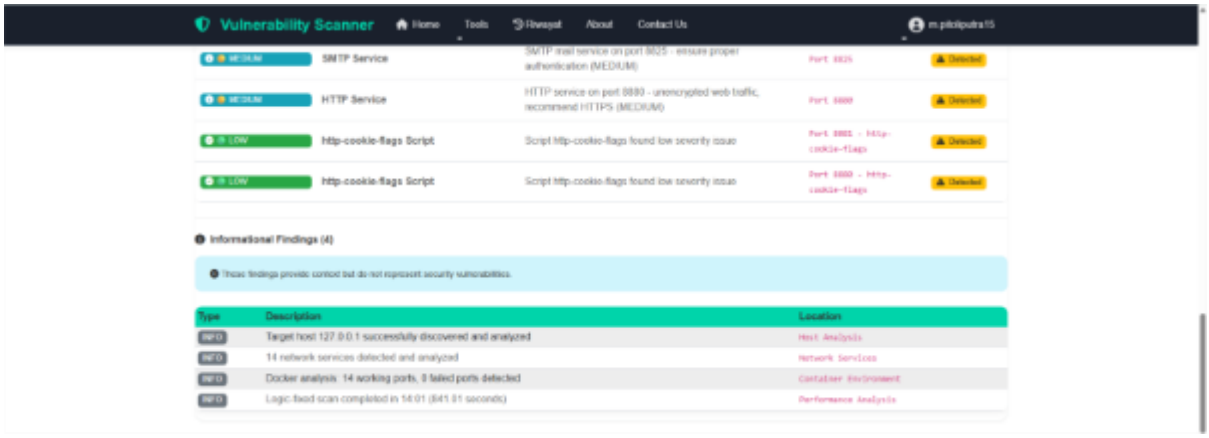
Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Security Summary
Kembali ke Riwayat
Hapus Hasil
Vulnerability Scanner
Quick Links
Contact

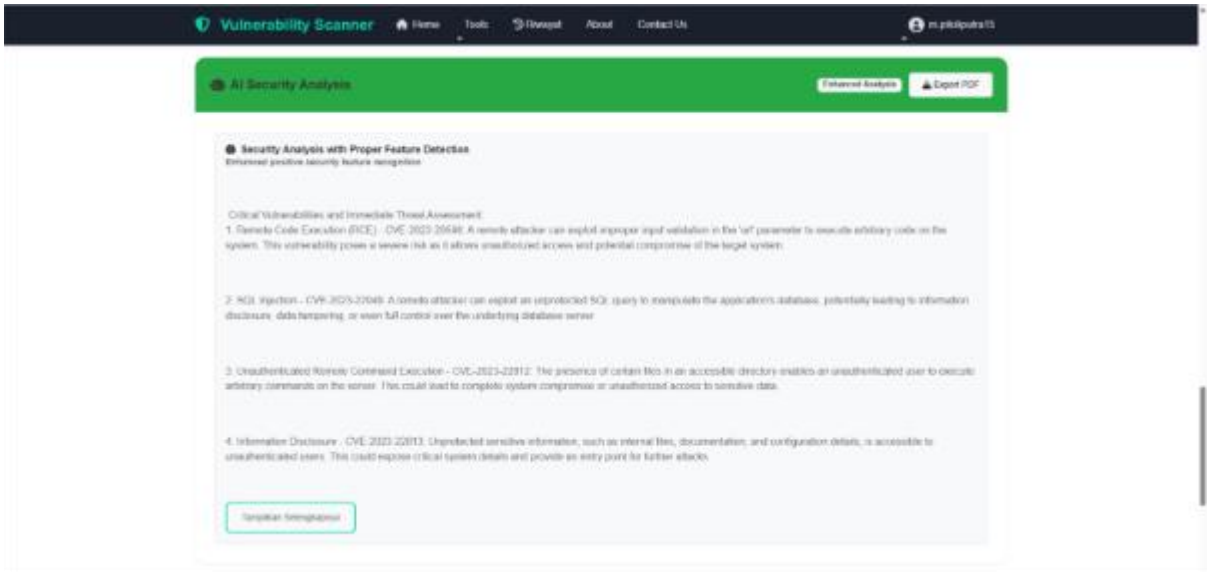
d. Pengujian (nmap scanner) *Website* localhost 1 *device*

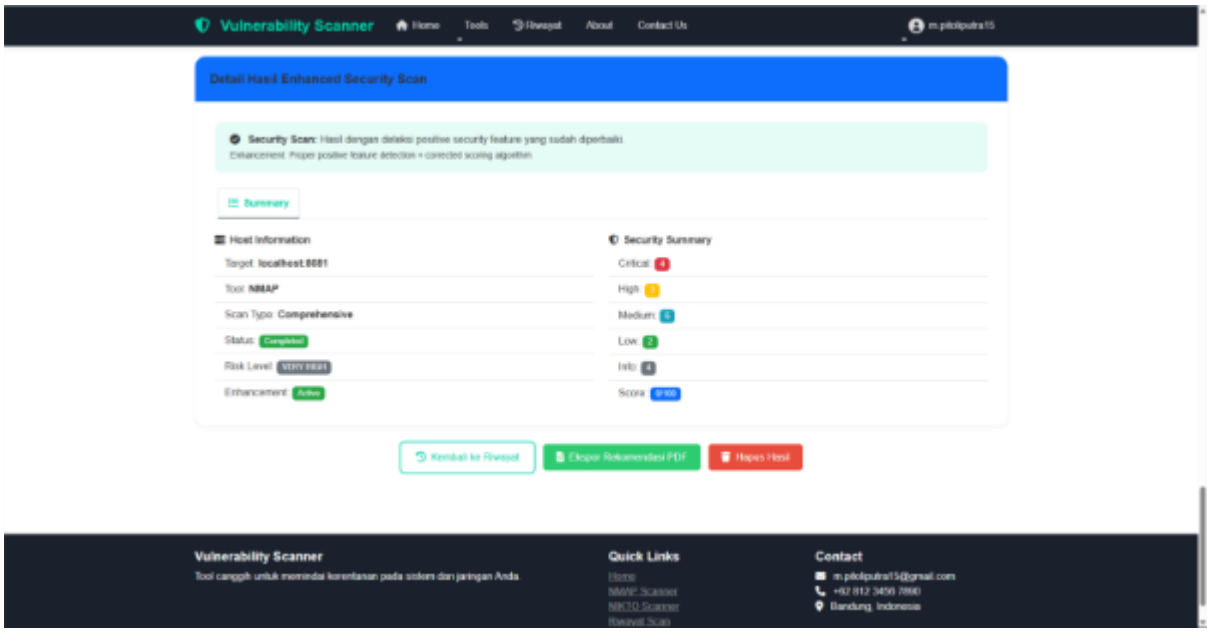
- *Website* 1 (localhost:8081)





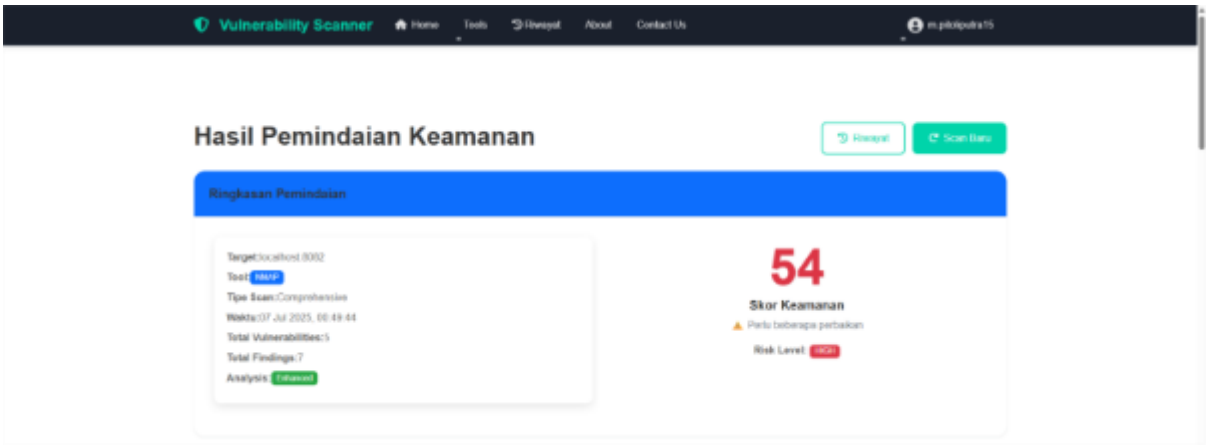

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
SMTP Service
HTTP Service
http-cookie-flags Script
http-cookie-flags Script
Informational Findings (4)
Type
Description
Location
Target host 127.0.0.1 successfully discovered and analyzed
14 network services detected and analyzed
Docker analysis: 14 working ports, 8 failed ports detected
Logic-based scan completed in 14.01 (841.81 seconds)

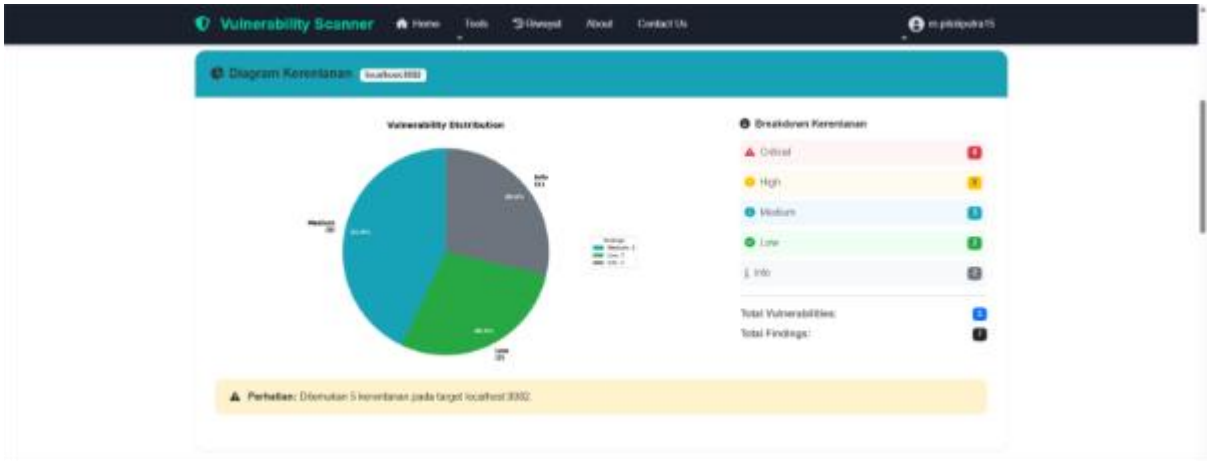

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

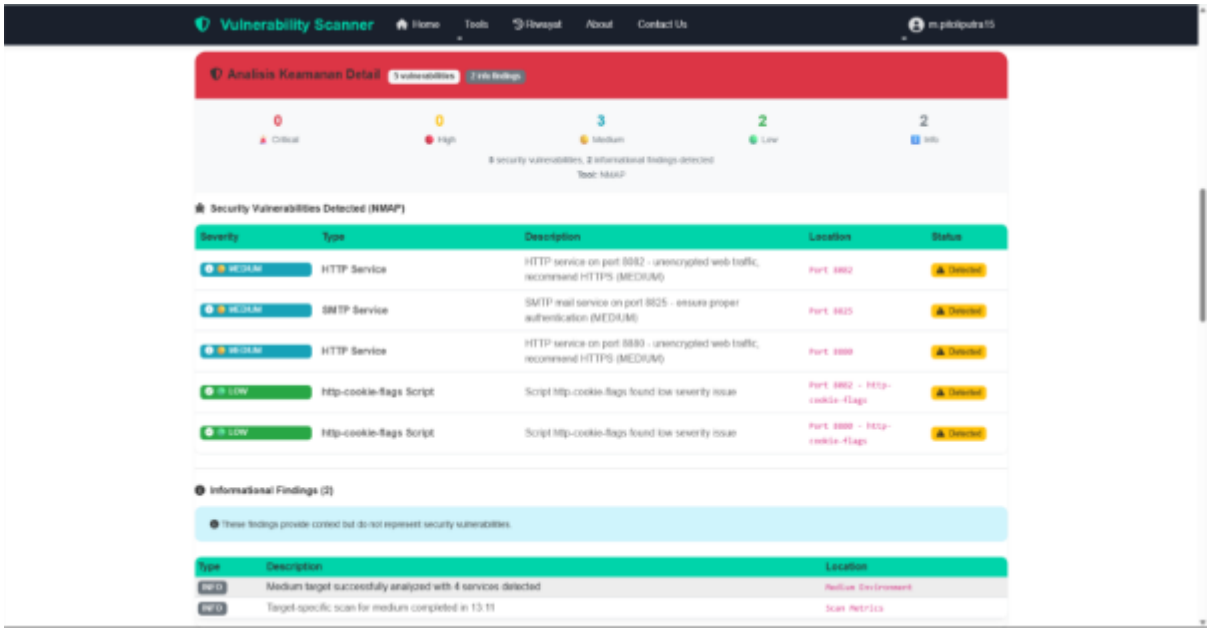

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Target: localhost:8081
Tool: NMAP
Scan Type: Comprehensive
Status: Completed
Risk Level:
Enhancement: Active
Security Summary
Critical: 4
High: 1
Medium: 5
Low: 2
Info: 4
Score:

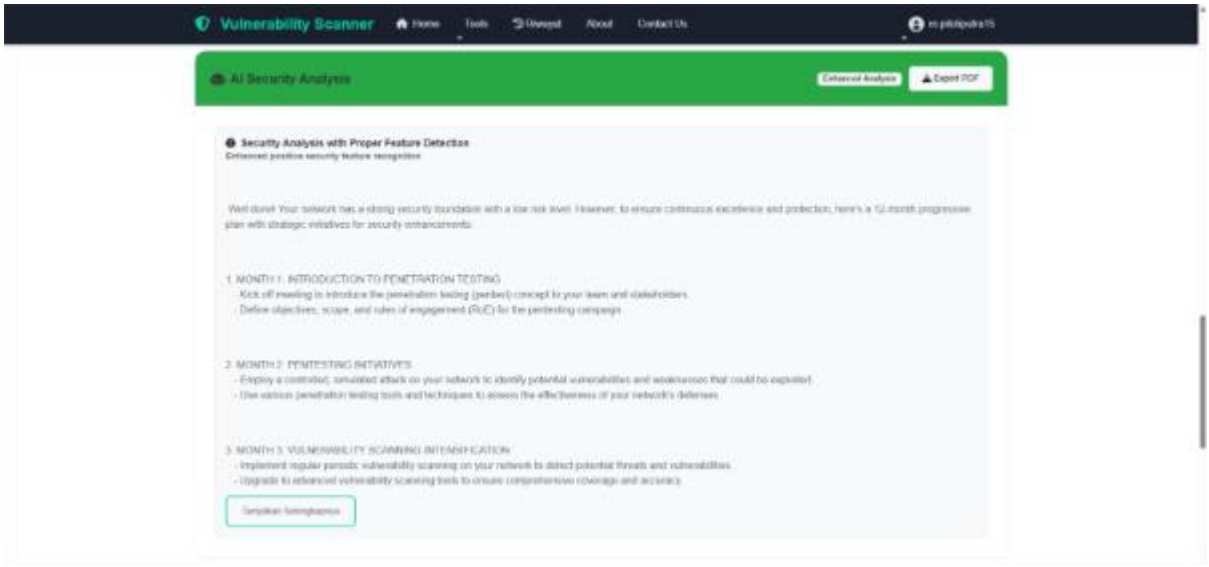
Vulnerability Scanner
Tool canggih untuk memindai kerentanan pada sistem dan jaringan Anda
Quick Links
Home
MMP Scanner
NIKTO Scanner
Riwayat Scan
Contact
Bandung, Indonesia

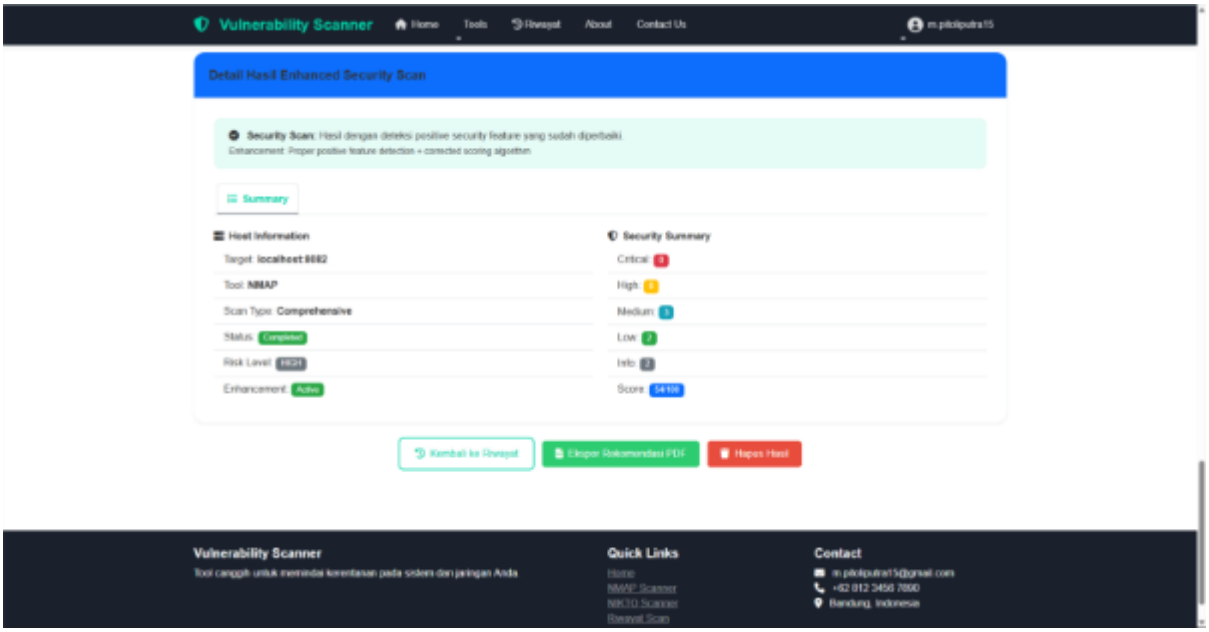
- *Website* 2 (localhost:8082)



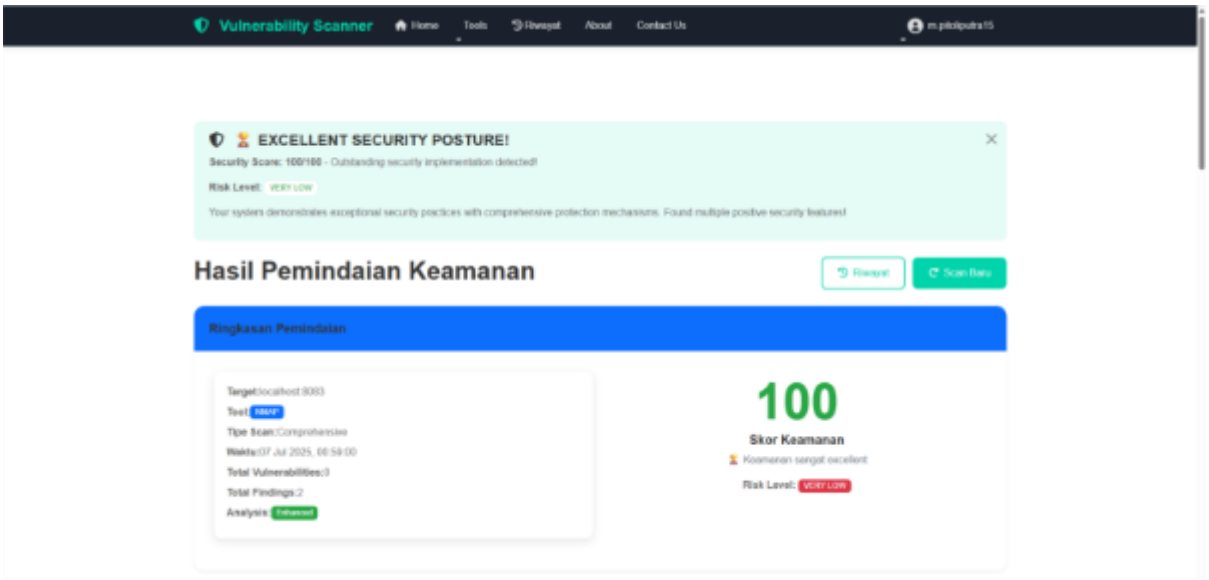


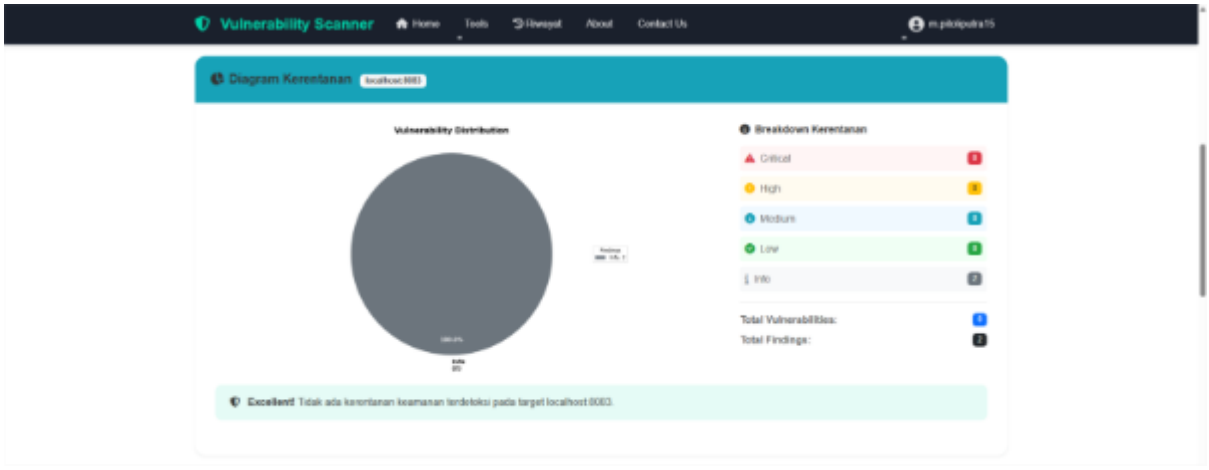


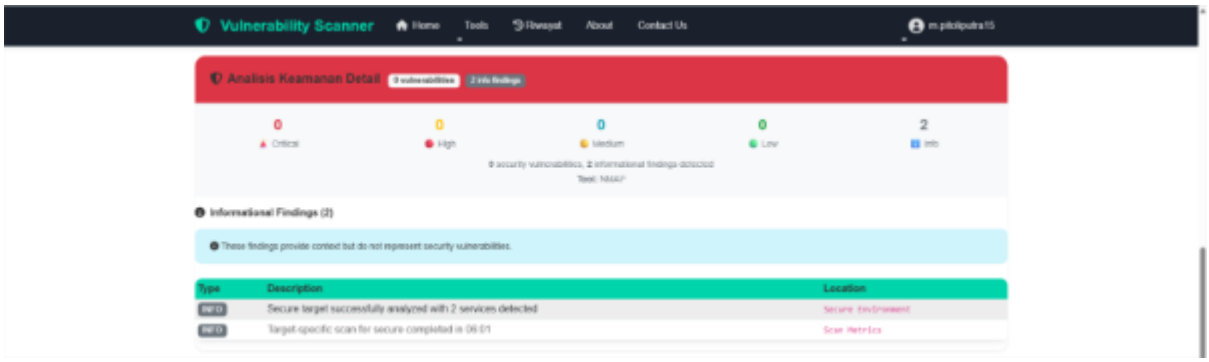

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

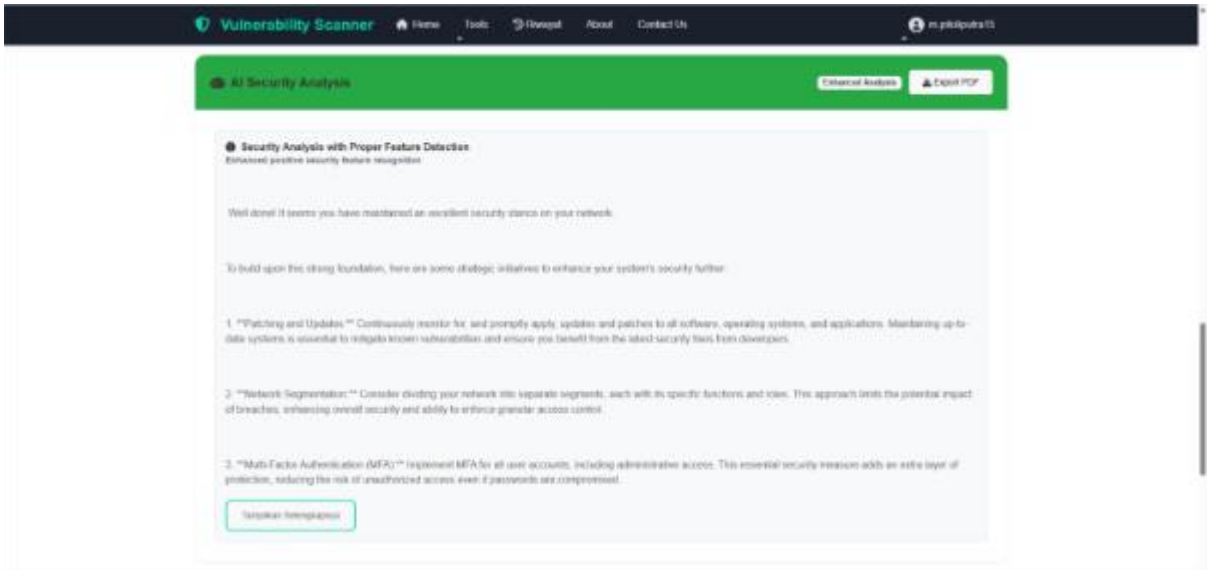

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Security Summary
Vulnerability Scanner
Quick Links
Contact

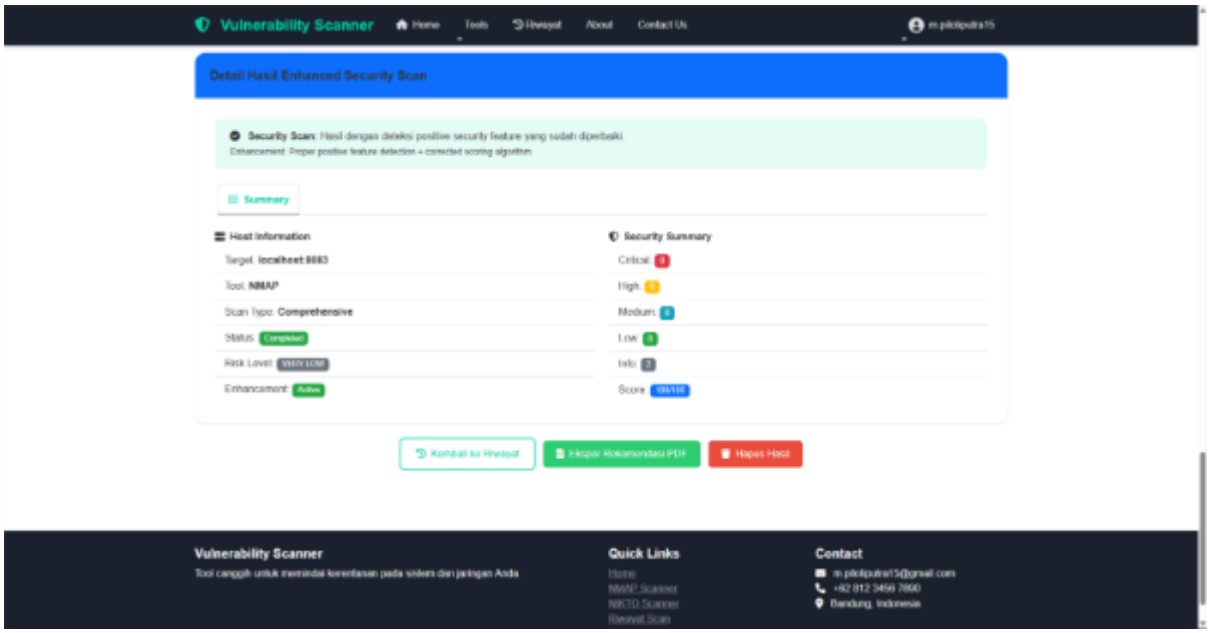
- *Website* 3 (localhost:8083)





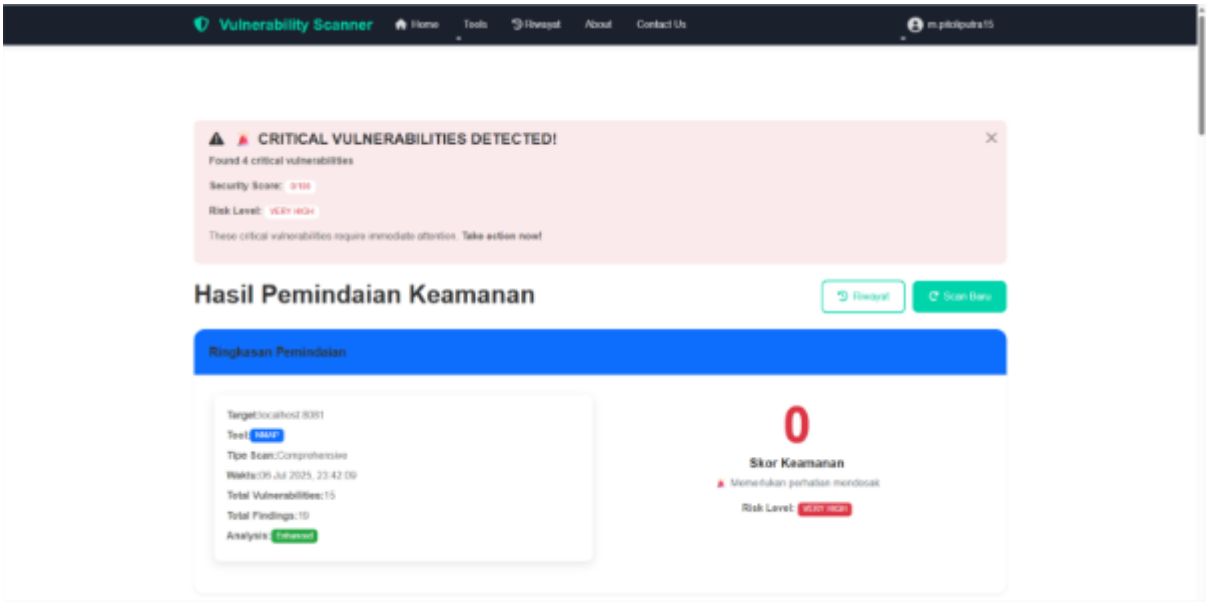


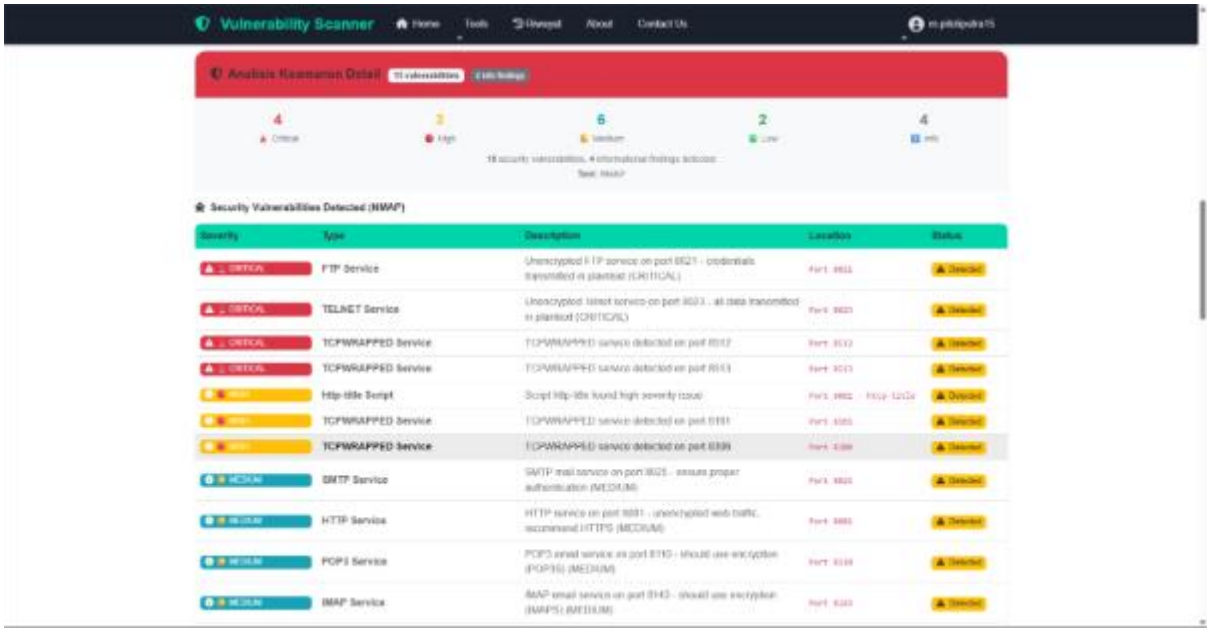

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Export PDF
Security Analysis with Proper Feature Detection

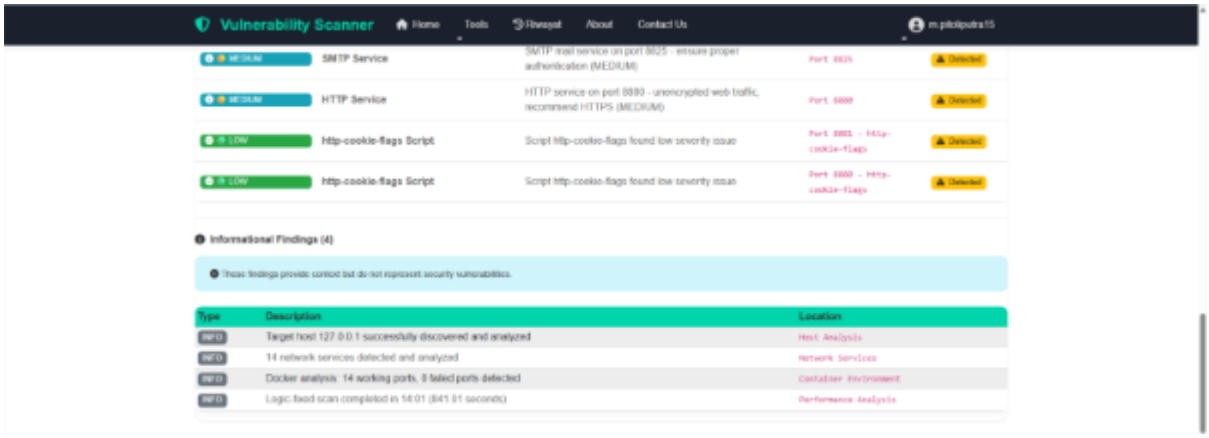

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Security Summary
Kembali ke Riwayat
Hapus Hasil
Vulnerability Scanner
Quick Links
Contact

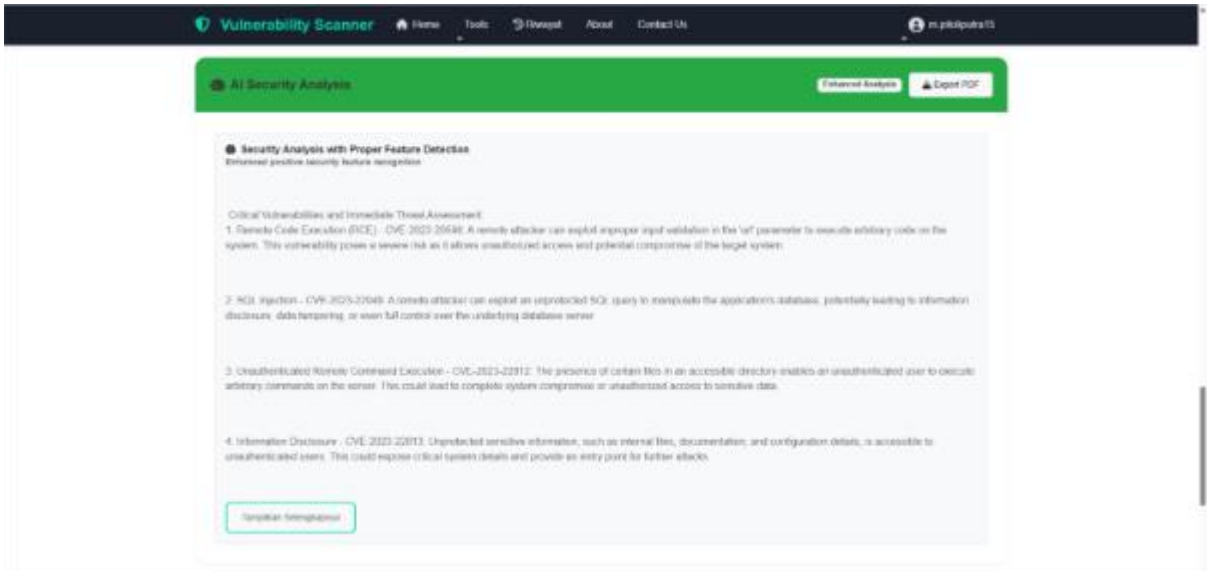
e. Pengujian (*tools* nmap) *Website* localhost beda *device*

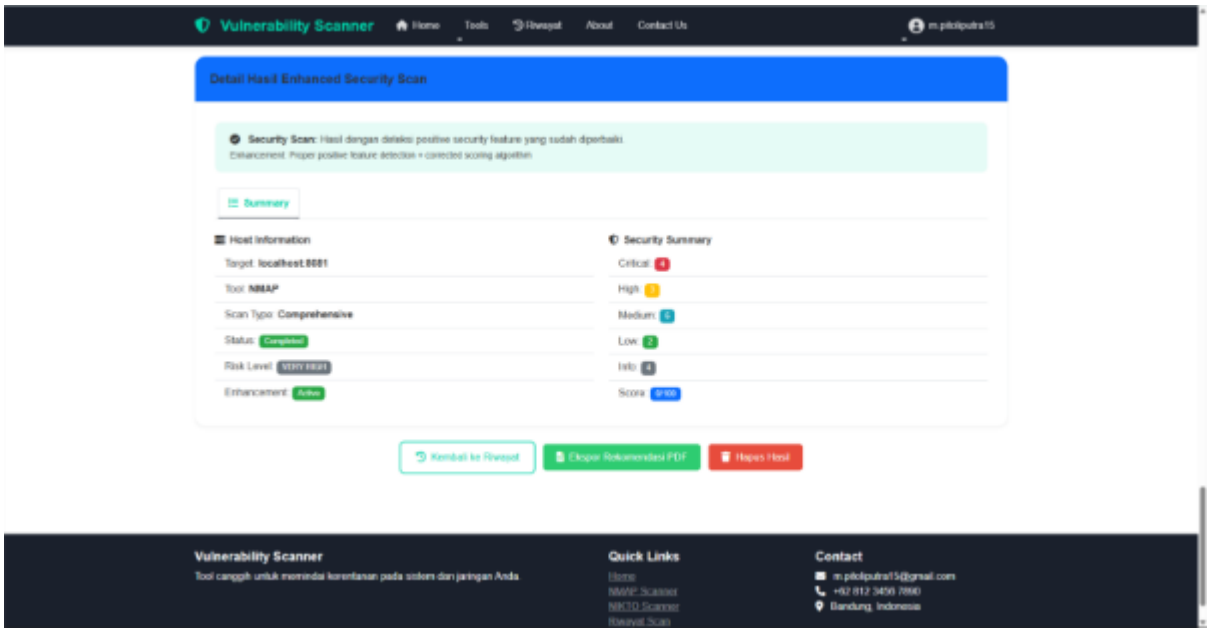
- *Website* 1 (localhost:8081)



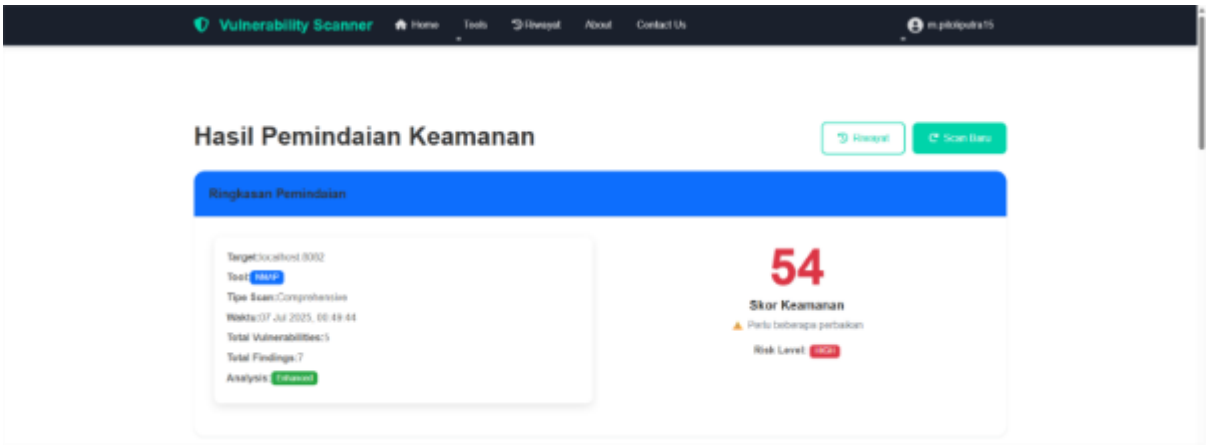


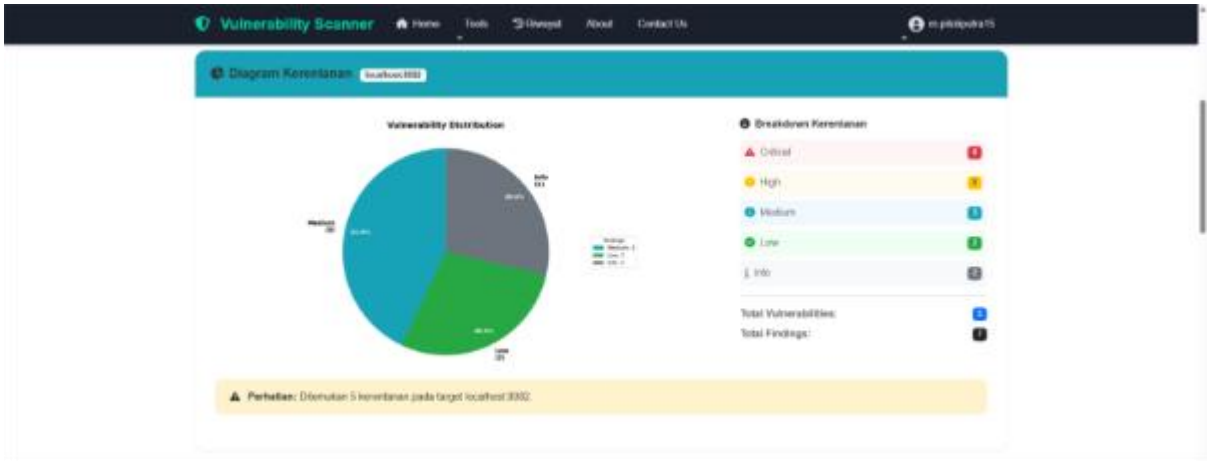

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
SMTP Service
HTTP Service
http-cookie-flags Script
http-cookie-flags Script
Informational Findings (4)
Type
Description
Location
Target host 127.0.0.1 successfully discovered and analyzed
14 network services detected and analyzed
Docker analysis: 14 working ports, 8 failed ports detected
Logic-based scan completed in 14.01 (841.81 seconds)

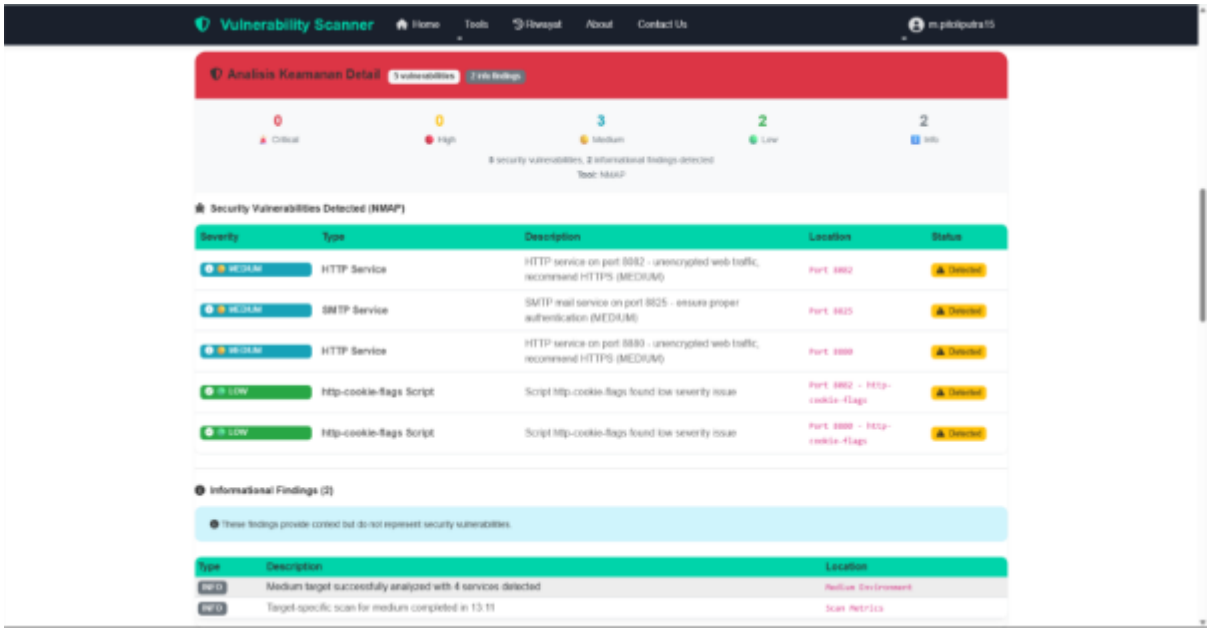

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

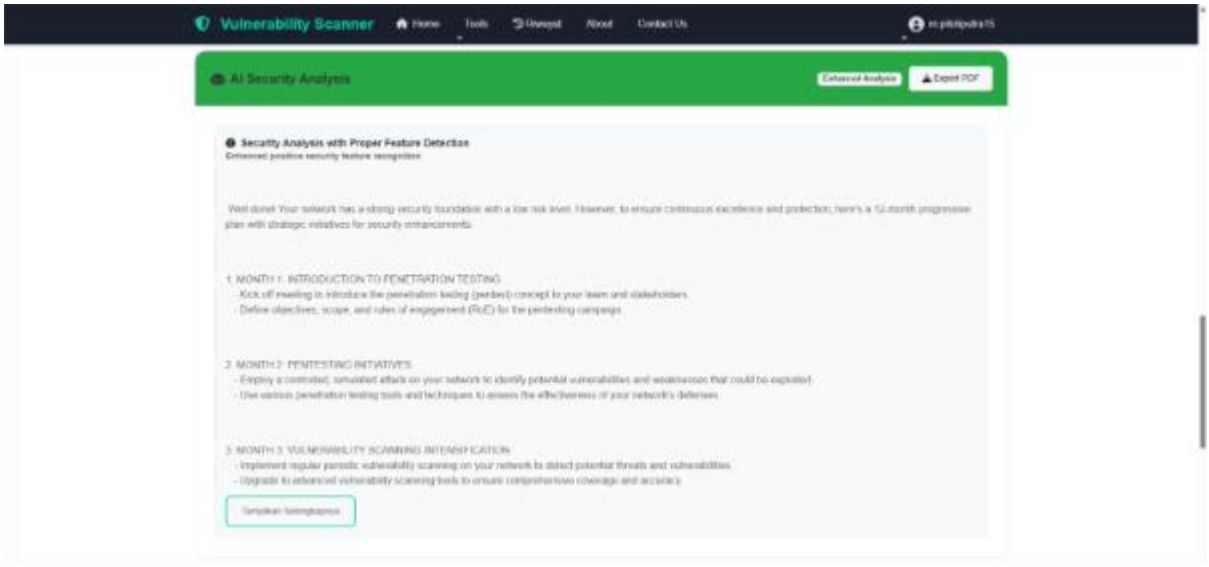

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Security Scan: Hasil dengan deteksi positive security feature yang sudah diperbaiki
Summary
Host Information
Target: localhost:8081
Tool: NMAP
Scan Type: Comprehensive
Status:
Risk Level:
Enhancement:
Security Summary
Critical: 4
High: 1
Medium: 5
Low: 2
Info: 4
Score:
Kembali ke Riwayat
Ekspor Rekomendasi PDF
Hapus Hasil
Vulnerability Scanner
Tool canggih untuk memindai kerentanan pada sistem dan jaringan Anda
Quick Links
Home
MWP Scanner
NIKTO Scanner
Riwayat Scan
Contact
Bandung, Indonesia

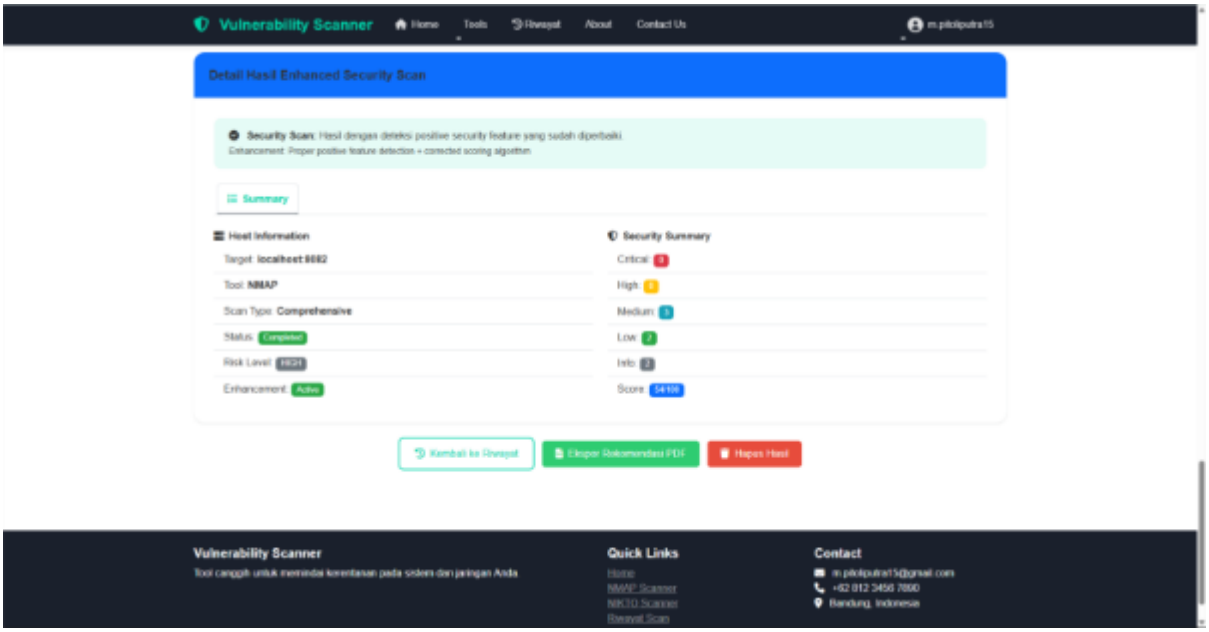
- *Website* 2 (localhost:8082)



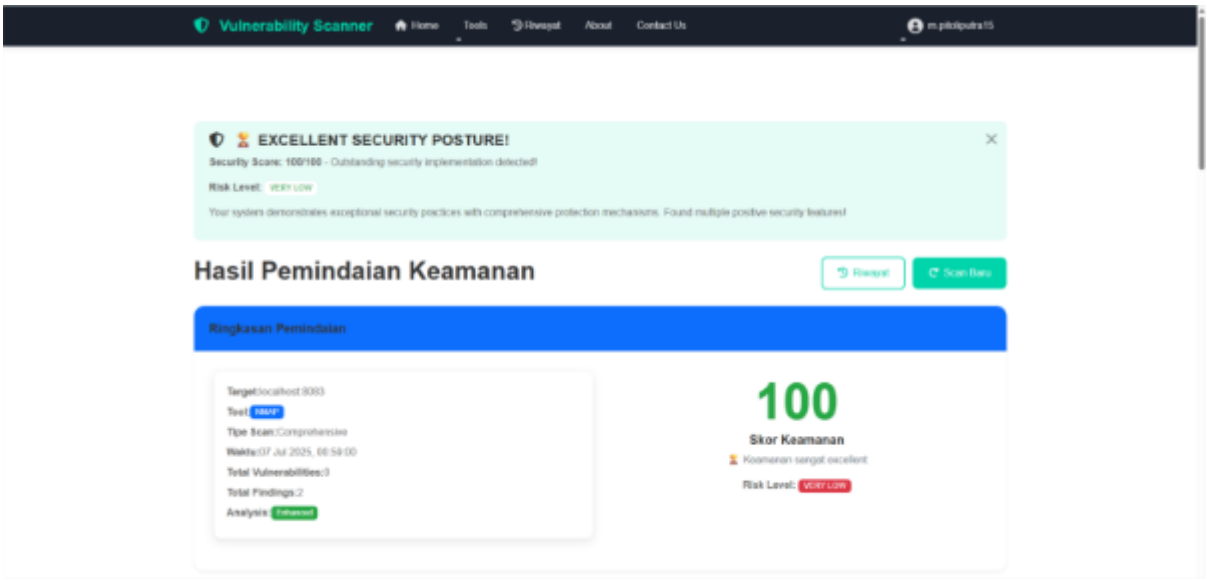


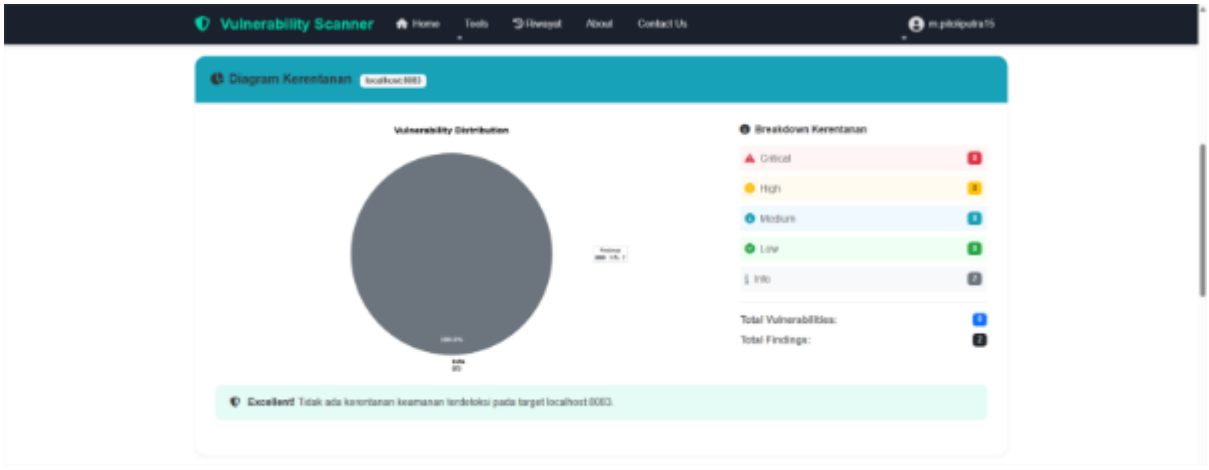


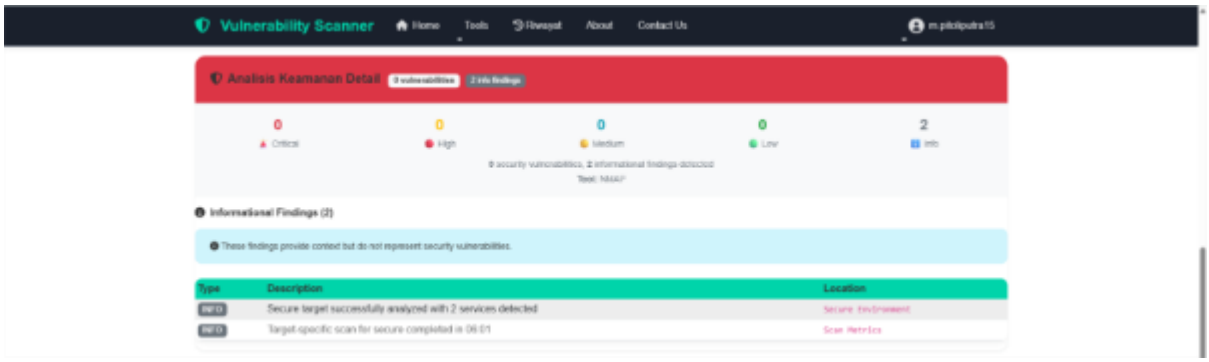

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
m.pikiputra15
AI Security Analysis
Enhanced Analysis
Export PDF
Security Analysis with Proper Feature Detection
Enhanced positive security feature recognition
1. MONTH 1: INTRODUCTION TO PENETRATION TESTING
- Kick off meeting to introduce the penetration testing (pentest) concept to your team and stakeholders.
- Define objectives, scope, and rules of engagement (RoE) for the pentesting campaign.
2. MONTH 2: PENTESTING INITIATIVES
- Employ a controlled, simulated attack on your network to identify potential vulnerabilities and weaknesses that could be exploited.
- Use various penetration testing tools and techniques to assess the effectiveness of your network's defenses.
3. MONTH 3: VULNERABILITY SCANNING INTENSIFICATION
- Implement regular periodic vulnerability scanning on your network to detect potential threats and vulnerabilities.
- Upgrade to advanced vulnerability scanning tools to ensure comprehensive coverage and accuracy.

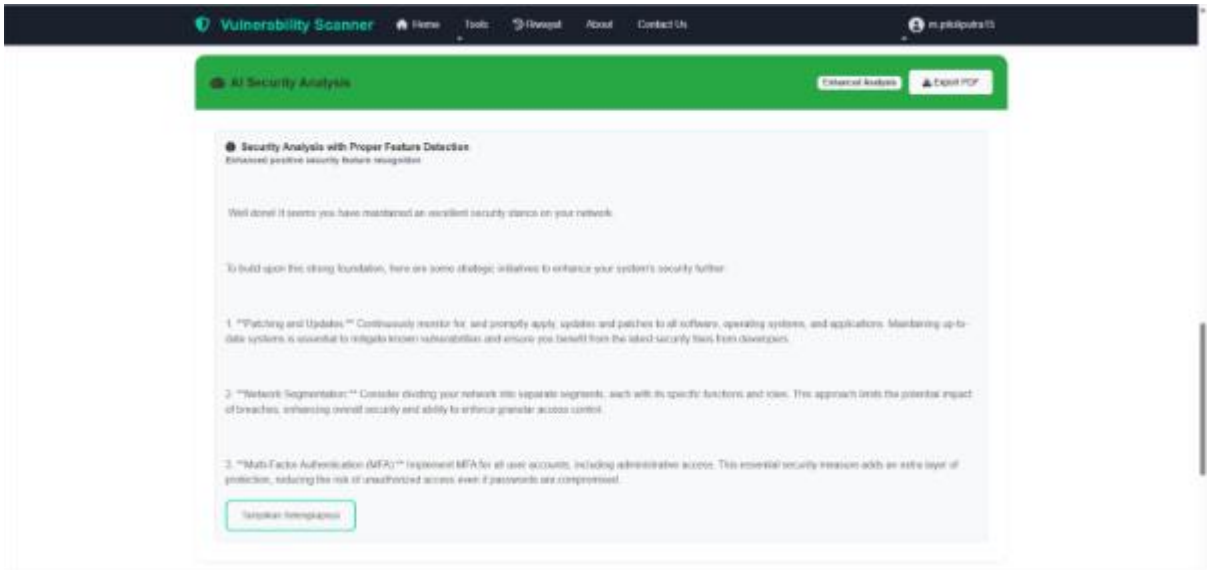

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
m.pikiputra15
Detail Hasil Enhanced Security Scan
Security Scan: Hasil dengan deteksi positive security feature yang sudah diperbaiki
Enhancement: Proper positive feature detection + corrected scoring algorithm
Summary
Host Information
Target: localhost:8082
Tool: NMAP
Scan Type: Comprehensive
Status: Completed
Risk Level: MEDIUM
Enhancement: Active
Security Summary
Critical: 0
High: 0
Medium: 1
Low: 2
Info: 3
Score: 54/100
Kembali ke Riwayat
Ekspor Rekomendasi PDF
Hapus Hasil
Vulnerability Scanner
Tool canggih untuk memindai kerentanan pada sistem dan jaringan Anda
Quick Links
Home
NMAP Scanner
NIKTO Scanner
Riwayat Scan
Contact
m.pikiputra15@gmail.com
+62 812 3456 7890
Bandung, Indonesia

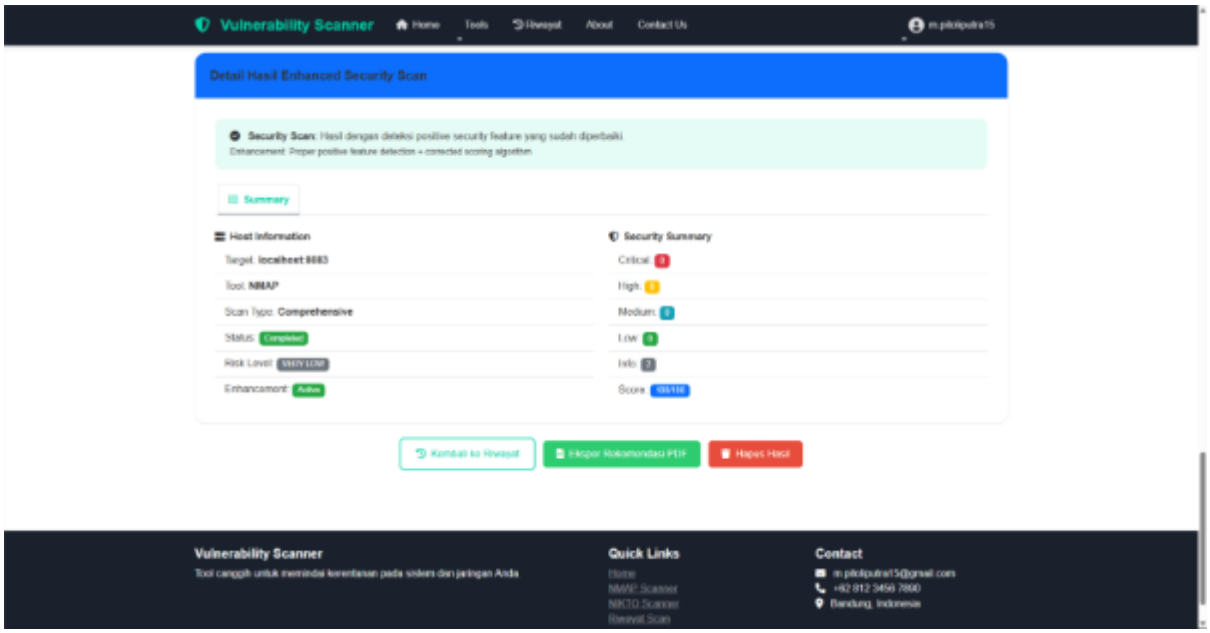
- *Website* 3 (localhost:8083)





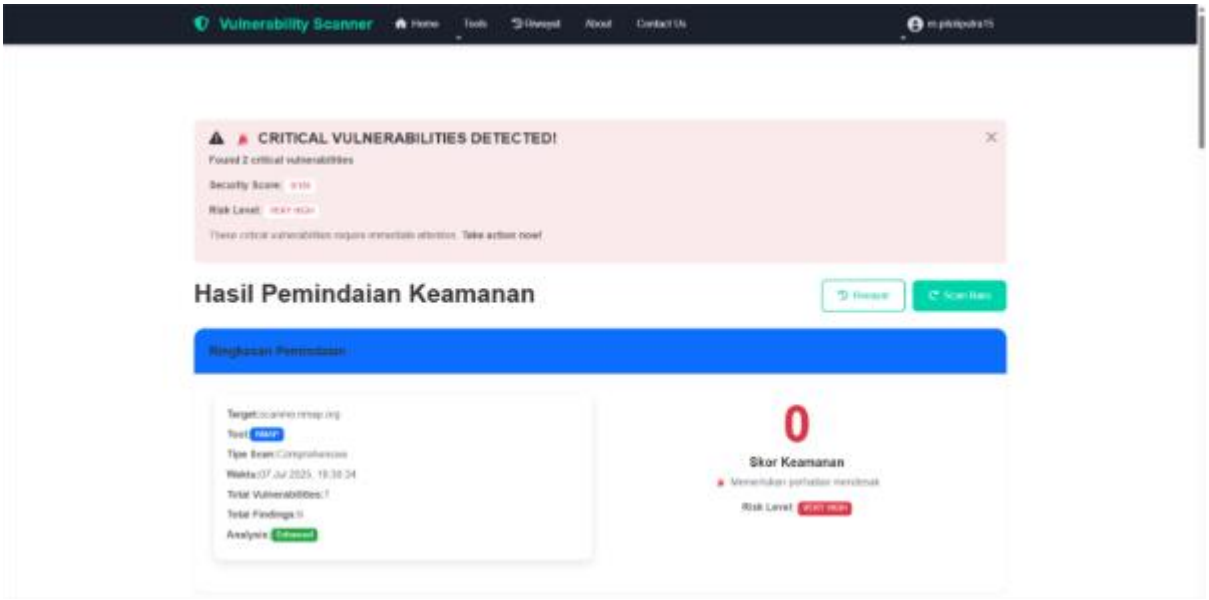


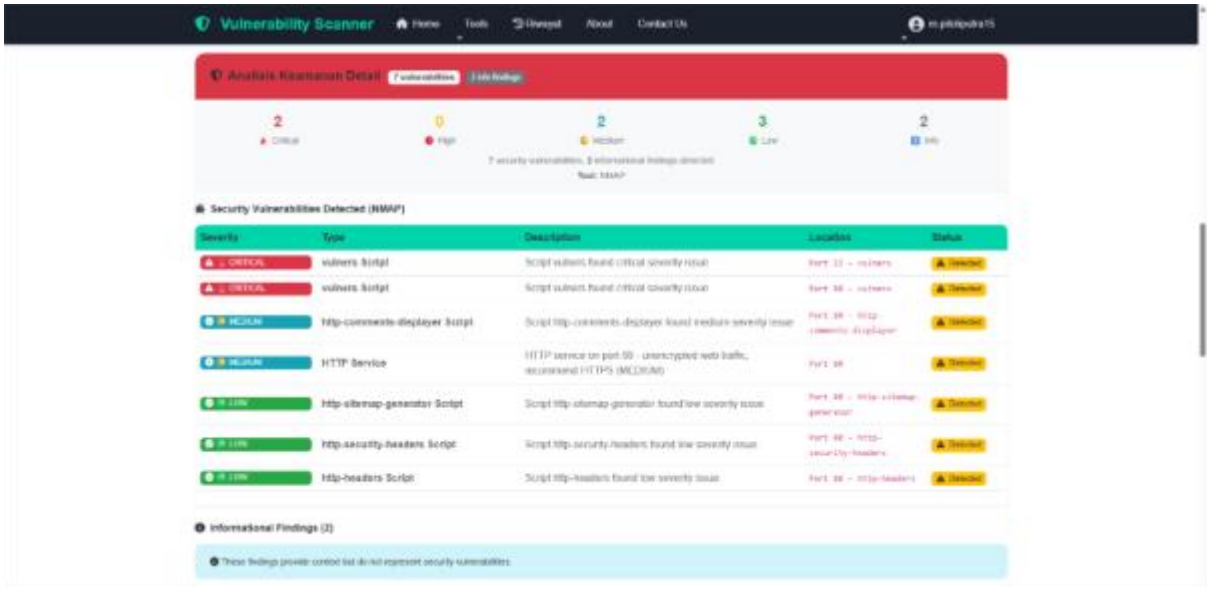

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Security Analysis with Proper Feature Detection

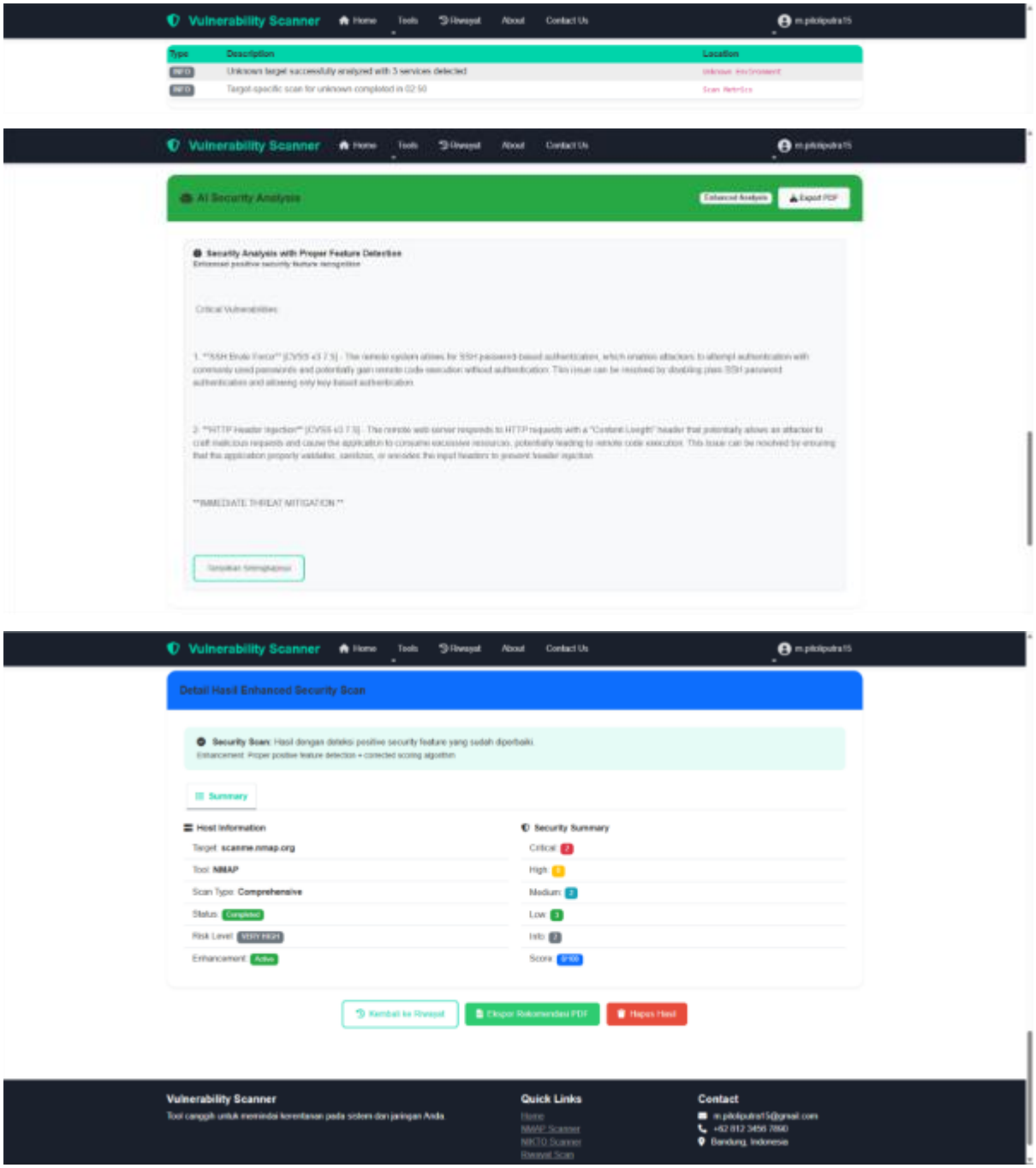

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Security Summary
Vulnerability Scanner
Quick Links
Contact

f. Pengujian (*tools* Nmap) *Website* internet

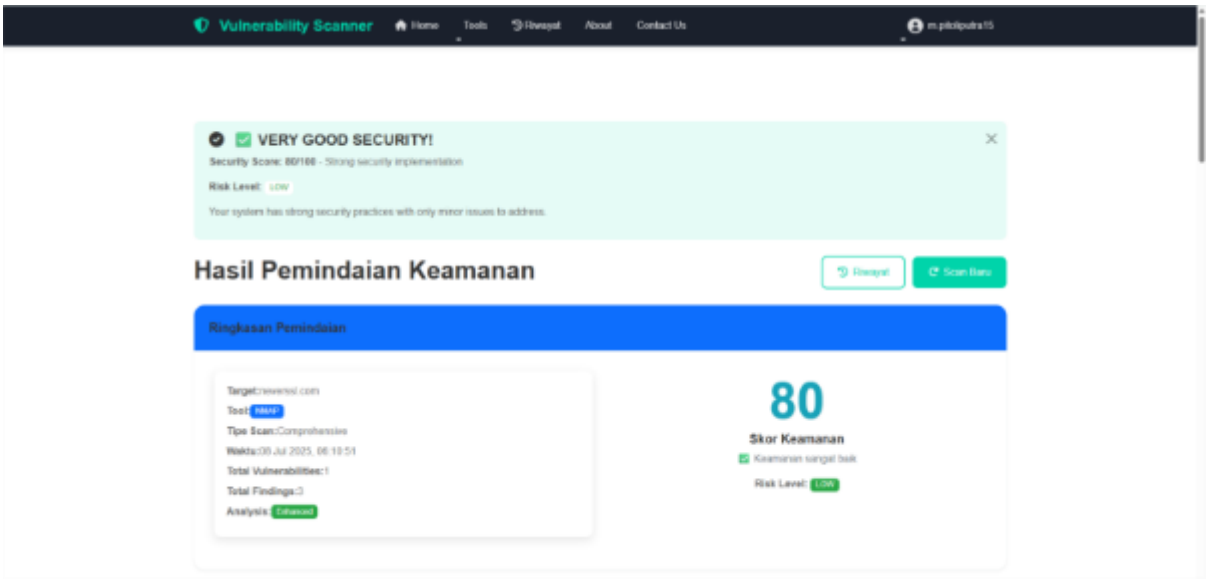
- *Website* 1 (scanme.nmap.org)

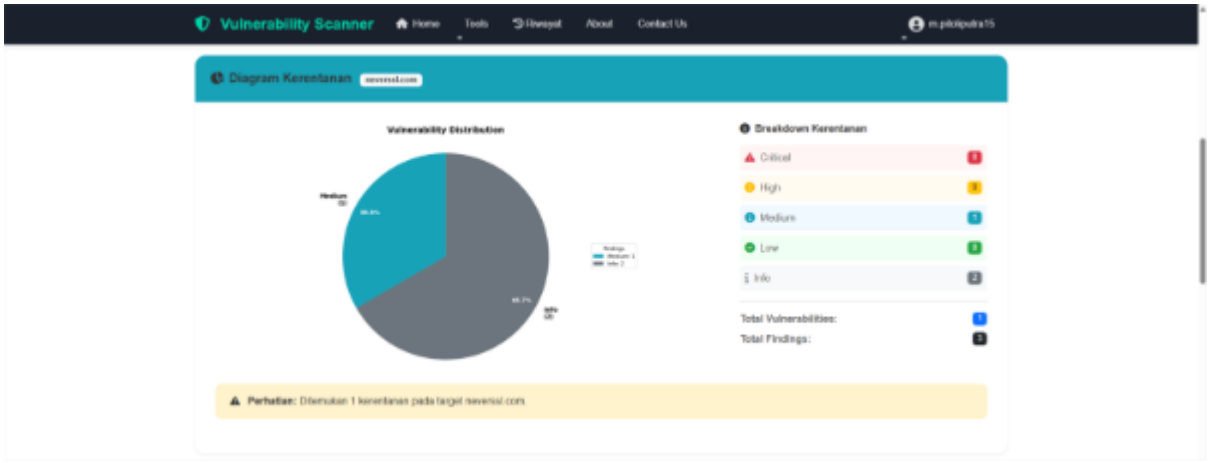


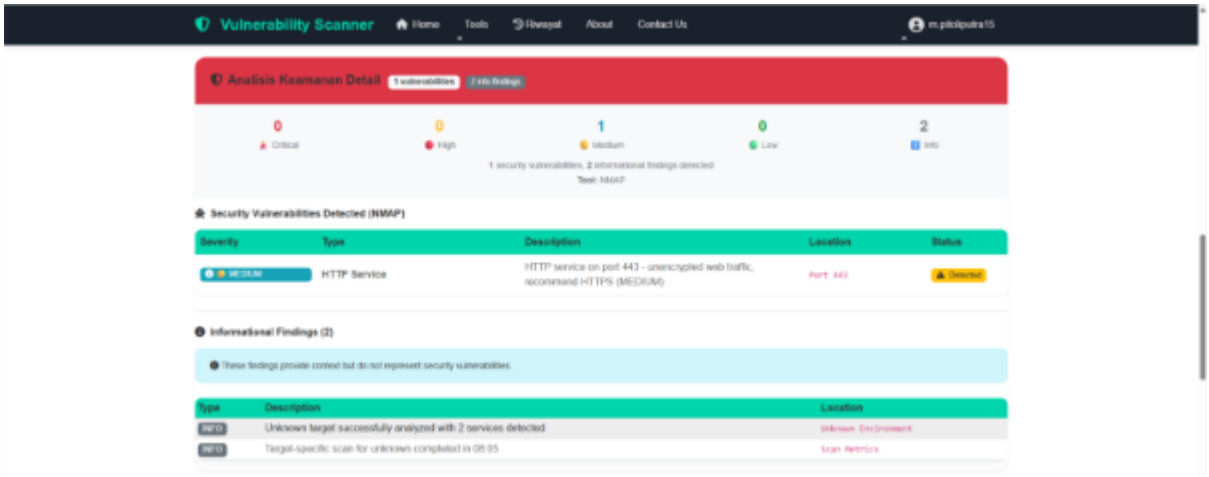


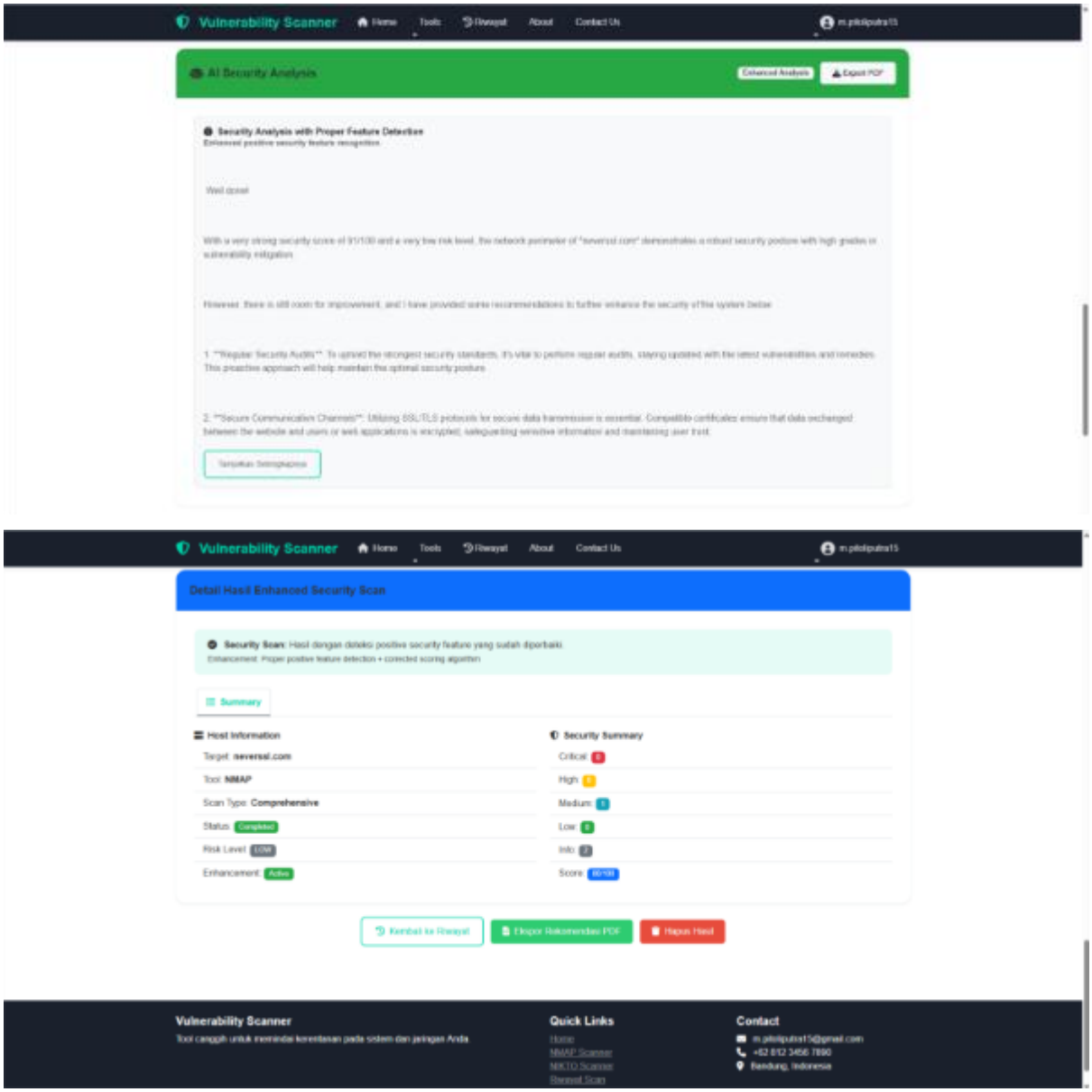

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Type
Description
Location
INFO
Unknown target successfully analyzed with 5 services detected
INFO
Target-specific scan for unknown completed in 02:50
Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Enhanced Analysis
Export PDF
Security Analysis with Proper Feature Detection
Critical Vulnerabilities
1. **SSH Brute Force** (CVSS v3.7.5) - The remote system allows for SSH password-based authentication, which enables attackers to attempt authentication with commonly used passwords and potentially gain remote code execution without authentication. This issue can be resolved by disabling plain SSH password authentication and allowing only key-based authentication.
2. **HTTP Header Injection** (CVSS v3.7.5) - The remote web server responds to HTTP requests with a "Content-Length" header that potentially allows an attacker to craft malicious requests and cause the application to consume excessive resources, potentially leading to remote code execution. This issue can be resolved by ensuring that the application properly validates, sanitizes, or encodes the input headers to prevent header injection.
**IMMEDIATE THREAT MITIGATION **
Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Security Scan: Hasil dengan deteksi positive security feature yang sudah diperbaiki
Enhancement: Proper positive feature detection + corrected scoring algorithm
Summary
Host Information
Target: scanme.nmap.org
Tool: NMAP
Scan Type: Comprehensive
Status: Completed
Risk Level: VERY HIGH
Enhancement: Active
Security Summary
Critical: 2
High: 1
Medium: 2
Low: 1
Info: 2
Score: 9/100
Kembali ke Riwayat
Ekspor Rekomendasi PDF
Hapus Hasil
Vulnerability Scanner
Tool canggih untuk memindai kerentanan pada sistem dan jaringan Anda
Quick Links
Home
NMAP Scanner
NIKTO Scanner
Riwayat Scan
Contact
Bandung, Indonesia

- *Website* 2 (neverssl.com)

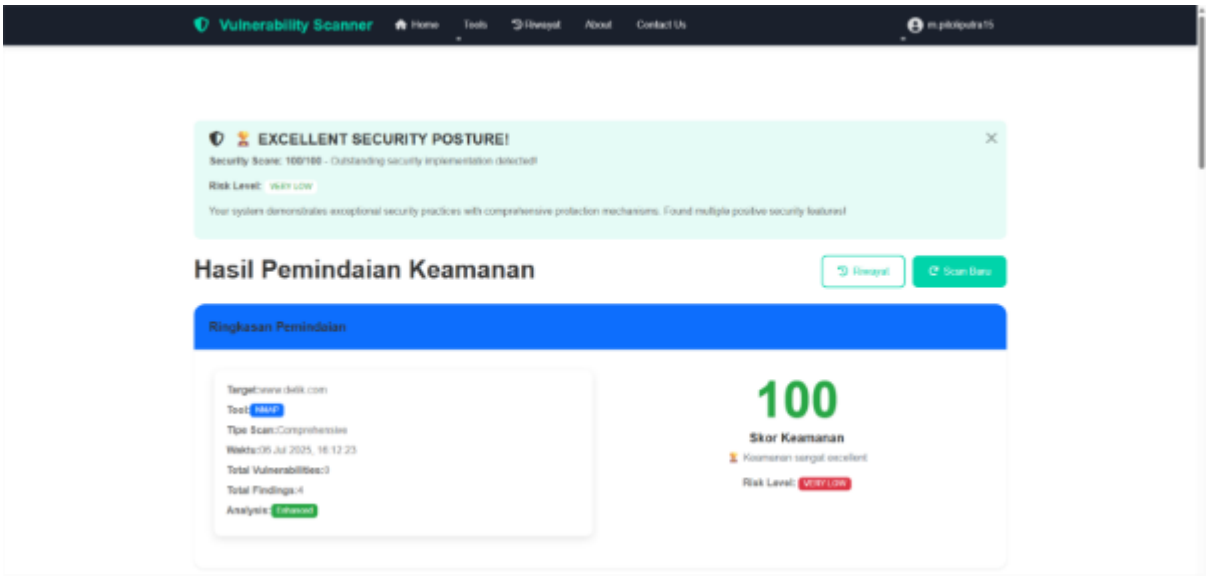


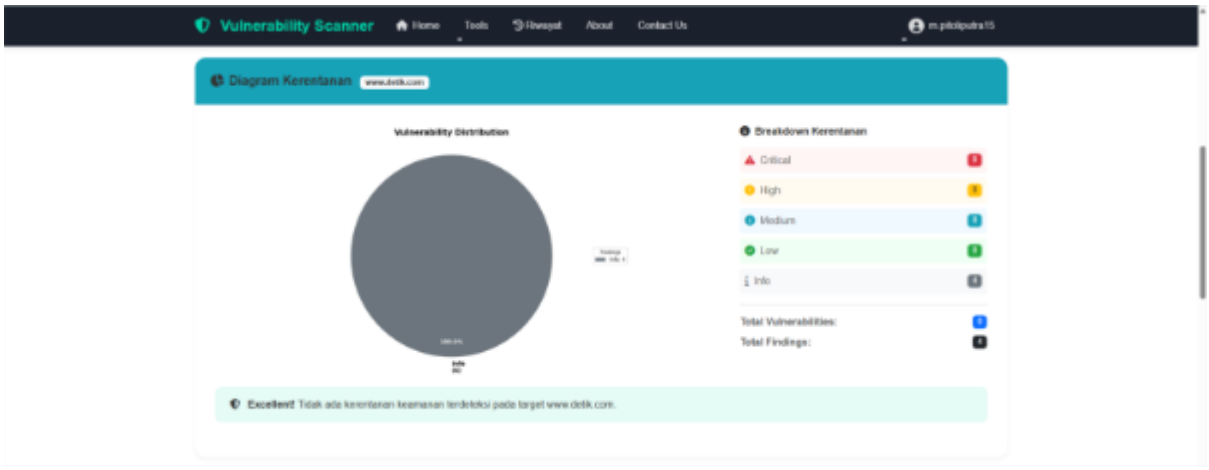


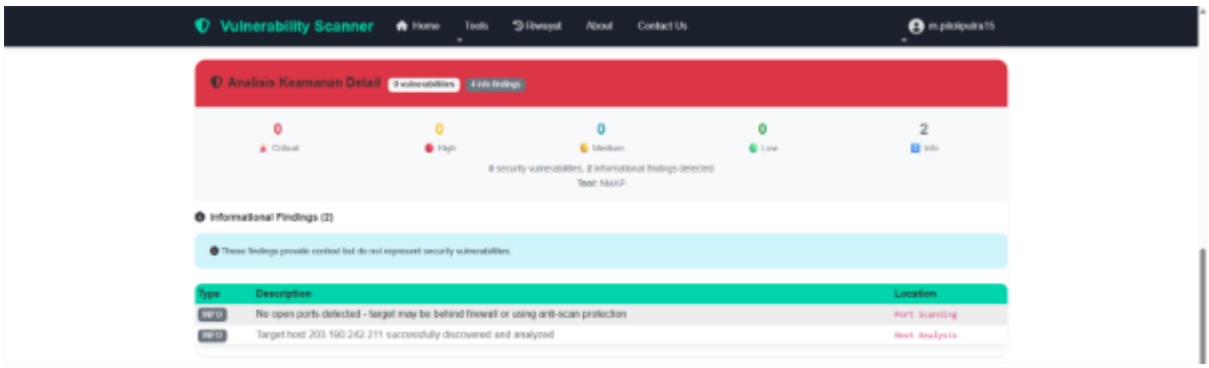


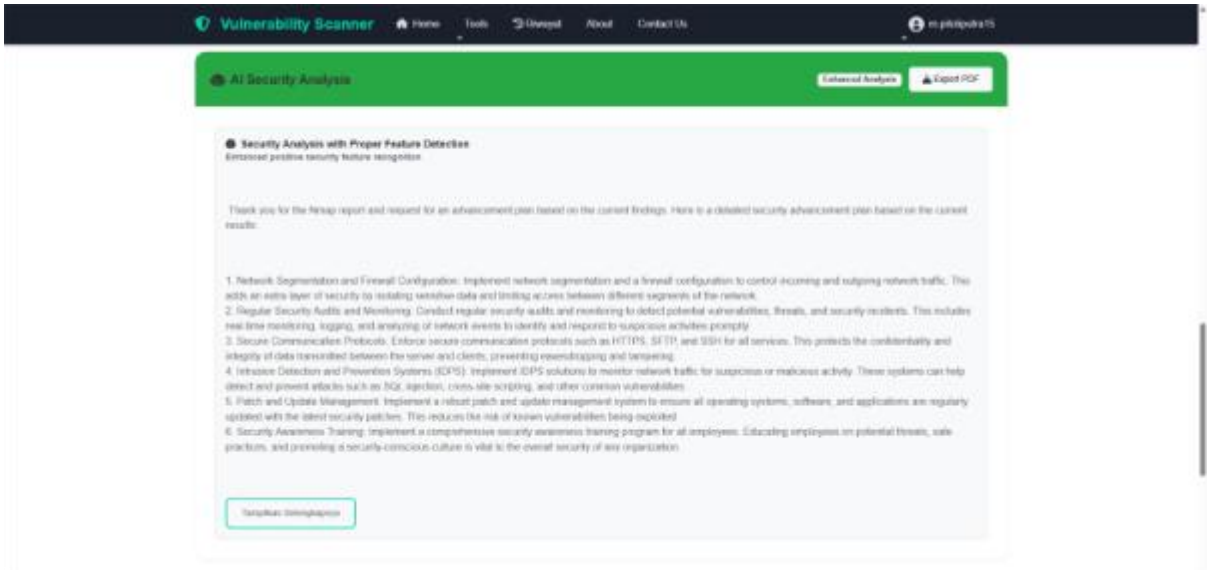

Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
m.pikiliputra15
AI Security Analysis
Enhanced Analysis
Export PDF
Security Analysis with Proper Feature Detection
Enhanced positive security feature recognition
Well done!
With a very strong security score of 91/100 and a very low risk level, the network perimeter of "reversal.com" demonstrates a robust security posture with high grades in vulnerability mitigation.
However, there is still room for improvement, and I have provided some recommendations to further enhance the security of the system below:
1. **Regular Security Audits**: To uphold the strongest security standards, it's vital to perform regular audits, staying updated with the latest vulnerabilities and remedies. This proactive approach will help maintain the optimal security posture.
2. **Secure Communication Channels**: Utilizing SSL/TLS protocols for secure data transmission is essential. Compatible certificates ensure that data exchanged between the website and users or web applications is encrypted, safeguarding sensitive information and maintaining user trust.
Simpulkan Strategisnya
Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
m.pikiliputra15
Detail Hasil Enhanced Security Scan
Security Scan: Hasil dengan deteksi positive security feature yang sudah diperbaiki
Enhancement: Proper positive feature detection + corrected scoring algorithm
Summary
Host Information
Target: reversal.com
Tool: NMAP
Scan Type: Comprehensive
Status: Completed
Risk Level: LOW
Enhancement: Active
Security Summary
Critical: 0
High: 1
Medium: 0
Low: 0
Info: 7
Score: 85/100
Kembali ke Riwayat
Ekspor Rekomendasi PDF
Hapus Hasil
Vulnerability Scanner
Tool canggih untuk meninjau kerentanan pada sistem dan jaringan Anda
Quick Links
Home
NMAP Scanner
MIKTO Scanner
Riwayat Scan
Contact
m.pikiliputra15@gmail.com
+62 812 3456 7890
Bandung, Indonesia

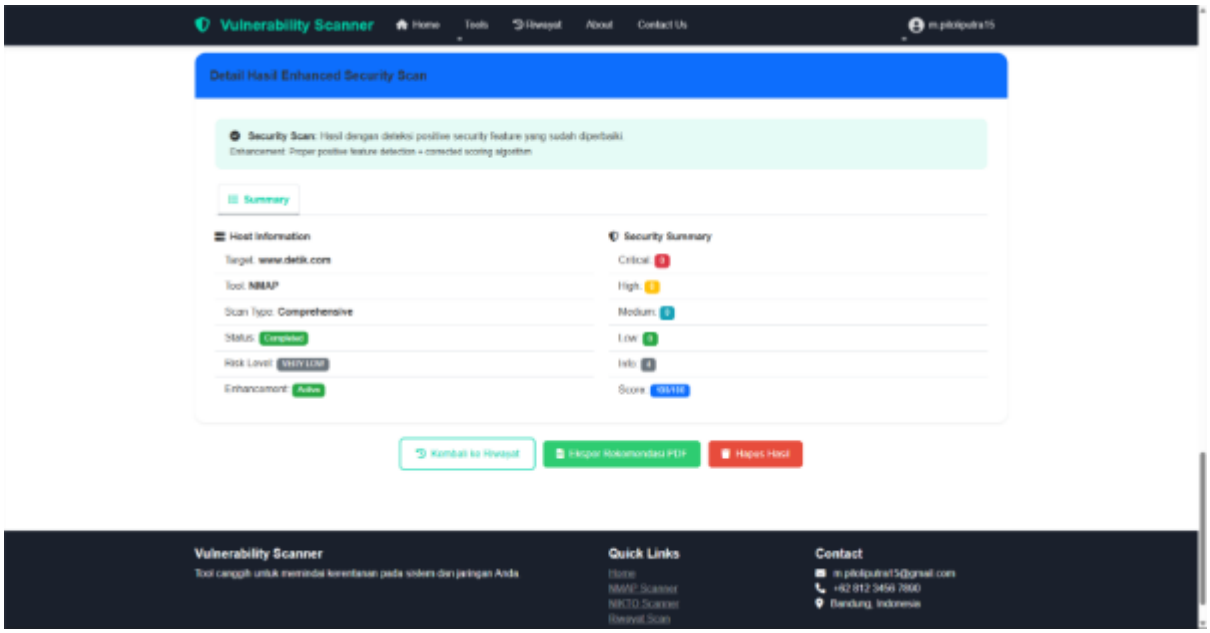
- *Website* 3 (www.detik.com)





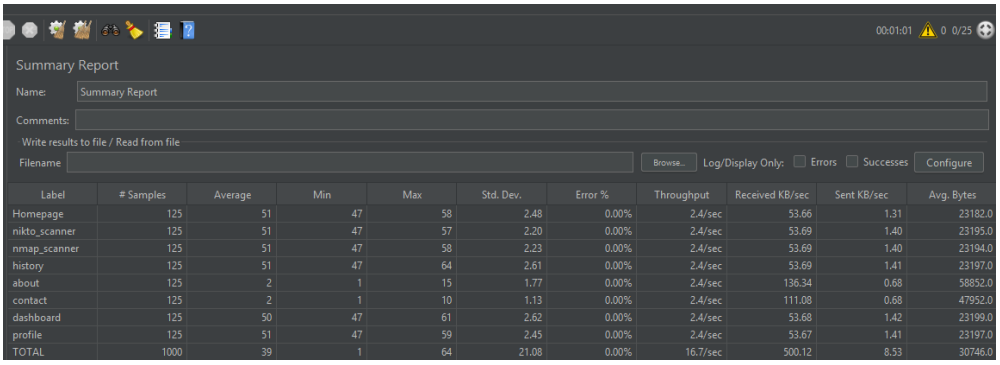



Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
AI Security Analysis
Export PDF
Security Analysis with Proper Feature Detection


Vulnerability Scanner
Home
Tools
Riwayat
About
Contact Us
Detail Hasil Enhanced Security Scan
Summary
Host Information
Target: www.detik.com
Tool: NMAP
Scan Type: Comprehensive
Status: Completed
Enhancement: Active
Security Summary
Critical: 3
High: 1
Medium: 0
Low: 0
Info: 4
Score: 65/100
Kembali ke Riwayat
Hapus Hasil
Vulnerability Scanner
Tool canggih untuk mendeteksi kerentanan pada sistem dan jaringan Anda
Quick Links
Home
NMAP Scanner
NIKTO Scanner
Riwayat Scan
Contact
Bandung, Indonesia

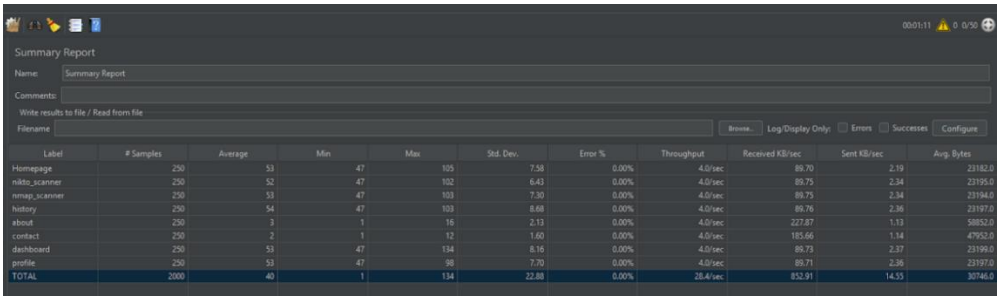
g. Pengujian menggunakan apache JMeter

- Pengujian 1



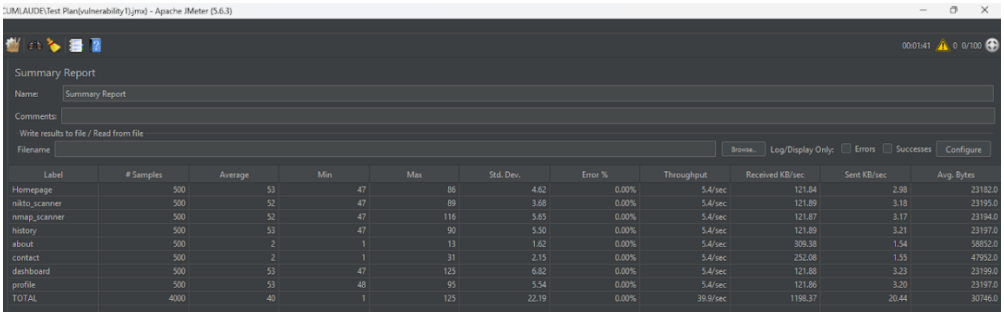
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Homepage	125	51	47	58	2.48	0.00%	2.4/sec	53.66	1.31	23182.0
nikto_scanner	125	51	47	57	2.20	0.00%	2.4/sec	53.69	1.40	23195.0
nmap_scanner	125	51	47	58	2.23	0.00%	2.4/sec	53.69	1.40	23194.0
history	125	51	47	64	2.61	0.00%	2.4/sec	53.69	1.41	23197.0
about	125	2	1	15	1.77	0.00%	2.4/sec	136.34	0.68	58852.0
contact	125	2	1	10	1.13	0.00%	2.4/sec	111.08	0.68	47952.0
dashboard	125	50	47	61	2.62	0.00%	2.4/sec	53.68	1.42	23199.0
profile	125	51	47	59	2.45	0.00%	2.4/sec	53.67	1.41	23197.0
TOTAL	1000	39	1	64	21.08	0.00%	16.7/sec	500.12	8.53	30746.0

- Pengujian 2



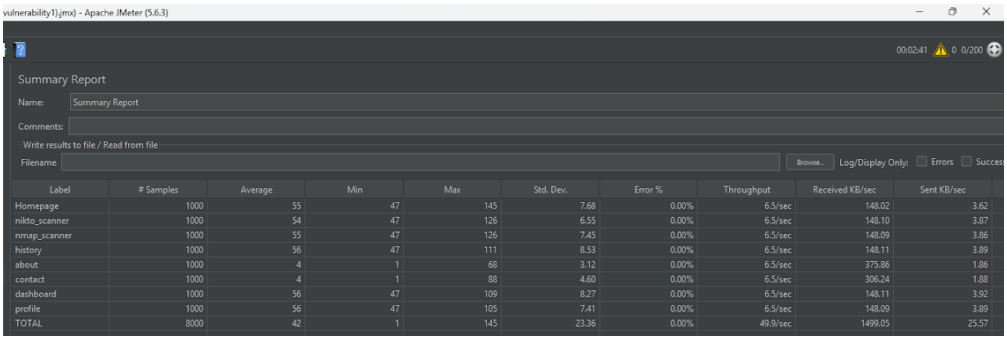
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Homepage	250	53	47	105	7.58	0.00%	4.0/sec	89.70	2.19	23182.0
nikto_scanner	250	52	47	102	6.43	0.00%	4.0/sec	89.75	2.34	23195.0
nmap_scanner	250	53	47	103	7.30	0.00%	4.0/sec	89.75	2.34	23194.0
history	250	54	47	103	8.68	0.00%	4.0/sec	89.76	2.36	23197.0
about	250	3	1	16	2.13	0.00%	4.0/sec	227.87	1.13	58852.0
contact	250	2	1	12	1.60	0.00%	4.0/sec	185.66	1.14	47952.0
dashboard	250	53	47	134	8.16	0.00%	4.0/sec	89.73	2.37	23199.0
profile	250	53	47	98	7.70	0.00%	4.0/sec	89.71	2.36	23197.0
TOTAL	2000	40	1	134	22.88	0.00%	28.4/sec	852.91	14.55	30746.0

- Pengujian 3



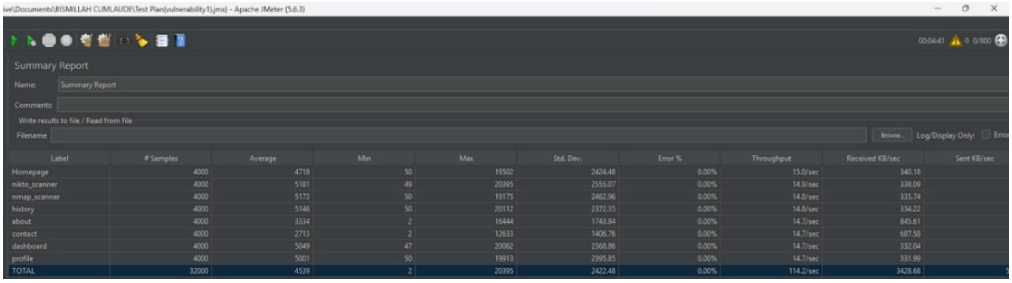
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Homepage	500	53	47	86	4.62	0.00%	5.4/sec	121.84	2.98	23182.0
nikto_scanner	500	52	47	89	3.68	0.00%	5.4/sec	121.89	3.18	23195.0
nmap_scanner	500	52	47	116	5.65	0.00%	5.4/sec	121.87	3.17	23194.0
history	500	53	47	90	5.50	0.00%	5.4/sec	121.89	3.21	23197.0
about	500	2	1	13	1.62	0.00%	5.4/sec	309.38	1.54	58852.0
contact	500	2	1	31	2.15	0.00%	5.4/sec	252.08	1.55	47952.0
dashboard	500	53	47	125	6.82	0.00%	5.4/sec	121.88	3.23	23199.0
profile	500	53	48	95	5.54	0.00%	5.4/sec	121.86	3.20	23197.0
TOTAL	4000	40	1	125	22.19	0.00%	39.9/sec	1198.37	20.44	30746.0

- Pengujian 4

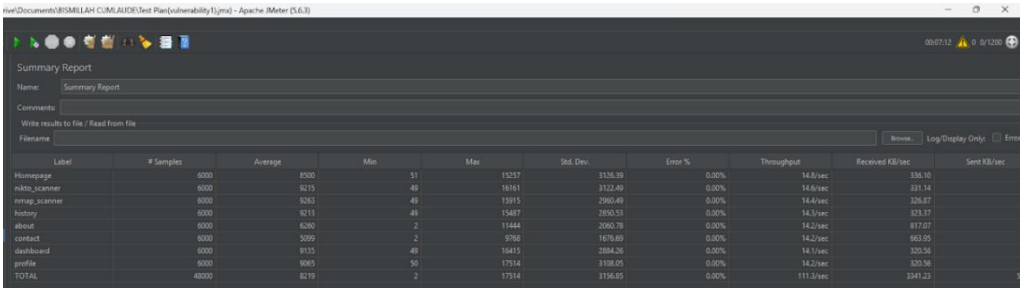


Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec
Homepage	1000	55	47	145	7.68	0.00%	6.5/sec	148.02	3.62
nikto_scanner	1000	54	47	126	6.55	0.00%	6.5/sec	148.10	3.87
nmap_scanner	1000	55	47	126	7.45	0.00%	6.5/sec	148.09	3.86
history	1000	56	47	111	8.53	0.00%	6.5/sec	148.11	3.89
about	1000	4	1	68	3.12	0.00%	6.5/sec	375.86	1.86
contact	1000	4	1	88	4.60	0.00%	6.5/sec	306.24	1.88
dashboard	1000	56	47	109	8.27	0.00%	6.5/sec	148.11	3.92
profile	1000	56	47	105	7.41	0.00%	6.5/sec	148.09	3.89
TOTAL	8000	42	1	145	23.36	0.00%	49.9/sec	1499.05	25.57

- Pengujian 5

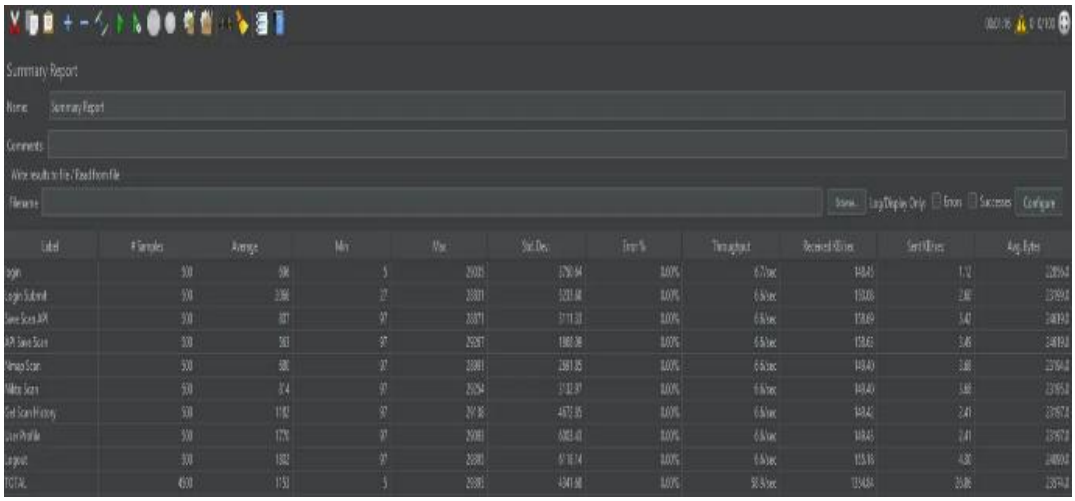


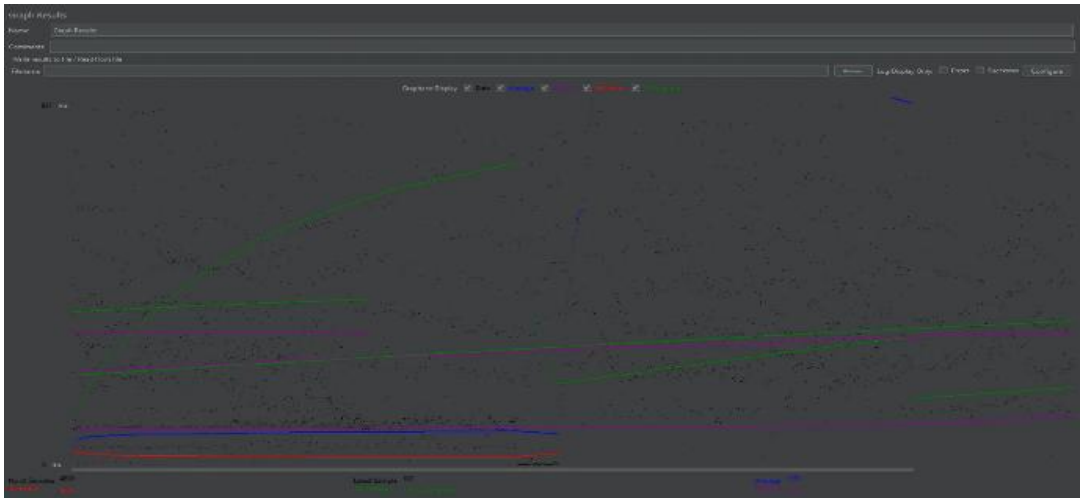
- Pengujian 6



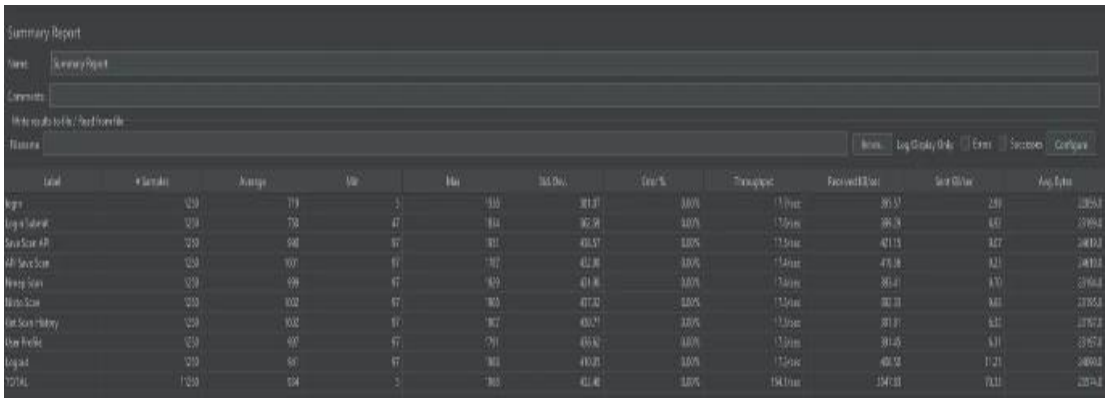
h. Pengujian Backend

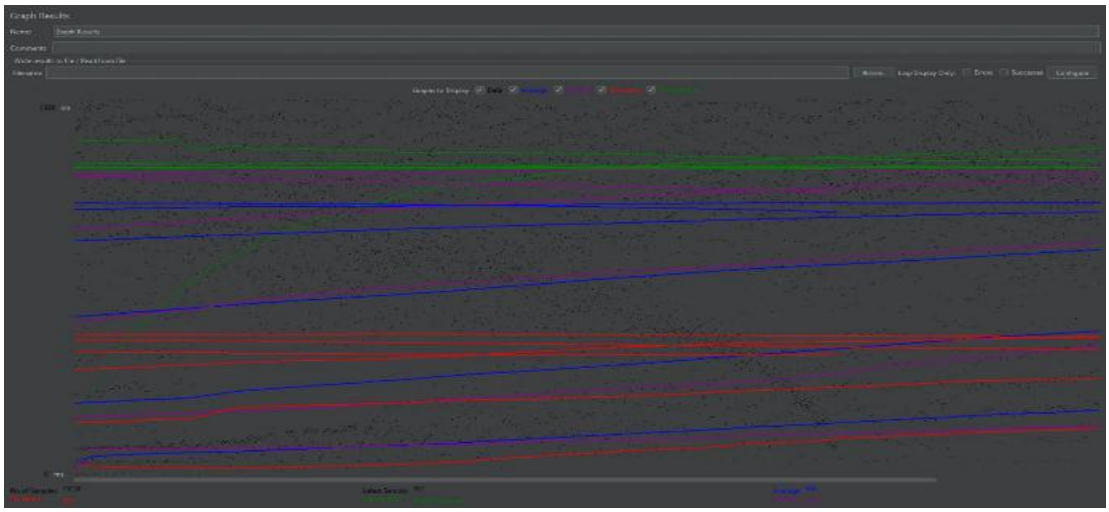
- Pengujian 1



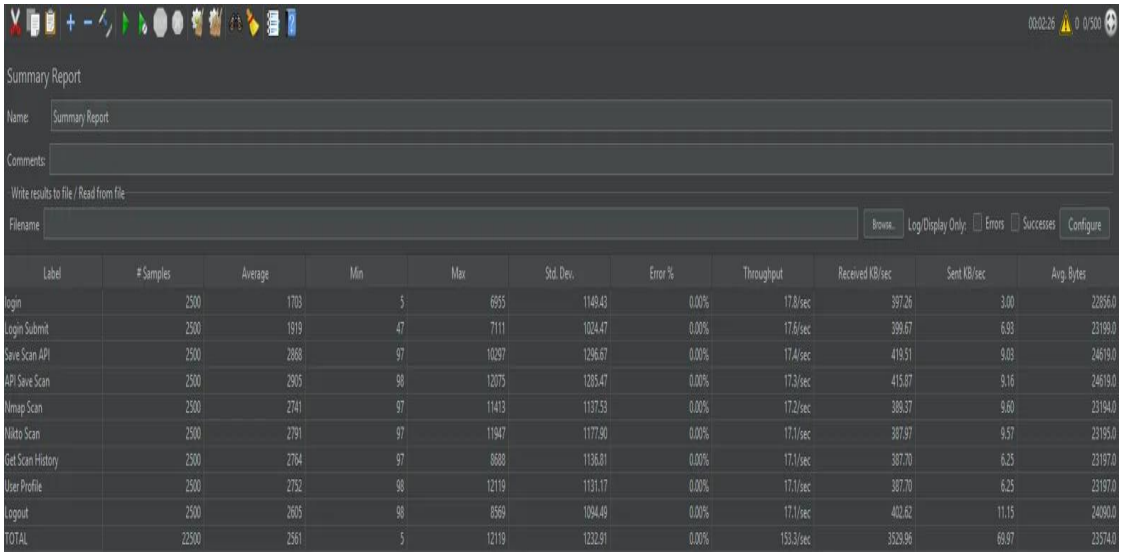


- Pengujian 2

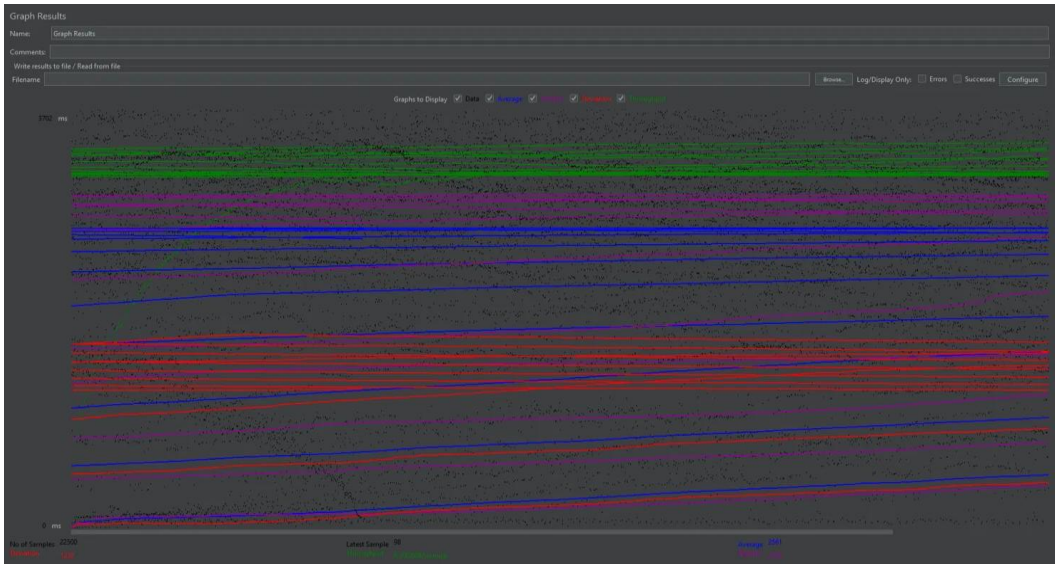




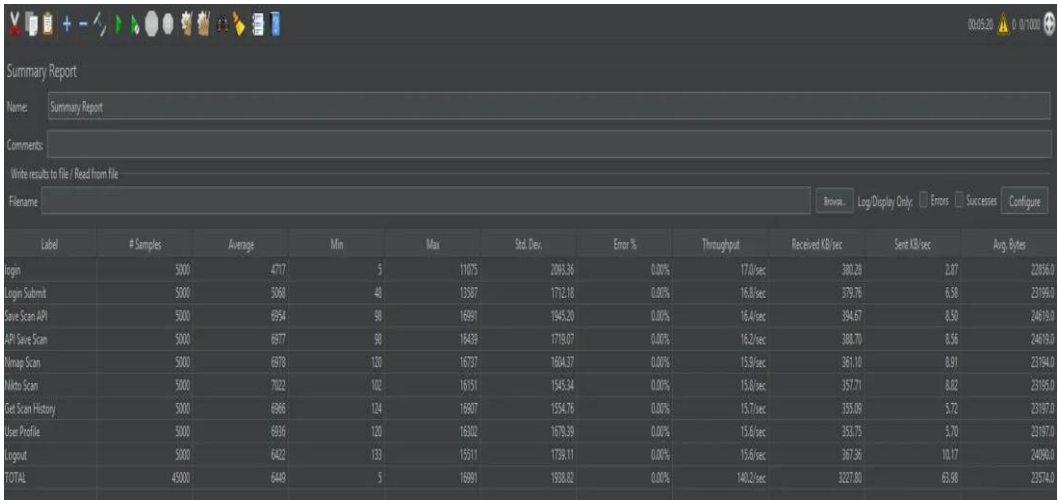
- Pengujian 3



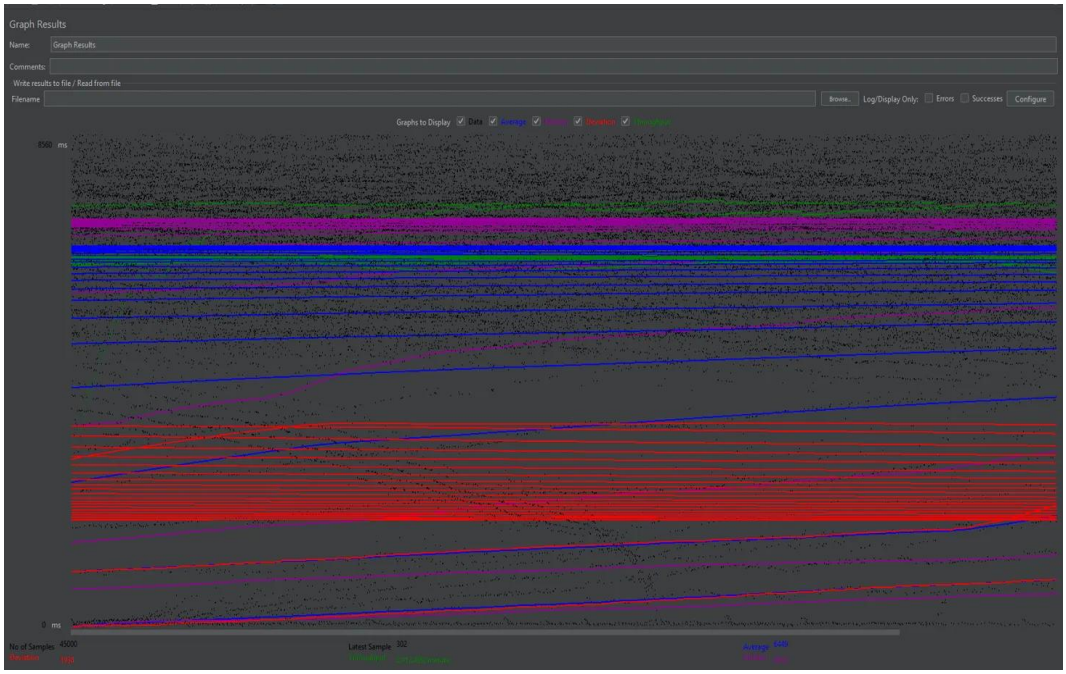
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
login	2500	1703	5	6955	1149.43	0.00%	17.8/sec	397.26	3.00	22856.0
Login Submit	2500	1919	47	7111	1024.47	0.00%	17.6/sec	399.67	6.93	23199.0
Save Scan API	2500	2868	97	10297	1296.67	0.00%	17.4/sec	419.51	9.03	24619.0
API Save Scan	2500	2905	98	12075	1285.47	0.00%	17.3/sec	415.87	9.16	24619.0
Nmap Scan	2500	2741	97	11413	1137.53	0.00%	17.2/sec	388.37	9.60	23194.0
Nikto Scan	2500	2791	97	11947	1177.90	0.00%	17.1/sec	387.97	9.57	23195.0
Get Scan History	2500	2764	97	8688	1136.81	0.00%	17.1/sec	387.70	6.25	23197.0
User Profile	2500	2752	98	12119	1131.17	0.00%	17.1/sec	387.70	6.25	23197.0
Logout	2500	2605	98	8569	1094.49	0.00%	17.1/sec	402.62	11.15	24090.0
TOTAL	22500	2561	5	12119	1232.91	0.00%	153.3/sec	3529.96	69.97	23574.0



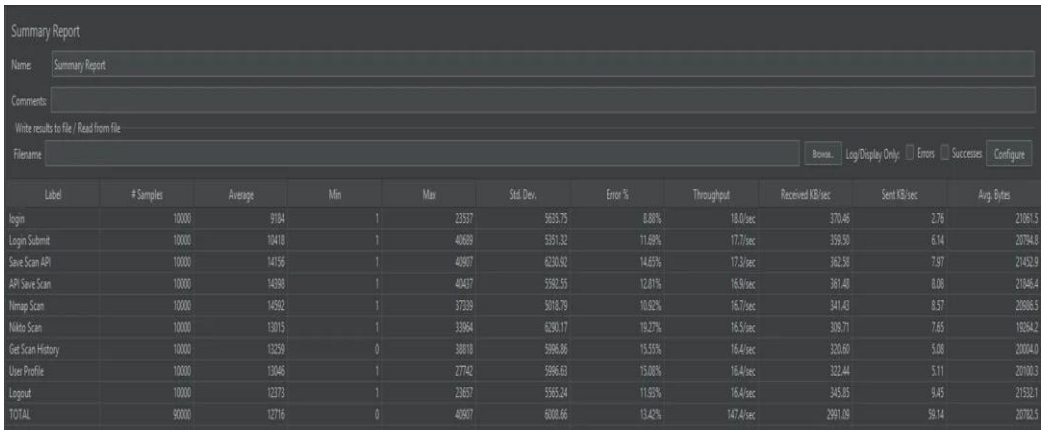
- Pengujian 4



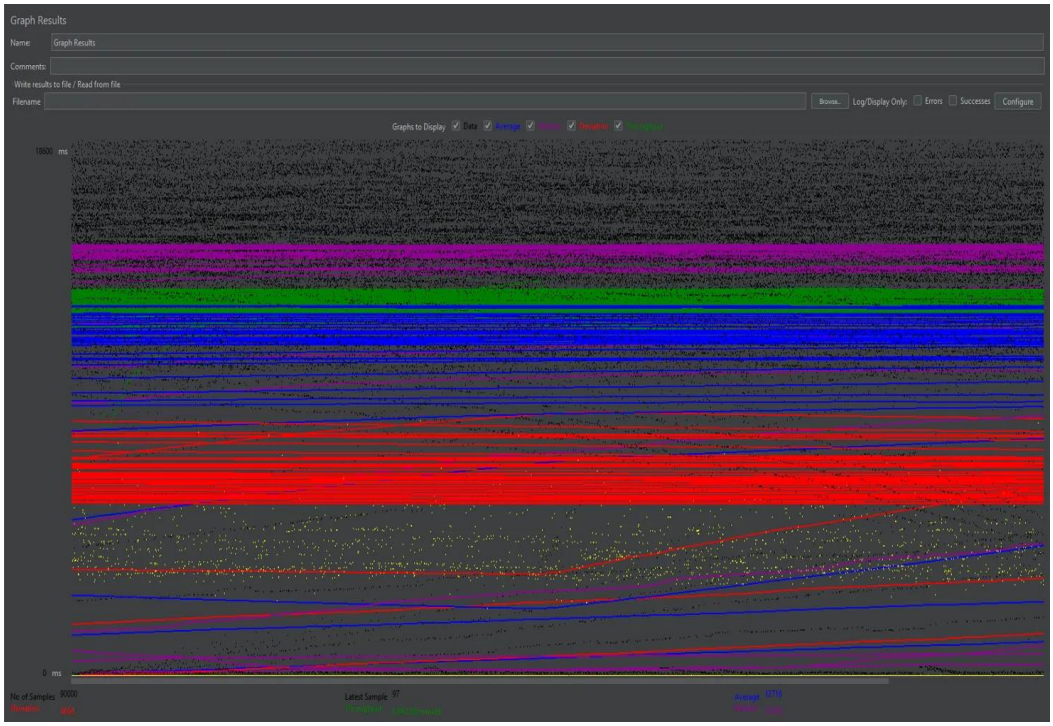
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
login	5000	4717	5	11075	2093.36	0.00%	17.0/sec	380.28	2.87	22856.0
Login Submit	5000	5068	48	13587	1712.18	0.00%	16.8/sec	379.76	6.58	23199.0
Save Scan API	5000	6954	98	16991	1945.20	0.00%	16.4/sec	394.67	8.50	24619.0
API Save Scan	5000	6977	98	16439	1719.07	0.00%	16.2/sec	388.70	8.56	24619.0
Nmap Scan	5000	6978	120	16737	1804.37	0.00%	15.9/sec	361.10	8.91	23194.0
Nikto Scan	5000	7022	102	16151	1545.34	0.00%	15.8/sec	357.71	8.82	23195.0
Get Scan History	5000	6966	124	16907	1554.76	0.00%	15.7/sec	355.09	5.72	23197.0
User Profile	5000	6936	120	16302	1679.39	0.00%	15.6/sec	353.75	5.70	23197.0
Logout	5000	6422	133	15511	1739.11	0.00%	15.6/sec	367.36	10.17	24090.0
TOTAL	45000	6449	5	16991	1938.82	0.00%	140.2/sec	3227.80	63.98	23574.0



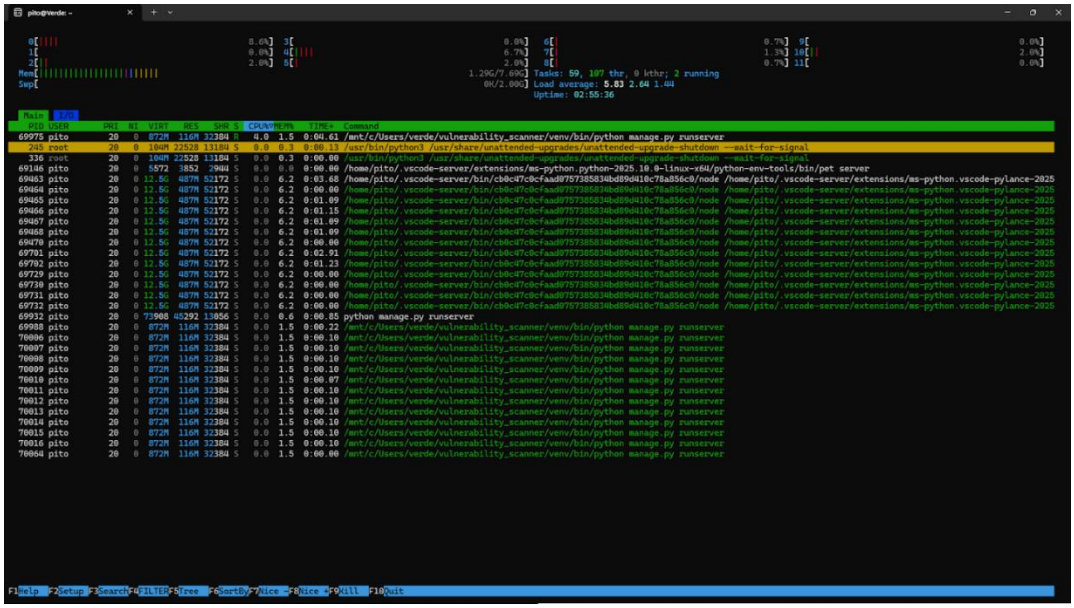
- Pengujian 5

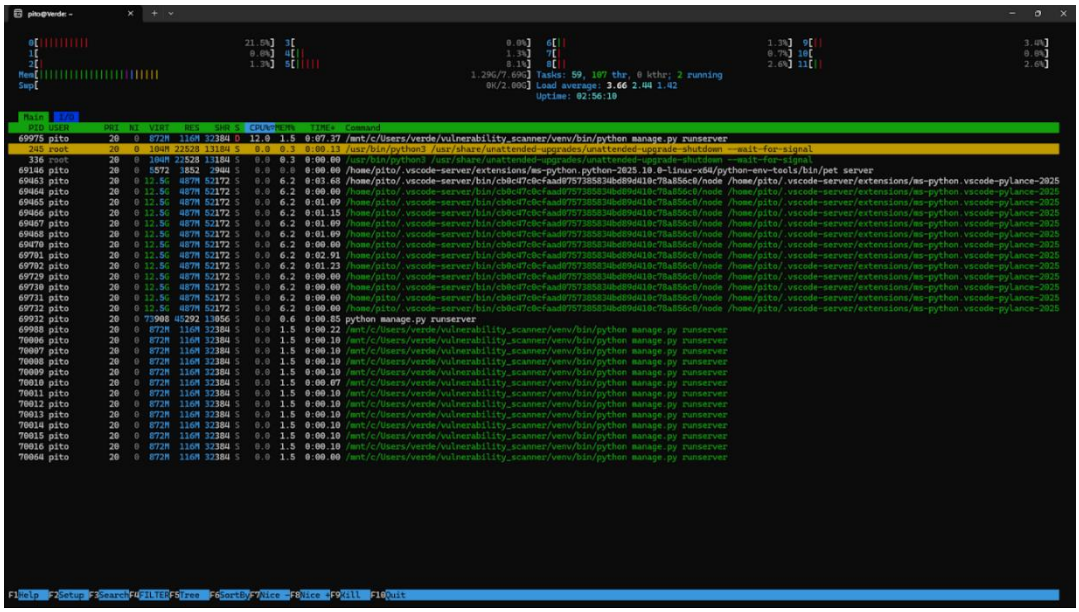


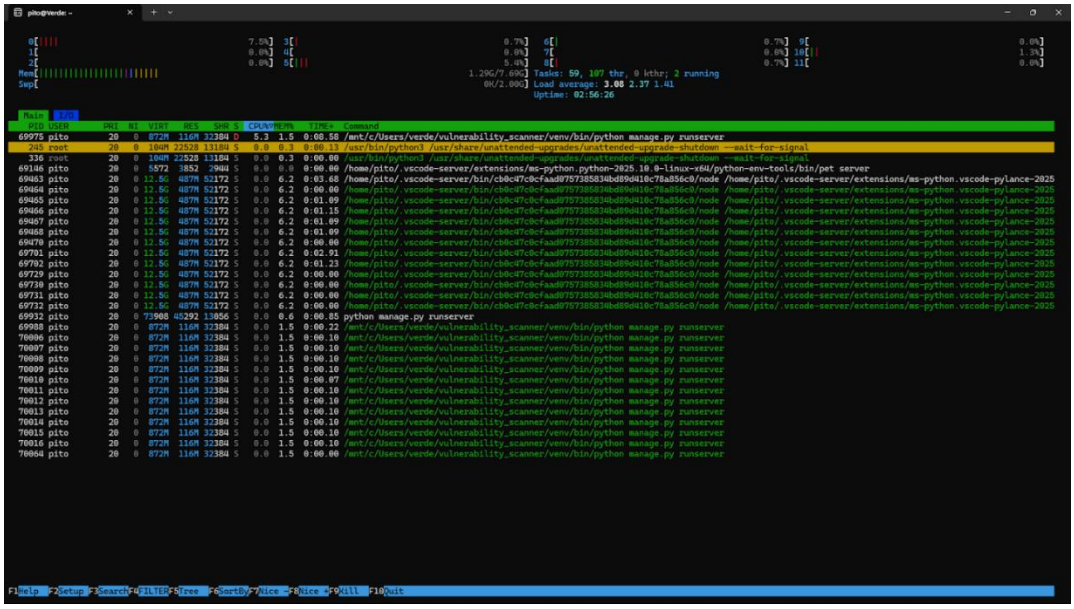
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
login	10000	9184	1	23537	5635.75	8.88%	18.0/sec	370.48	2.76	21061.5
Login Submit	10000	10418	1	40689	5351.32	11.69%	17.7/sec	359.50	6.14	20794.8
Save Scan API	10000	14156	1	40907	6230.92	14.65%	17.3/sec	362.58	7.97	21452.9
API Save Scan	10000	14398	1	40437	5582.55	12.81%	16.9/sec	361.48	8.06	21846.4
Nmap Scan	10000	14592	1	37339	5018.79	10.92%	16.7/sec	341.43	8.57	20886.5
Nikto Scan	10000	13015	1	33964	6280.17	19.27%	16.5/sec	309.71	7.65	19264.2
Get Scan History	10000	13259	0	38818	5996.66	15.55%	16.4/sec	320.60	5.08	20004.0
User Profile	10000	13046	1	27742	5996.63	15.06%	16.4/sec	322.44	5.11	20100.3
Logout	10000	12373	1	23657	5565.24	11.65%	16.4/sec	345.85	9.45	21532.1
TOTAL	90000	12716	0	40907	6008.66	13.42%	147.4/sec	2991.08	59.14	20783.5

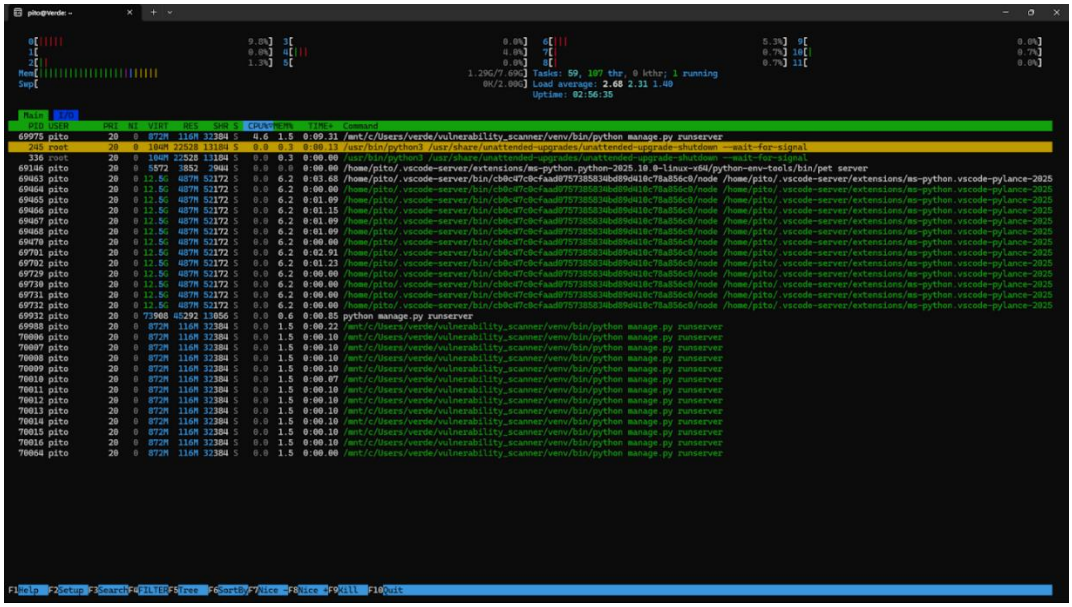


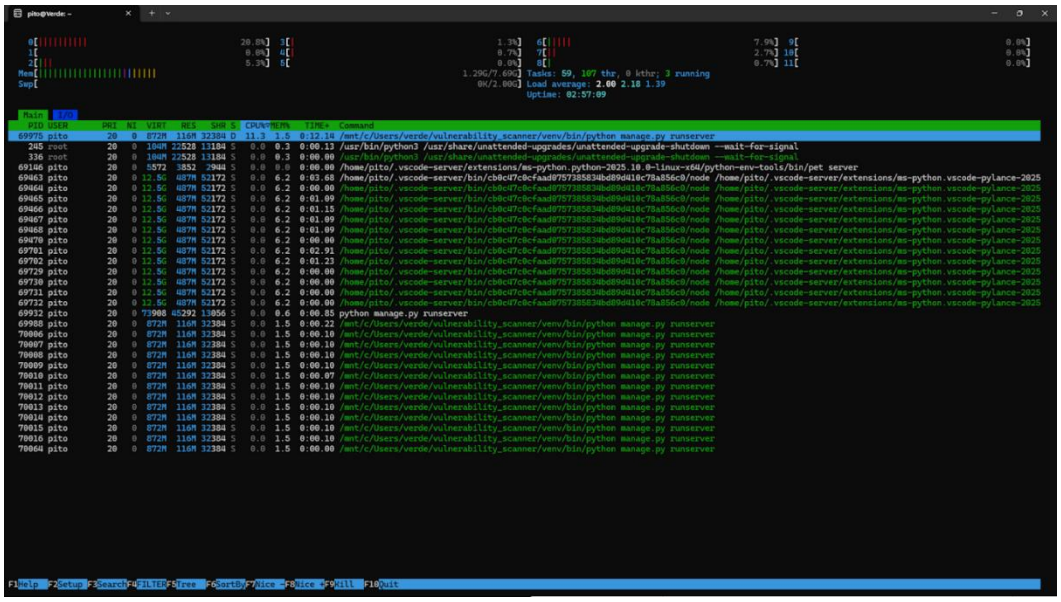
- Pengujian HTOP normal


pito@Verde: ~
Tasks: 59, 107 thr, 0 kthr; 2 running
Load average: 5.83 2.64 1.44
Uptime: 02:55:36
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 872M 116M 32384 R 4.0 1.5 0:04.61 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
245 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.13 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
336 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
69146 pito 20 0 8572 3852 2944 S 0.0 0.0 0:00.00 /home/pito/.vscode-server/extensions/ms-python.python-2025.10.0-linux-x64/python-env-tools/bin/pet server
69463 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:03.68 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69464 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69465 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69466 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.15 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69467 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69468 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69470 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69701 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:02.91 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69702 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.23 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69729 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69730 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69731 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69732 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
69988 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.22 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70064 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.00 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
F1Help F2Setup F3Search F4FILTER F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

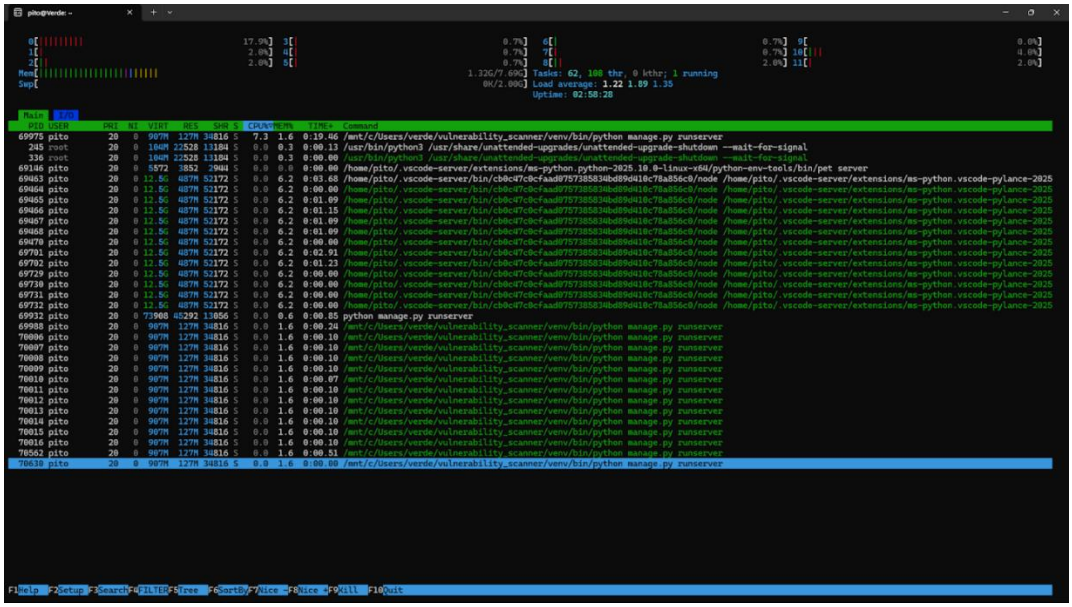

pito@Verde: ~
Tasks: 59, 107 thr, 0 kthr; 2 running
Load average: 3.66 2.44 1.42
Uptime: 02:56:10
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 872M 116M 32384 D 12.0 1.5 0:07.37 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
245 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.13 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
336 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
69146 pito 20 0 8572 3852 2944 S 0.0 0.0 0:00.00 /home/pito/.vscode-server/extensions/ms-python.python-2025.10.0-linux-x64/python-env-tools/bin/pet server
69463 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:03.68 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69464 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69465 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69466 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.15 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69467 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69468 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69470 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69701 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:02.91 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69702 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.23 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69729 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69730 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69731 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69732 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
69988 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.22 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70064 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.00 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
F1Help F2Setup F3Search F4FILTER F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

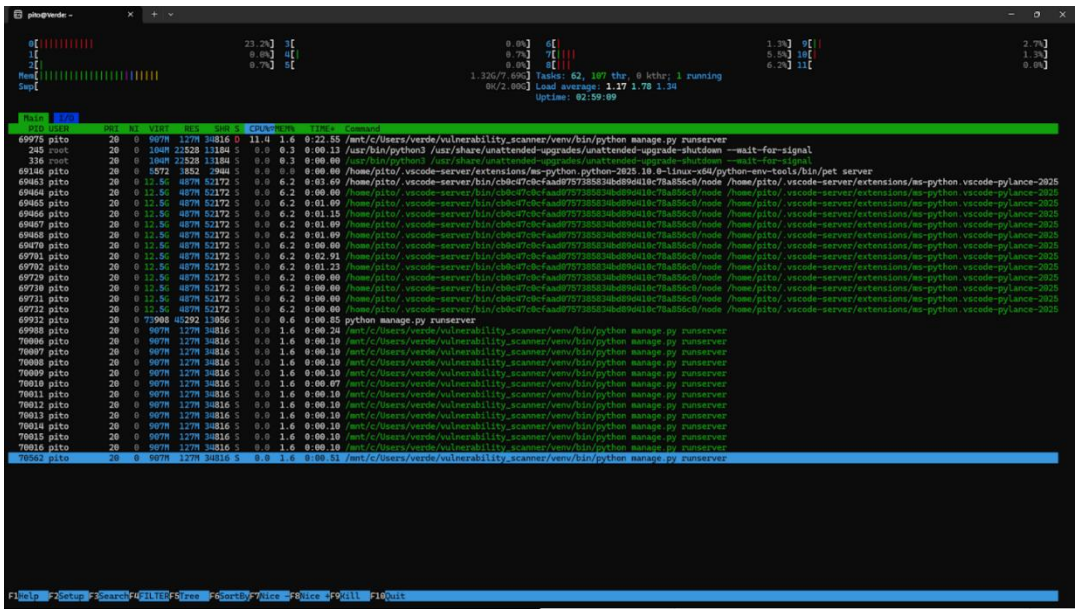

pito@Verde: ~
Tasks: 59, 107 thr, 0 kthr; 2 running
Load average: 3.08 2.37 1.41
Uptime: 02:56:26
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 872M 116M 32384 D 5.3 1.5 0:08.58 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
245 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.13 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
336 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
69146 pito 20 0 8572 3852 2944 S 0.0 0.0 0:00.00 /home/pito/.vscode-server/extensions/ms-python.python-2025.10.0-linux-x64/python-env-tools/bin/pet server
69463 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:03.68 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69464 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69465 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69466 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.15 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69467 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69468 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69470 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69701 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:02.91 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69702 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.23 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69729 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69730 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69731 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69732 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
69988 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.22 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70064 pito 20 0 872M 116M 32384 S 0.0 1.5 0:00.00 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
F1Help F2Setup F3Search F4FILTER F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

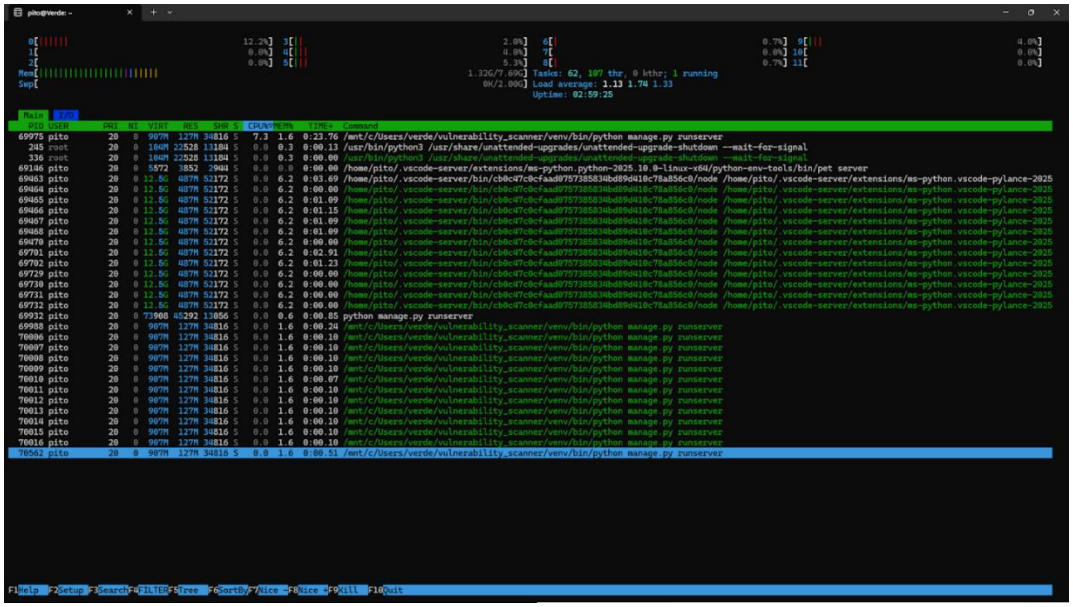


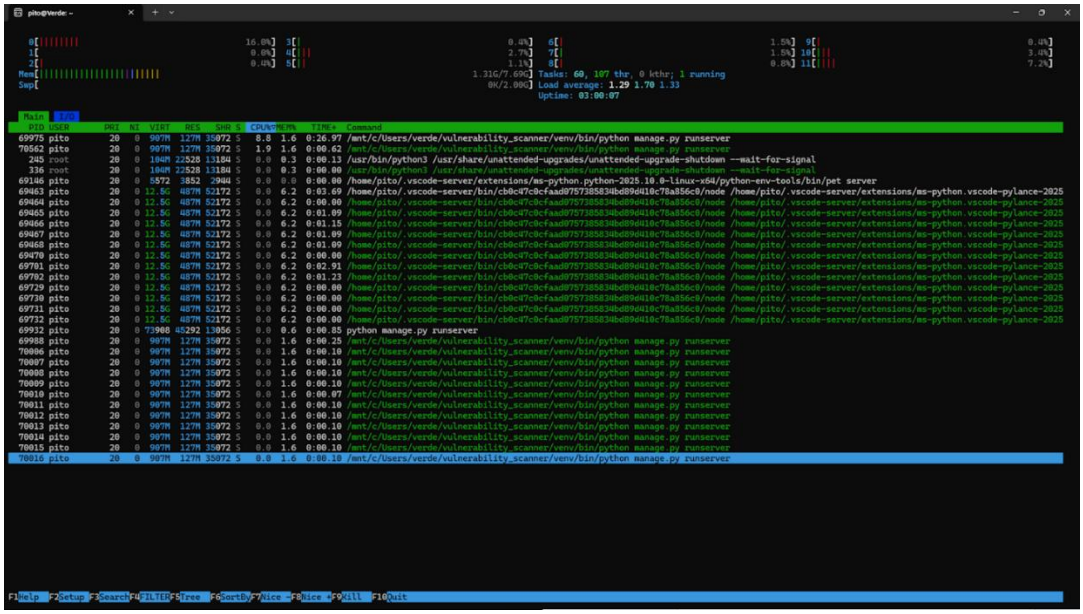


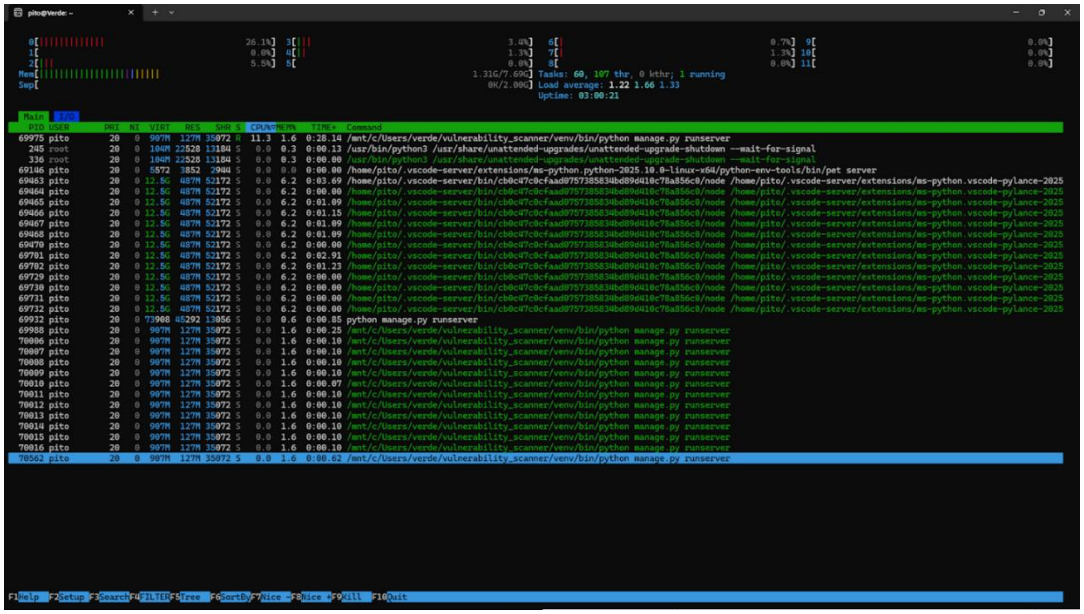
- Pengujian HTOP Nikto


pito@Verde: ~
0[||||||||| 17.9%] 3[0.7%] 6[0.7%] 9[0.0%]
1[2.8%] 4[0.7%] 7[0.7%] 10[||| 4.0%]
2[|| 2.8%] 5[0.7%] 8[|| 2.0%] 11[2.8%]
Mem[|||||||||||||||||| 1.32G/7.69G] Tasks: 62, 108 thr, 0 kthr; 1 running
Swp[0K/2.00G] Load average: 1.22 1.89 1.35
Uptime: 02:58:28
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 907M 127M 34816 S 7.3 1.6 0:19.46 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
245 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.13 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
336 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
69146 pito 20 0 8572 3852 2944 S 0.0 0.0 0:00.00 /home/pito/.vscode-server/extensions/ms-python.python-2025.10.0-linux-x64/python-env-tools/bin/pet server
69463 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:03.68 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69464 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69465 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69466 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.15 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69467 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69468 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69470 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69701 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:02.91 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69702 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.23 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69729 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69730 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69731 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69732 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
69988 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.24 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70562 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.51 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70630 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.00 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
F1Help F2Setup F3Search F4FILTER F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

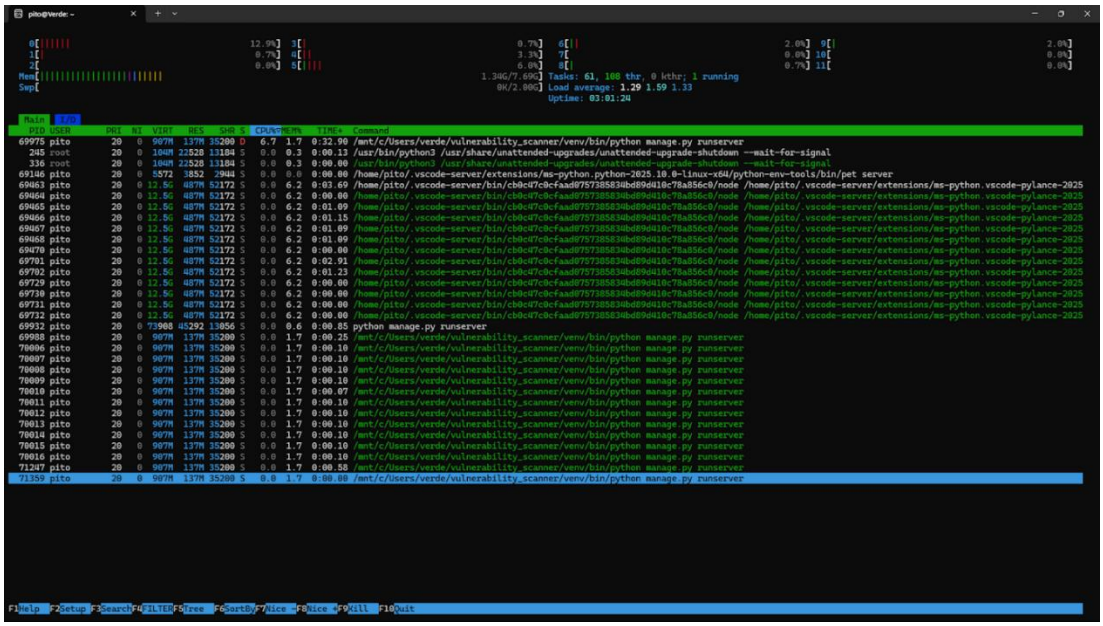

pito@Verde: ~
0[||||||||| 23.2%] 3[0.0%] 6[1.3%] 9[2.7%]
1[0.0%] 4[0.7%] 7[5.5%] 10[1.3%]
2[0.7%] 5[0.0%] 8[6.2%] 11[0.0%]
Mem[|||||||||||||||||| 1.32G/7.69G] Tasks: 62, 107 thr, 0 kthr; 1 running
Swp[0K/2.00G] Load average: 1.17 1.78 1.34
Uptime: 02:59:09
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 907M 127M 34816 D 11.4 1.6 0:22.55 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
245 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.13 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
336 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
69146 pito 20 0 8572 3852 2944 S 0.0 0.0 0:00.00 /home/pito/.vscode-server/extensions/ms-python.python-2025.10.0-linux-x64/python-env-tools/bin/pet server
69463 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:03.69 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69464 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69465 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69466 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.15 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69467 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69468 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69470 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69701 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:02.91 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69702 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.23 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69729 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69730 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69731 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69732 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c4f7c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
69988 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.24 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70562 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.51 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
F1Help F2Setup F3Search F4FILTER F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

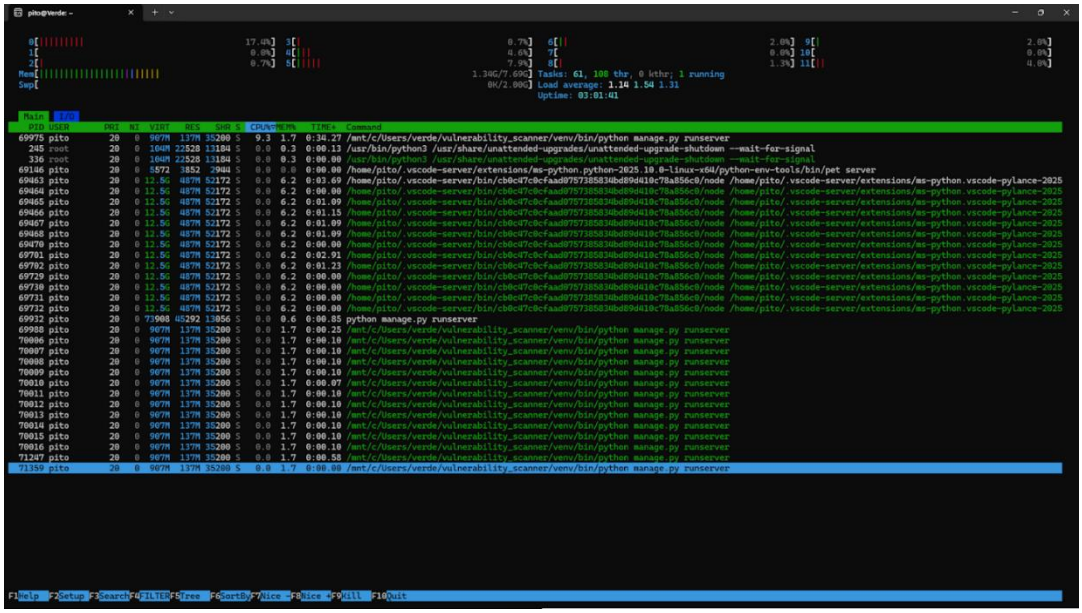

pito@Verde: ~
Tasks: 62, 107 thr, 0 kthr; 1 running
Load average: 1.13 1.74 1.33
Uptime: 02:59:25
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 907M 127M 34816 S 7.3 1.6 0:23.76 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
245 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.13 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
336 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
69146 pito 20 0 8572 3852 2944 S 0.0 0.0 0:00.00 /home/pito/.vscode-server/extensions/ms-python.python-2025.10.0-linux-x64/python-env-tools/bin/pet server
69463 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:03.69 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69464 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69465 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69466 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.15 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69467 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69468 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69470 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69701 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:02.91 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69702 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.23 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69729 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69730 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69731 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69732 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
69988 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.24 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 907M 127M 34816 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70562 pito 20 0 907M 127M 34816 S 0.8 1.6 0:00.51 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
F1Help F2Setup F3Search F4FILTER F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

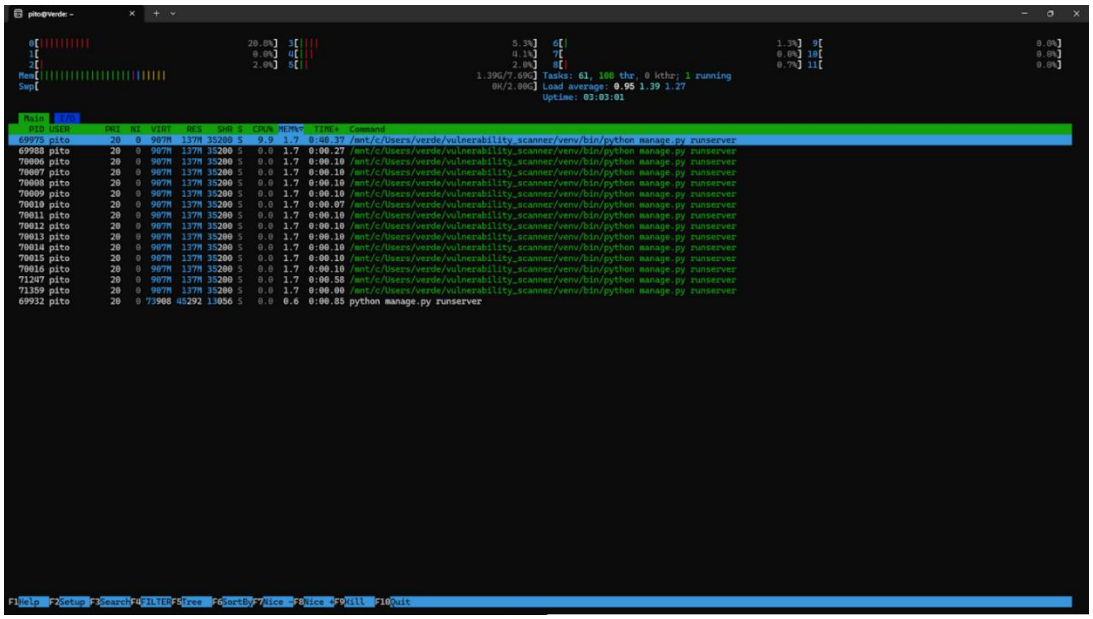

pito@Verde: ~
Tasks: 60, 107 thr, 0 kthr; 1 running
Load average: 1.29 1.70 1.33
Uptime: 03:00:07
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 907M 127M 35072 S 8.8 1.6 0:26.97 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70562 pito 20 0 907M 127M 35072 S 1.9 1.6 0:00.62 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
245 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.13 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
336 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
69146 pito 20 0 8572 3852 2944 S 0.0 0.0 0:00.00 /home/pito/.vscode-server/extensions/ms-python.python-2025.10.0-linux-x64/python-env-tools/bin/pet server
69463 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:03.69 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69464 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69465 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69466 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.15 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69467 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69468 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69470 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69701 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:02.91 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69702 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.23 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69729 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69730 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69731 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69732 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad0757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
69988 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.25 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 907M 127M 35072 S 0.0 1.6 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
F1Help F2Setup F3Search F4FILTER F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

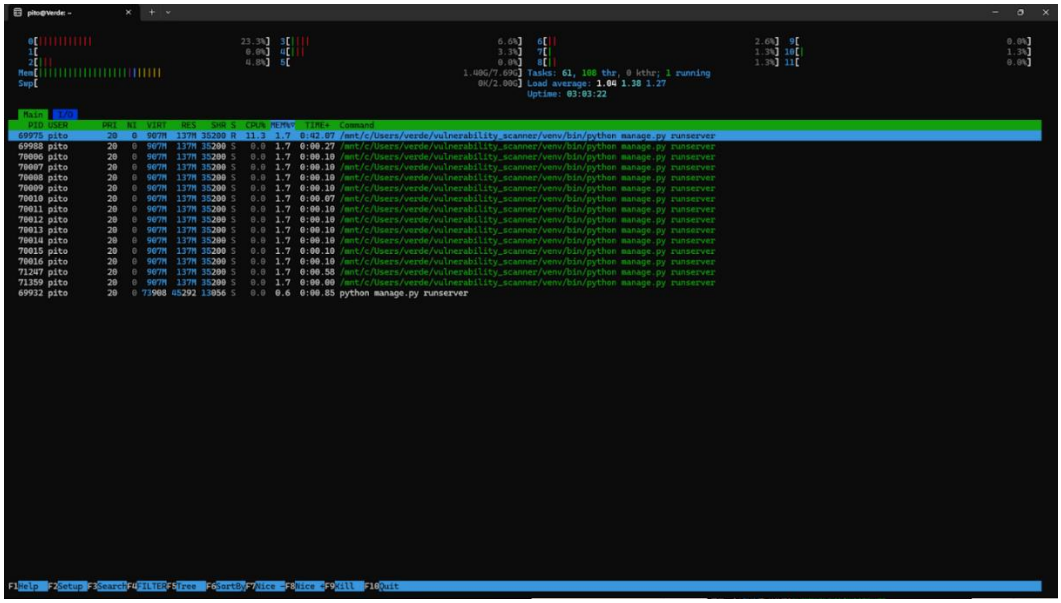


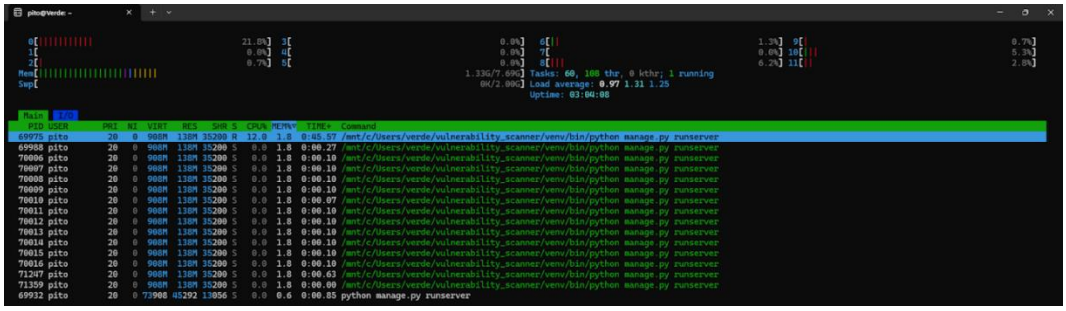
- Pengujian HTOP Nmap




pito@Verde: ~
Tasks: 61, 108 thr, 0 kthr; 1 running
Load average: 1.14 1.54 1.31
Uptime: 03:01:41
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 907M 137M 35200 S 9.3 1.7 0:34.27 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
245 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.13 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
336 root 20 0 104M 22528 13184 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
69146 pito 20 0 8572 3852 2944 S 0.0 0.0 0:00.00 /home/pito/.vscode-server/extensions/ms-python.python-2025.10.0-linux-x64/python-env-tools/bin/pet server
69463 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:03.69 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69464 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69465 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69466 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.15 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69467 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69468 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.09 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69470 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69701 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:02.91 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69702 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:01.23 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69729 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69730 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69731 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69732 pito 20 0 12.5G 487M 52172 S 0.0 6.2 0:00.00 /home/pito/.vscode-server/bin/cb0c47c0cfaad9757385834bd89d410c78a856c0/node /home/pito/.vscode-server/extensions/ms-python.vscode-pylance-2025
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
69988 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.25 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
71247 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.58 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
71359 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.08 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
F1Help F2Setup F3SearchF4FILTERF5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

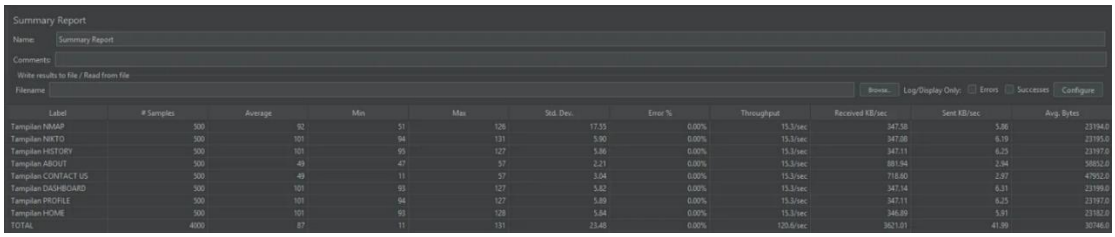

pito@Verde: ~
Tasks: 61, 108 thr, 0 kthr; 1 running
Load average: 0.95 1.39 1.27
Uptime: 03:03:01
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 907M 137M 35200 S 9.9 1.7 0:40.37 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
69988 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.27 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
71247 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.58 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
71359 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.00 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
F1Help F2Setup F3SearchF4FILTERF5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

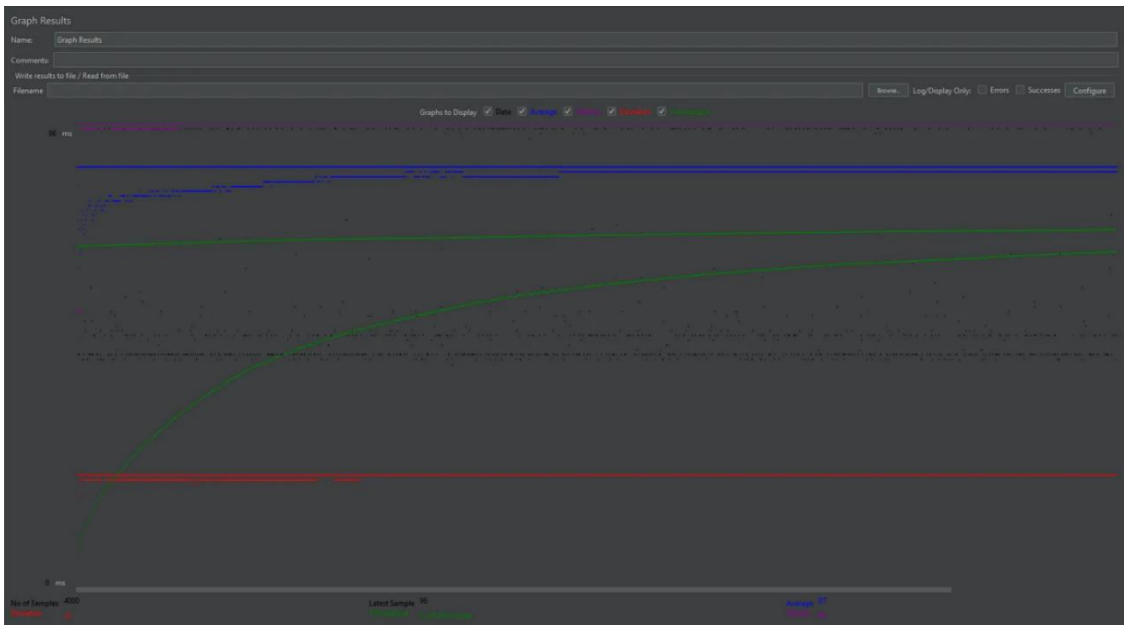

pito@Verde: ~
Tasks: 61, 108 thr, 0 kthr; 1 running
Load average: 1.04 1.38 1.27
Uptime: 03:03:22
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
69975 pito 20 0 907M 137M 35200 R 11.3 1.7 0:42.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
69988 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.27 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70006 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70007 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70008 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70009 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70010 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.07 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70011 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70012 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70013 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70014 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70015 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
70016 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.10 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
71247 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.58 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
71359 pito 20 0 907M 137M 35200 S 0.0 1.7 0:00.00 /mnt/c/Users/verde/vulnerability_scanner/venv/bin/python manage.py runserver
69932 pito 20 0 73908 45292 13056 S 0.0 0.6 0:00.85 python manage.py runserver
F1Help F2Setup F3SearchF4FILTERF5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit



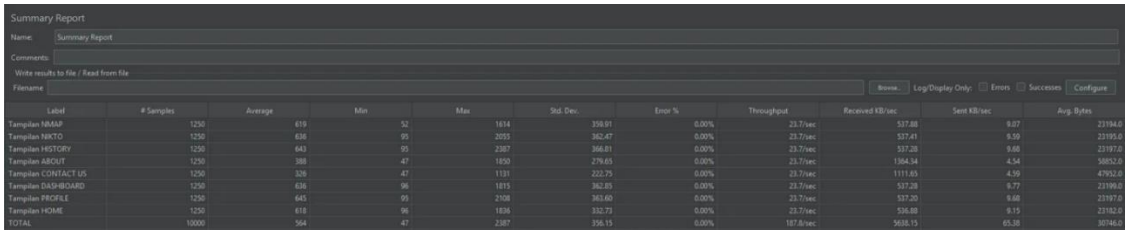
i. Pengujian Frontend

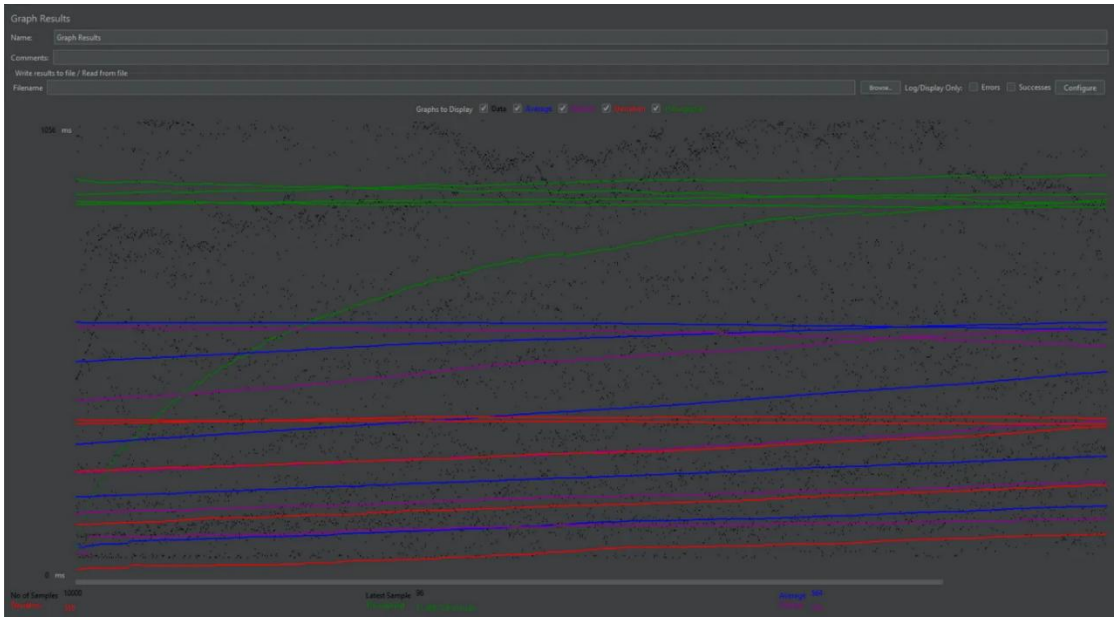
- Pengujian 1



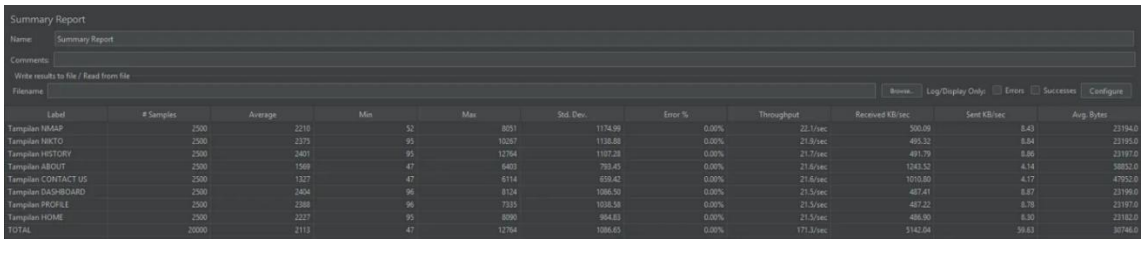


- Pengujian 2

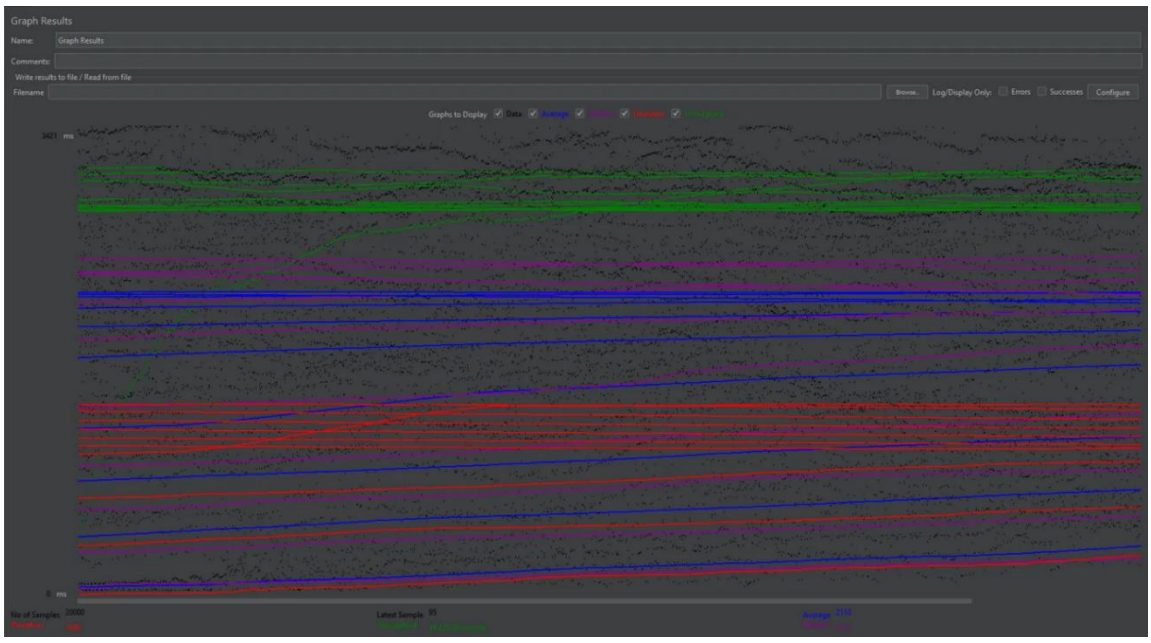




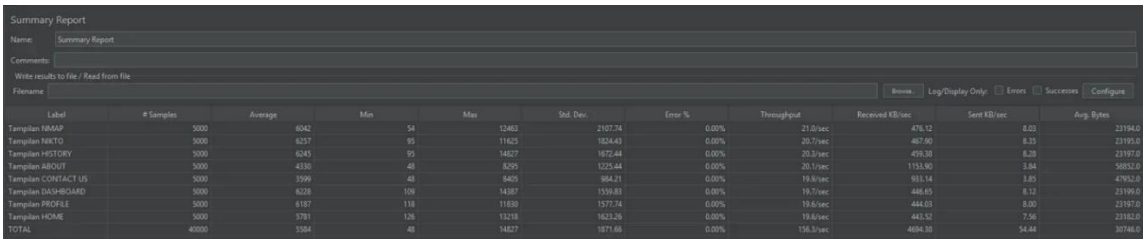
- Pengujian 3

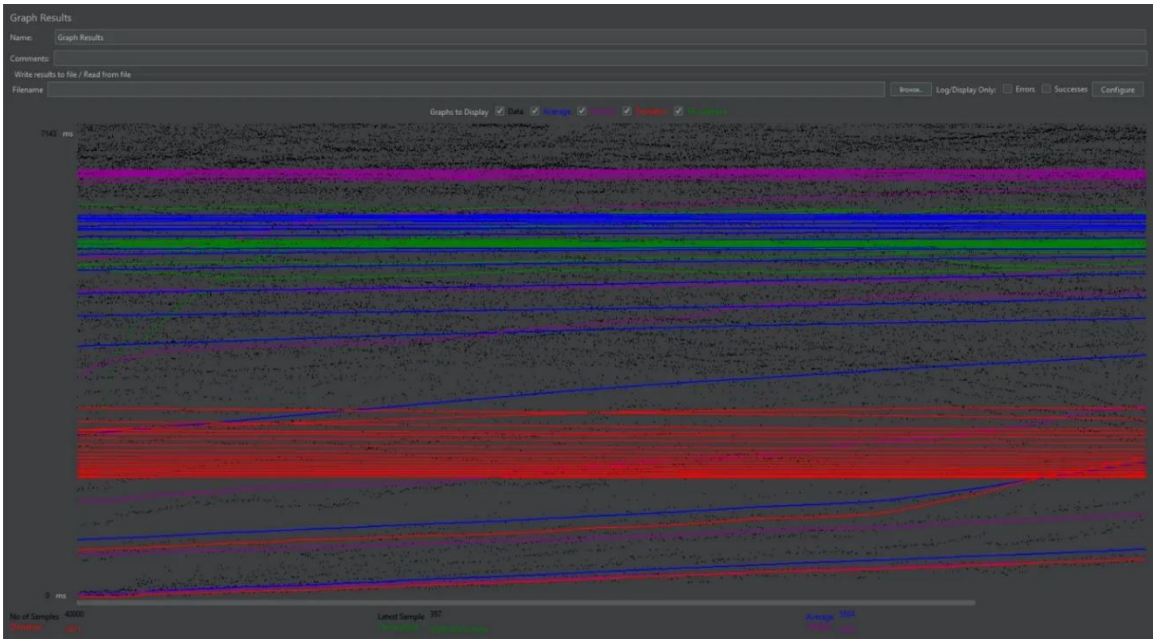


Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Tampilan NMAP	2500	2210	52	8051	1174.99	0.00%	22.1/sec	500.09	8.43	23194.0
Tampilan NIKTO	2500	2373	95	10267	1138.88	0.00%	21.9/sec	495.32	8.84	23195.0
Tampilan HISTORY	2500	2401	95	12764	1107.28	0.00%	21.7/sec	491.79	8.86	23197.0
Tampilan ABOUT	2500	1569	47	6403	793.45	0.00%	21.6/sec	1243.52	4.14	58852.0
Tampilan CONTACT US	2500	1327	47	6114	659.42	0.00%	21.6/sec	1010.80	4.17	47952.0
Tampilan DASHBOARD	2500	2404	96	8124	1086.50	0.00%	21.5/sec	487.41	8.87	23199.0
Tampilan PROFILE	2500	2388	96	7335	1038.58	0.00%	21.5/sec	487.22	8.78	23197.0
Tampilan HOME	2500	2227	95	8090	984.83	0.00%	21.5/sec	486.80	8.30	23182.0
TOTAL	20000	2113	47	12764	1086.65	0.00%	171.3/sec	5142.04	59.63	30746.0

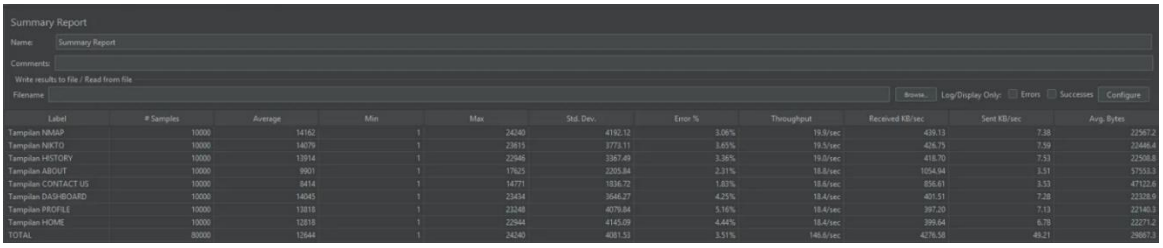


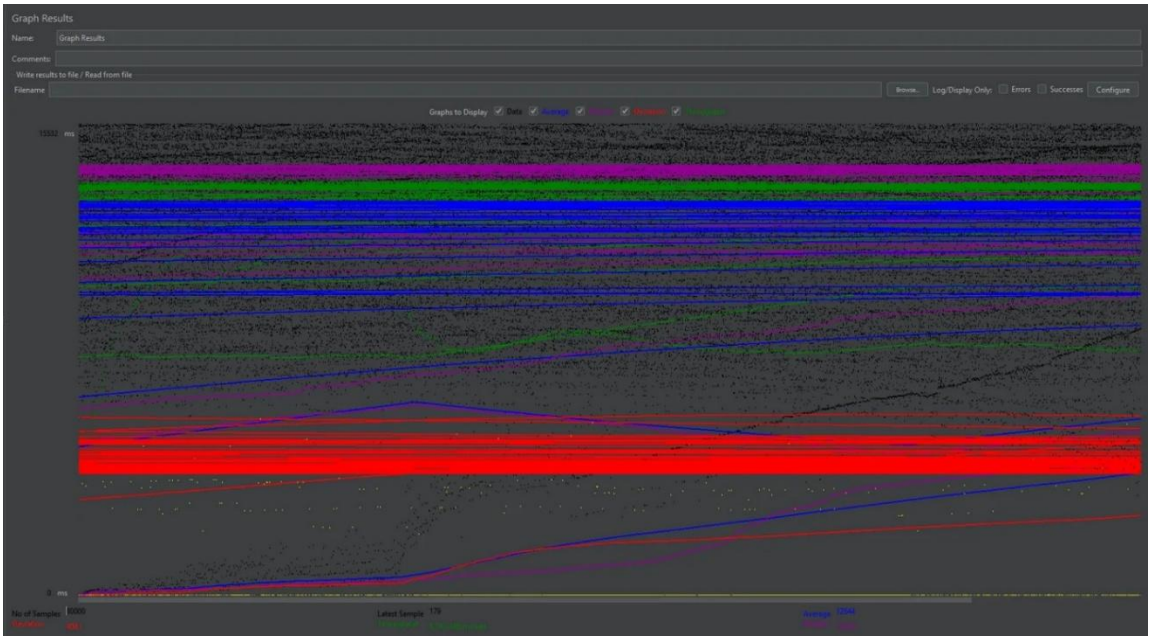
- Pengujian 4





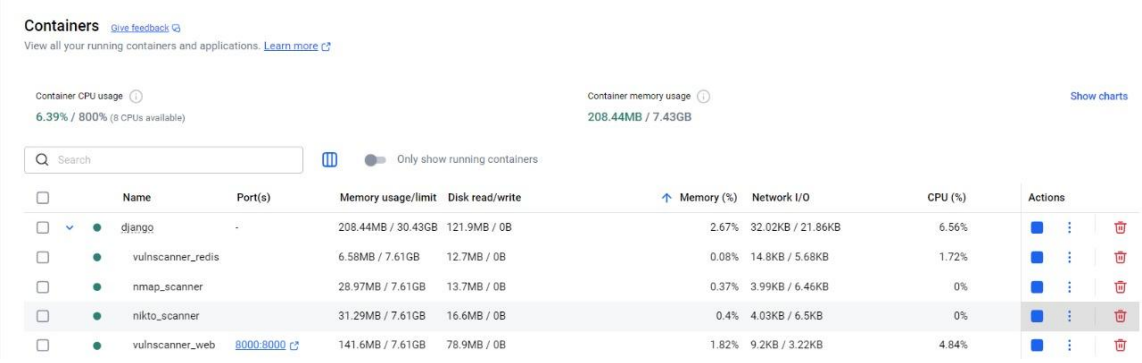
- Pengujian 5

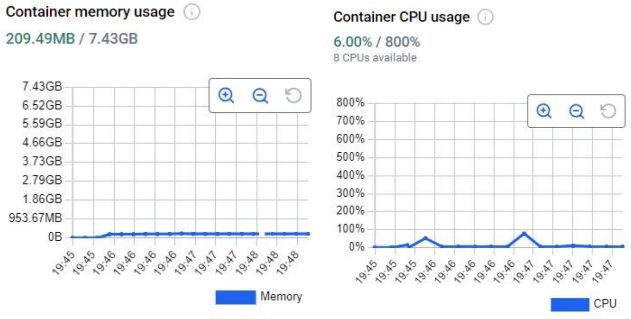




j. Pengujian Docker

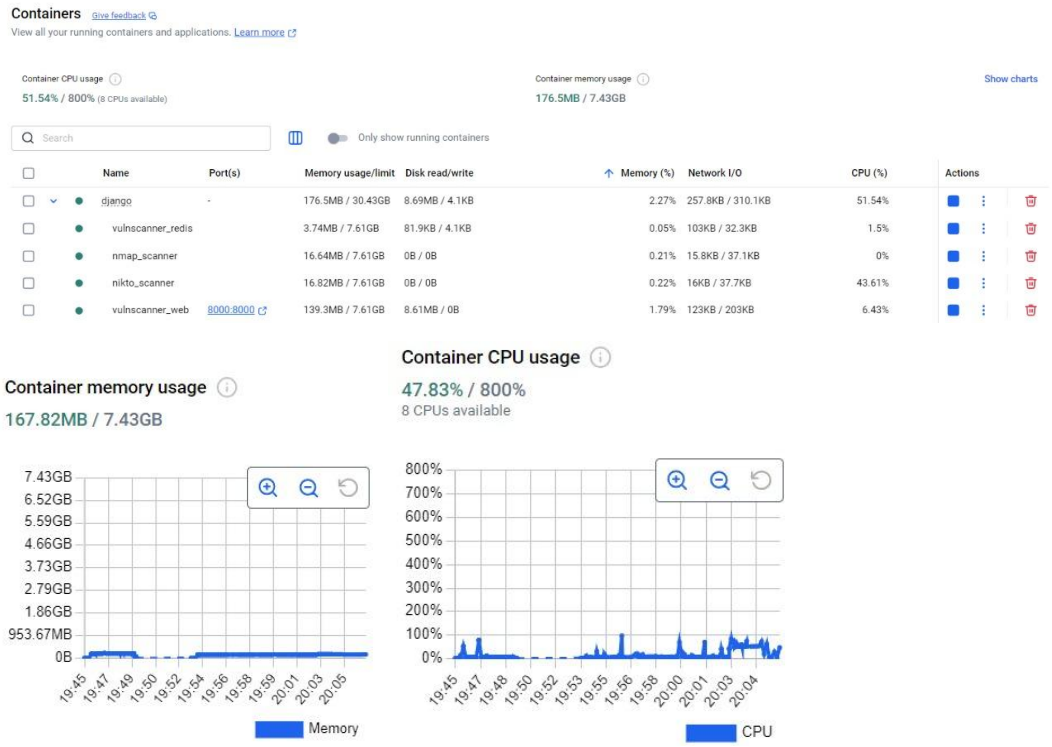
1. Docker Aktif tanpa Pemindaian



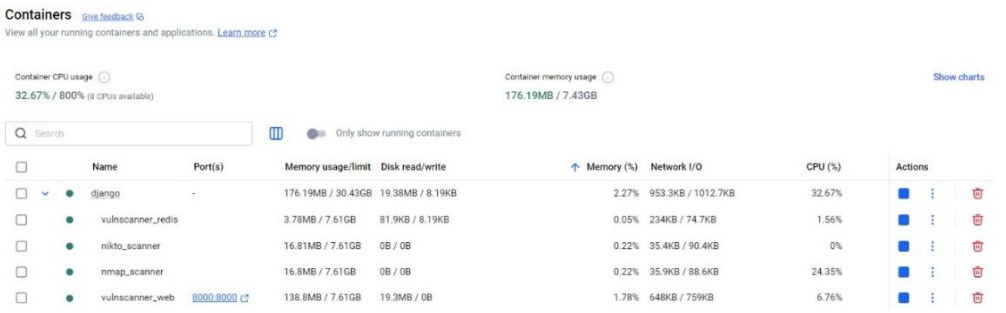


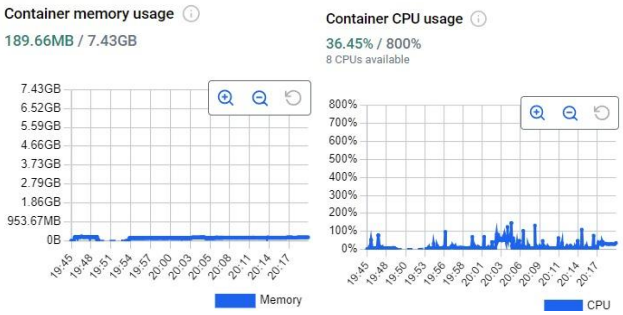
2. Docker Aktif dengan Pemindaian

- Nikto



- Nmap





- Nikto Nmap Bersamaan