Jurnal Minfo Polgan (JMP)
Volume 15, Nomor 3, September-Desember 2026
DOI : https://doi.org/10.33395/jmp.v15i3.16457
e-ISSN : 2797-3298
p-ISSN : 2089-9424

Implementasi Vulnerability Scanner Berbasis Python untuk Analisis Kerentanan dan Rekomendasi Mitigasi Keamanan Website

[1]Muhamad Saroful Hakim, [2]Azhari Shouni Barkah, [3]Darso
[1*,2,3]Teknologi Informasi & Informatika , Fakultas Ilmu Komputer Universitas Amikom Purwokerto, Banyumas, Indonesia
*Korespondensi: sarofulkimha@gmail.com

Submit : 09 Jun 2026 | **Diterima** : 01 Jul 2026 | **Terbit** : 01 Sept 2026

ABSTRACT

The rapid growth of websites has increased the need for security systems capable of protecting data and services from cyber threats. Vulnerabilities such as weak security configurations, incomplete security headers, and inadequate cookie protection can increase the risk of website attacks. This study aims to implement a Python-based vulnerability scanner to analyze vulnerabilities and provide automated security mitigation recommendations. The research employed the Prototype method, including requirements gathering, prototype development, evaluation, implementation, testing, and final system development. The proposed system is capable of examining HTTPS implementation, SSL/TLS certificate validity, security headers, cookie security, directory listing exposure, robots.txt exposure, and supported HTTP methods. Black Box Testing results showed that all system functions operated as expected. The system successfully performs website security scanning, generates a Security Score, classifies findings based on risk levels, and provides improvement recommendations along with reports in HTML, CSV, and JSON formats. The developed system supports vulnerability assessment activities more quickly, efficiently, and consistently than manual inspection methods.

Keywords: *Python; security mitigation; vulnerability analysis; vulnerability assessment; website security.*

ABSTRAK

Perkembangan website yang semakin pesat meningkatkan kebutuhan akan sistem keamanan yang mampu melindungi data dan layanan dari berbagai ancaman siber. Kerentanan seperti konfigurasi keamanan yang lemah, security header yang tidak lengkap, dan keamanan cookie yang kurang optimal dapat meningkatkan risiko serangan pada website. Penelitian ini bertujuan mengimplementasikan vulnerability scanner berbasis Python untuk menganalisis kerentanan dan memberikan rekomendasi mitigasi keamanan secara otomatis. Metode yang digunakan adalah Prototype yang meliputi pengumpulan kebutuhan, pembuatan prototype, evaluasi, implementasi, pengujian, dan sistem final. Sistem yang dikembangkan mampu memeriksa HTTPS, validitas sertifikat SSL/TLS, security header, keamanan cookie, directory listing, robots.txt exposure, dan metode HTTP. Pengujian menggunakan Black Box Testing menunjukkan seluruh fitur berjalan sesuai kebutuhan. Hasil penelitian menunjukkan sistem mampu melakukan pemindaian keamanan website, menghasilkan Security Score, mengelompokkan temuan berdasarkan tingkat risiko, serta menyediakan rekomendasi perbaikan dan laporan dalam format HTML, CSV, dan JSON. Sistem ini membantu proses vulnerability assessment menjadi lebih cepat, efisien, dan konsisten.

Kata Kunci: analisis kerentanan; keamanan website; mitigasi keamanan; Python; vulnerability assessment.

PENDAHULUAN

Perkembangan teknologi informasi telah mendorong penggunaan website sebagai media utama dalam berbagai aktivitas, mulai dari pendidikan, bisnis, pemerintahan, hingga layanan publik (He et al. 2026). Website tidak hanya berfungsi sebagai sarana penyampaian informasi, tetapi juga menjadi platform transaksi dan pengelolaan data yang penting bagi organisasi (Siddiq et al. 2025). Kemudahan akses melalui internet membuat jumlah website terus meningkat setiap tahun. Kondisi ini menunjukkan bahwa website telah menjadi bagian yang tidak terpisahkan dari kehidupan masyarakat modern dan mendukung transformasi digital

(Shidqi et al. 2025).

Seiring meningkatnya penggunaan website, aspek keamanan menjadi perhatian utama dalam pengelolaan sistem informasi (Jehian et al. 2025). Keamanan website bertujuan untuk menjaga kerahasiaan data agar tidak diakses oleh pihak yang tidak berwenang, memastikan integritas data tetap terjaga dari manipulasi, serta menjamin ketersediaan layanan bagi pengguna (Jayanto 2025). Ketiga aspek tersebut dikenal sebagai Confidentiality, Integrity, dan Availability (CIA) yang menjadi dasar dalam keamanan informasi. Kegagalan menjaga salah satu aspek tersebut dapat mengakibatkan kerugian bagi organisasi maupun pengguna (Ainurrohman dan Nurasri 2025).

Website yang terhubung dengan internet rentan terhadap berbagai ancaman keamanan yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab. Beberapa kerentanan yang sering ditemukan antara lain SQL Injection yang memungkinkan penyerang mengakses basis data secara ilegal, Cross-Site Scripting (XSS) yang dapat menyisipkan skrip berbahaya ke halaman web, Directory Traversal yang memungkinkan akses ke file sensitif, serta Broken Authentication yang dapat menyebabkan penyalahgunaan akun pengguna (Ariyadi dan Fadli 2025). Ancaman tersebut menunjukkan pentingnya pengujian keamanan secara berkala untuk mengurangi risiko serangan siber. Kerentanan pada website dapat memberikan dampak yang signifikan bagi organisasi maupun pengguna (Corradini 2025). Bagi organisasi, kebocoran data dapat menyebabkan kerugian finansial, penurunan reputasi, dan hilangnya kepercayaan pelanggan (Anugrah et al. 2025). Sementara itu, pengguna dapat mengalami pencurian data pribadi, penyalahgunaan akun, hingga kerugian materi akibat aktivitas kriminal siber (Djabar 2025). Selain itu, gangguan pada layanan website dapat menghambat aktivitas operasional organisasi. Oleh karena itu, identifikasi dan penanganan kerentanan perlu dilakukan secara proaktif untuk meminimalkan dampak (Issaf dan Koprawi 2025).

Vulnerability assessment merupakan proses identifikasi, analisis, dan evaluasi terhadap kerentanan yang terdapat pada suatu sistem. Proses ini berperan penting sebagai langkah awal dalam meningkatkan keamanan website karena mampu memberikan informasi mengenai celah keamanan yang berpotensi dimanfaatkan oleh penyerang (Okonkwo et al. 2025). Hasil vulnerability assessment dapat digunakan sebagai dasar dalam menentukan prioritas perbaikan dan strategi mitigasi risiko. Dengan melakukan penilaian kerentanan secara berkala, organisasi dapat meningkatkan tingkat keamanan sistem serta mengurangi kemungkinan terjadinya insiden keamanan informasi (Sales dan Ticle 2025). Pemeriksaan keamanan website secara manual sering kali memerlukan waktu, tenaga, dan keahlian khusus yang tidak sedikit. Proses ini mengharuskan administrator atau penguji keamanan melakukan pengecekan satu per satu terhadap berbagai potensi kerentanan yang terdapat pada website. Selain memakan waktu, pemeriksaan manual juga berisiko mengalami kesalahan akibat faktor manusia, terutama pada sistem yang memiliki kompleksitas tinggi. Oleh karena itu, diperlukan solusi yang mampu membantu proses identifikasi kerentanan secara lebih cepat, efisien, dan konsisten melalui pendekatan otomatisasi.

Python merupakan salah satu bahasa pemrograman yang banyak digunakan dalam bidang keamanan siber karena memiliki sintaks yang sederhana dan didukung oleh berbagai pustaka yang mendukung proses pengujian keamanan (Munandar, Ocsa, dan Saian 2026). Berbagai framework dan library seperti Requests, BeautifulSoup, dan Scapy menjadikan Python sebagai pilihan yang tepat dalam pengembangan alat deteksi kerentanan website berbasis otomatis. Pemilihan Python dalam penelitian ini didasarkan pada beberapa keunggulan yang dimilikinya (Ariyadi dan Fadli 2025). Python memiliki struktur kode yang mudah dipahami sehingga mempercepat proses pengembangan dan pemeliharaan sistem (Rizkayanti dan Yunanri 2023). Keunggulan tersebut menjadikan Python sebagai platform yang sesuai untuk membangun sistem deteksi kerentanan website.

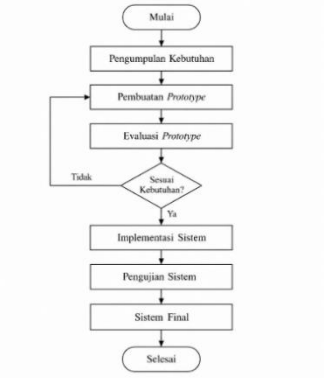
Beberapa penelitian terdahulu menunjukkan bahwa vulnerability assessment efektif dalam mengidentifikasi kerentanan keamanan website. (Anwari, Wedana, dan Deva 2022) berhasil menemukan berbagai celah keamanan menggunakan XSpear, XSSCon, dan PwnXSS. (Rizkayanti dan Yunanri 2023) menggunakan OWASP ZAP untuk mengidentifikasi kerentanan pada website sistem informasi kependudukan, sedangkan (Chowdhury, Rahman, dan Rahman 2024) menunjukkan bahwa pendekatan berbasis Python mampu membantu proses deteksi kerentanan secara otomatis dan lebih efisien. Namun, sebagian besar penelitian masih memanfaatkan tools yang telah tersedia, sehingga masih diperlukan pengembangan sistem vulnerability assessment berbasis Python yang lebih fleksibel dan dapat disesuaikan dengan

kebutuhan pengguna. Oleh karena itu, penelitian ini mengusulkan implementasi sistem deteksi kerentanan website berbasis Python untuk mendukung proses penilaian keamanan website secara otomatis.

Berdasarkan berbagai permasalahan keamanan website yang masih sering ditemukan, diperlukan suatu sistem yang mampu membantu proses identifikasi kerentanan secara cepat dan akurat. Penelitian ini berfokus pada implementasi sistem deteksi kerentanan website berbasis Python untuk mendukung proses penilaian keamanan. Sistem yang dibangun diharapkan dapat membantu administrator maupun pengembang website dalam mengenali potensi risiko keamanan sejak dini. Selain itu, penelitian ini diharapkan memberikan kontribusi dalam pengembangan solusi keamanan berbasis otomatis yang mudah diterapkan pada berbagai jenis website.

METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah metode Prototype. Menurut Pressman, metode Prototype merupakan metode pengembangan sistem yang dilakukan melalui pembuatan model awal (prototype) yang kemudian dievaluasi dan disempurnakan secara berulang berdasarkan kebutuhan pengguna hingga menghasilkan sistem yang sesuai dengan tujuan penelitian . Metode ini dipilih karena memungkinkan proses pengembangan sistem deteksi kerentanan website dilakukan secara bertahap dan lebih terarah. Dalam penelitian ini dilakukan modifikasi dengan menambahkan proses pengujian keamanan website pada tahap implementasi sistem. Adapun tahapan yang diterapkan meliputi pengumpulan kebutuhan, pembuatan prototype, evaluasi prototype, implementasi sistem, pengujian sistem, dan sistem final.



Gambar 1 Alur Metode Prototype

Berdasarkan Gambar 1, tahapan metode Prototype yang digunakan dalam penelitian ini dijelaskan sebagai berikut.

Pengumpulan Kebutuhan

Tahap pengumpulan kebutuhan dilakukan untuk mengidentifikasi permasalahan dan kebutuhan sistem yang akan dikembangkan. Proses ini dilakukan melalui studi literatur mengenai keamanan website, vulnerability assessment, dan Python. Selain itu, dianalisis kebutuhan fungsional serta nonfungsional sistem. Hasil tahap ini menjadi dasar dalam menentukan fitur dan proses pada sistem deteksi kerentanan website (Fauzi et al. 2026).

Pembuatan Prototype

Tahap pembuatan prototype dilakukan dengan merancang model awal sistem yang mencakup antarmuka, proses pemindaian, dan laporan hasil analisis. Prototype digunakan sebagai acuan evaluasi serta penyempurnaan sebelum sistem diimplementasikan secara penuh (Nainggolan 2026).

Evaluasi Prototype

ahap evaluasi prototype dilakukan untuk memastikan bahwa rancangan sistem telah sesuai dengan kebutuhan penelitian. Evaluasi mencakup desain antarmuka, alur proses, dan fungsi yang tersedia pada prototype. Hasil evaluasi digunakan untuk menemukan kekurangan atau kesalahan sehingga dapat dilakukan perbaikan dan penyempurnaan sebelum sistem diimplementasikan (Putrie et al. 2026).

Implementasi Sistem

Tahap implementasi sistem dilakukan dengan menerjemahkan rancangan prototype ke dalam aplikasi menggunakan bahasa pemrograman Python. Sistem dikembangkan untuk melakukan analisis keamanan website secara otomatis, meliputi pemeriksaan HTTPS, SSL/TLS, security header, keamanan cookie, directory listing, robots.txt, metode HTTP, serta penyajian laporan hasil analisis keamanan website (Giawa, Abdy, dan Alasi 2026).

Pengujian Sistem

Tahap pengujian dilakukan menggunakan metode Black Box Testing untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan. Pengujian mencakup fitur pemindaian keamanan website dan menunjukkan bahwa sistem mampu mendeteksi kerentanan secara otomatis dengan baik (Akbar 2026).

Sistem Final

Tahap sistem final merupakan tahap akhir setelah implementasi dan pengujian selesai dilakukan. Sistem telah disempurnakan sehingga mampu melakukan pemindaian keamanan website secara otomatis, mendeteksi potensi kerentanan, menampilkan hasil analisis, serta menghasilkan laporan yang mendukung evaluasi dan peningkatan keamanan website (Irawan et al. 2026).

HASIL DAN PEMBAHASAN

Hasil Pengumpulan Kebutuhan

Tahap pengumpulan kebutuhan dilakukan melalui studi literatur mengenai keamanan website, vulnerability assessment, serta teknologi yang digunakan dalam pengembangan sistem. Berdasarkan hasil analisis, sistem yang dikembangkan harus mampu melakukan pemeriksaan keamanan website secara otomatis dan menampilkan hasil analisis dalam bentuk laporan yang mudah dipahami pengguna, Kebutuhan fungsional yang diperoleh meliputi:

Table 1 Hasil Pengumpulan Kebutuhan

No	Kebutuhan Fungsional
1.	Sistem mampu menerima input URL website.
2.	Sistem mampu melakukan pemindaian keamanan website.
3.	Sistem mampu memeriksa penggunaan HTTPS.
4.	Sistem mampu memeriksa validitas sertifikat SSL/TLS.
5.	Sistem mampu memeriksa Security Header.
6.	Sistem mampu memeriksa keamanan Cookie.
7.	Sistem mampu mendeteksi Directory Listing.
8.	Sistem mampu memeriksa Robots.txt Exposure.
9.	Sistem mampu menampilkan skor keamanan website.
10.	Sistem mampu menghasilkan laporan hasil pemindaian.

Sedangkan kebutuhan nonfungsional meliputi penggunaan bahasa pemrograman *Python*, dan *google colaboratory* antarmuka yang mudah digunakan, dan waktu pemrosesan yang efisien.

Hasil Pembuatan Prototype

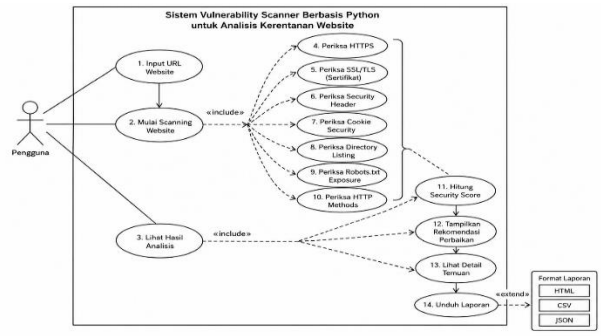
Berdasarkan kebutuhan yang telah diperoleh, dilakukan perancangan prototype sistem. Prototype dirancang untuk memberikan gambaran mengenai proses kerja sistem mulai dari input URL website hingga menampilkan hasil analisis keamanan. Tahap ini menghasilkan rancangan antarmuka yang terdiri dari:

Table 2 Hasil Kebutuhan Fitur

No	Hasil Antarmuka
1.	Halaman input URL website.
2.	Tombol proses scanning.
3.	Halaman hasil analisis.
4.	Tampilan skor keamanan.

No	Hasil Antarmuka
5.	Rekomendasi perbaikan keamanan.
6.	Hasil Laporan Pemindaian

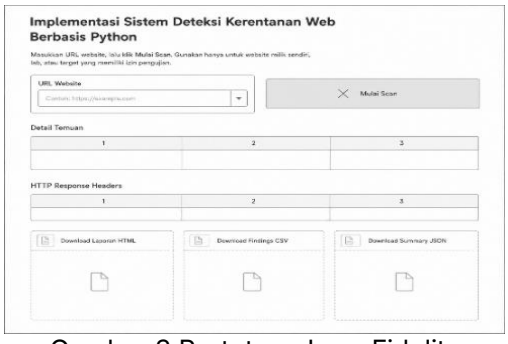
Berdasarkan kebutuhan fitur pada Tabel 2, selanjutnya dilakukan perancangan Use Case Diagram untuk menggambarkan interaksi antara pengguna dengan sistem serta fungsi-fungsi yang tersedia pada aplikasi deteksi kerentanan website. Use Case Diagram digunakan untuk memodelkan kebutuhan fungsional sistem secara visual sehingga alur penggunaan sistem dapat dipahami dengan lebih jelas. Diagram ini menjadi dasar dalam pengembangan prototype antarmuka karena menunjukkan layanan yang dapat diakses oleh pengguna dan proses yang dilakukan oleh sistem. Hasil perancangan Use Case Diagram dapat dilihat pada Gambar 2.



Gambar 2 Usecase Diagram

Berdasarkan Gambar 2, Use Case Diagram menggambarkan interaksi antara pengguna dan sistem vulnerability scanner yang dikembangkan. Pengguna dapat memasukkan URL website, menjalankan proses scanning, melihat hasil analisis, serta mengunduh laporan hasil pemeriksaan. Selama proses scanning, sistem melakukan berbagai pemeriksaan keamanan seperti HTTPS, SSL/TLS, Security Header, Cookie Security, Robots.txt, dan HTTP Methods. Hasil pemeriksaan kemudian digunakan untuk menghasilkan security score, rekomendasi perbaikan, dan detail temuan kerentanan yang dapat digunakan sebagai bahan evaluasi keamanan website.

Berdasarkan Use Case Diagram yang telah dirancang, selanjutnya dibuat prototype antarmuka sistem yang menggambarkan keseluruhan tampilan aplikasi deteksi kerentanan website. Prototype ini berfungsi sebagai acuan dalam proses implementasi sistem dan evaluasi kesesuaian fitur yang akan dikembangkan. Hasil prototype tampilan utuh sistem dapat dilihat pada Gambar 3.



Gambar 3 Prototype Low-Fidelity

Berdasarkan Gambar 3, prototype sistem dirancang untuk memberikan gambaran awal mengenai antarmuka dan alur kerja aplikasi deteksi kerentanan website sebelum dilakukan implementasi secara penuh. Tampilan sistem terdiri dari form input URL website yang digunakan untuk memasukkan alamat website target, tombol Mulai Scan untuk menjalankan proses pemindaian, area Detail Temuan untuk menampilkan hasil deteksi kerentanan, serta bagian HTTP Response Headers yang menampilkan informasi konfigurasi website. Selain itu, tersedia fitur unduh laporan dalam format HTML, CSV, dan JSON. Prototype ini digunakan sebagai acuan

dalam proses evaluasi dan pengembangan sistem pada tahap implementasi berikutnya.

Evaluasi Prototype

Setelah prototype selesai dirancang, dilakukan proses evaluasi untuk memastikan bahwa seluruh kebutuhan sistem yang telah diidentifikasi pada tahap pengumpulan kebutuhan telah terakomodasi dengan baik. Evaluasi dilakukan dengan membandingkan fitur yang terdapat pada prototype terhadap kebutuhan fungsional sistem yang telah ditetapkan. Hasil evaluasi prototype dapat dilihat pada Tabel 3.

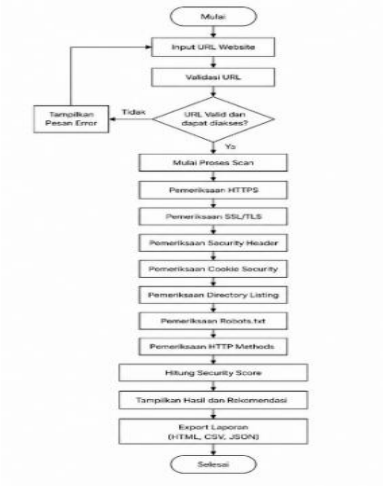
Table 3 Hasil Evaluasi Prototype

No	Kebutuhan Sistem	Implementasi Prototype	Keterangan
1.	Input URL Website	Tersedia textbox URL	Terpenuhi
2.	Proses Scanning	Tersedia tombol Mulai Scan	Terpenuhi
3.	Hasil Analisis	Tersedia panel Detail Temuan	Terpenuhi
4.	Skor Keamanan	Tersedia bagian Security Score	Terpenuhi
5.	Rekomendasi Perbaikan	Tersedia informasi rekomendasi	Terpenuhi
6.	Laporan Pemindaian	Tersedia ekspor HTML, CSV, JSON	Terpenuhi

Berdasarkan hasil evaluasi pada Tabel 3, seluruh kebutuhan fungsional sistem telah berhasil diakomodasi dalam prototype yang dirancang. Fitur input URL website, proses scanning, hasil analisis, tampilan skor keamanan, rekomendasi perbaikan, serta laporan pemindaian telah tersedia sesuai kebutuhan sistem. Dengan demikian, prototype dinyatakan memenuhi kebutuhan penelitian dan dapat dilanjutkan ke tahap implementasi sistem menggunakan bahasa pemrograman Python.

Implementasi Sistem

Setelah prototype dinyatakan sesuai dengan kebutuhan sistem, tahap implementasi dilakukan dengan menerjemahkan seluruh rancangan ke dalam aplikasi berbasis Python. Sistem dikembangkan untuk mendukung proses input URL, pemindaian keamanan website, analisis hasil pemeriksaan, perhitungan Security Score, serta pembuatan laporan hasil pemindaian. Implementasi ini menghasilkan aplikasi yang mampu melakukan analisis keamanan website secara otomatis sesuai kebutuhan pengguna.



Gambar 4 Alur Sistem

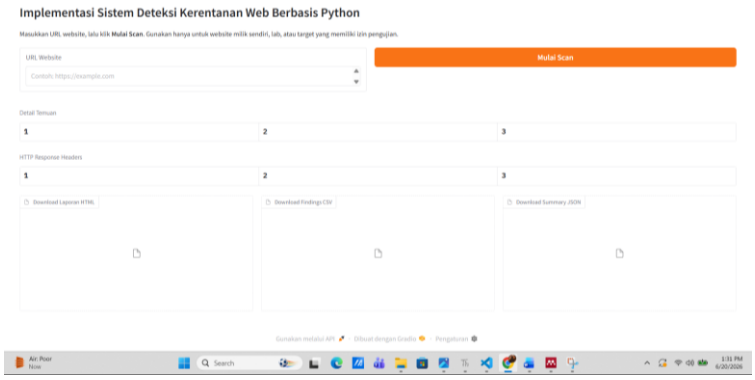
Berdasarkan flowchart pada Gambar 4, proses dimulai dengan pengguna memasukkan URL website yang akan dianalisis. Sistem kemudian melakukan validasi URL untuk memastikan alamat website dapat diakses. Jika URL tidak valid, sistem akan menampilkan pesan kesalahan. Apabila valid, sistem melanjutkan proses pemindaian dengan memeriksa aspek keamanan website, meliputi HTTPS, SSL/TLS, Security Header, Cookie Security, Directory Listing,

Robots.txt, dan HTTP Methods. Hasil pemeriksaan digunakan untuk menghitung skor keamanan website. Selanjutnya, sistem menampilkan hasil analisis beserta rekomendasi perbaikan dan menyediakan fitur ekspor laporan dalam format HTML, CSV, atau JSON.

Berdasarkan alur sistem yang telah dirancang, tahap selanjutnya adalah implementasi sistem menggunakan bahasa pemrograman Python. Implementasi dilakukan dengan menerapkan seluruh proses yang terdapat pada flowchart ke dalam bentuk aplikasi yang dapat melakukan pemindaian keamanan website secara otomatis. Fitur yang diimplementasikan meliputi input URL website, proses scanning, pemeriksaan aspek keamanan, perhitungan skor keamanan, penyajian hasil analisis, serta pembuatan laporan hasil pemindaian.

Halaman Utama

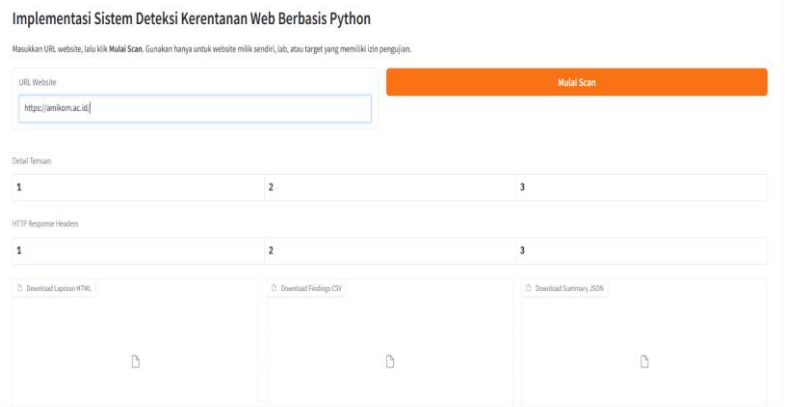
Halaman utama sistem merupakan antarmuka yang digunakan pengguna untuk memasukkan URL website yang akan dianalisis. Pada halaman ini tersedia kolom input URL, tombol Mulai Scan, area hasil temuan, informasi HTTP Response Headers, serta fitur unduh laporan hasil pemindaian, yang dilampirkan pada Gambar 5.



Gambar 5 Halaman Utama

Menginput Link Website yang akan di analisis

Pada tahap ini, pengguna memasukkan alamat atau URL website yang akan dianalisis melalui kolom input yang tersedia pada sistem. URL yang dimasukkan berfungsi sebagai target pemeriksaan keamanan yang akan diproses oleh sistem. Setelah pengguna memasukkan URL, sistem akan melakukan validasi untuk memastikan format alamat website telah sesuai dan dapat diakses. Apabila URL valid, proses selanjutnya dapat dilakukan. Tahap ini menjadi langkah awal yang penting karena menentukan objek website yang akan dilakukan vulnerability assessment, yang dilampirkan pada Gambar 6.



Gambar 6 Input Link Website

Berdasarkan Gambar 6, pengguna memasukkan URL website yang akan dianalisis melalui kolom input yang tersedia pada sistem. Pada penelitian ini, website yang dipilih sebagai objek pengujian adalah amikom.ac.id. Website tersebut digunakan sebagai target pemindaian untuk menguji kemampuan sistem dalam melakukan vulnerability assessment secara otomatis. Setelah URL dimasukkan, sistem akan memproses dan melakukan pemeriksaan terhadap berbagai aspek keamanan website sebelum menampilkan hasil analisis kepada pengguna.

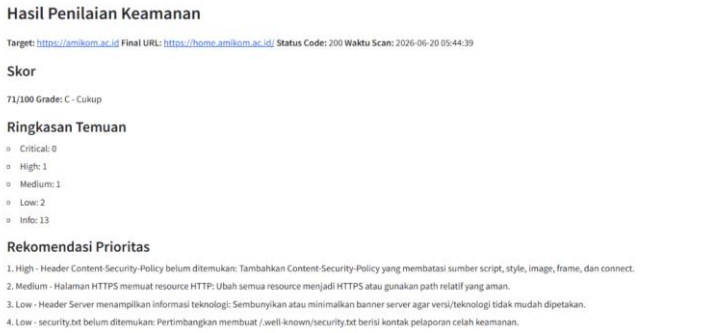
Proses Scanning Pemeriksaan Website

Pada tahap ini, sistem melakukan proses scanning terhadap website yang telah dimasukkan oleh pengguna. Proses scanning dilakukan secara otomatis untuk mengumpulkan informasi dan menganalisis berbagai aspek keamanan website. Pemeriksaan meliputi penggunaan protokol HTTPS, validitas sertifikat SSL/TLS, konfigurasi security header, keamanan cookie, keberadaan file robots.txt, directory listing, serta metode HTTP yang diizinkan oleh server. Selama proses berlangsung, sistem mengirimkan permintaan ke server website dan mengolah respons yang diterima untuk mengidentifikasi potensi kerentanan. Hasil dari proses scanning selanjutnya digunakan sebagai dasar dalam perhitungan skor keamanan dan penyusunan laporan hasil analisis website, yang dilampirkan pada Gambar 7.

Gambar 7 Proses Scanning

Tampilan Security Score dan Rekomendasi

Halaman Security Score dan Rekomendasi menampilkan hasil penilaian keamanan website berdasarkan proses scanning yang telah dilakukan oleh sistem. Pada bagian Security Score, sistem memberikan nilai dalam bentuk persentase atau skor numerik yang menggambarkan tingkat keamanan website secara keseluruhan, mulai dari kategori rendah, sedang, hingga tinggi. Skor ini dihitung berdasarkan indikator seperti kerentanan yang terdeteksi, konfigurasi keamanan, serta potensi risiko yang ditemukan pada website target yang dilampirkan pada Gambar 8.



Gambar 8 Security Score dan Rekomendasi

Berdasarkan hasil pada gambar 8, sistem melakukan penilaian keamanan terhadap website target dan menghasilkan Security Score sebesar 71/100 dengan grade C (Cukup). Artinya, tingkat keamanan website masih berada pada kategori sedang, belum aman sepenuhnya, namun juga tidak dalam kondisi berisiko tinggi secara kritis. Pada bagian Ringkasan Temuan, sistem mendeteksi beberapa tingkat kerentanan, yaitu Critical: 0 (tidak ada celah yang sangat berbahaya), High: 1 (terdapat 1 masalah serius), Medium: 1, Low: 2, dan Info: 13 yang berisi informasi tambahan terkait konfigurasi website. Ini menunjukkan bahwa meskipun tidak ada kerentanan kritis, masih terdapat beberapa celah yang perlu diperbaiki. Pada Rekomendasi Prioritas, sistem menyarankan beberapa perbaikan penting. Pertama, penambahan Content-Security-Policy (CSP) untuk mencegah serangan injeksi seperti script berbahaya. Kedua, perbaikan penggunaan HTTPS karena masih ditemukan resource yang memuat HTTP tidak

aman. Ketiga, menyembunyikan informasi server agar tidak mudah dieksploitasi. Keempat, menambahkan file security.txt sebagai standar pelaporan celah keamanan.

Tampilan Detail Temuan

Menyajikan informasi lebih rinci mengenai hasil pemindaian keamanan website yang telah dilakukan oleh sistem. Pada bagian ini, setiap temuan dikelompokkan berdasarkan tingkat risiko, yaitu Critical, High, Medium, Low, dan Info. Setiap kategori menampilkan jumlah temuan serta penjelasan singkat mengenai jenis kerentanan yang terdeteksi. Detail ini mencakup informasi seperti jenis celah keamanan, lokasi atau parameter yang terdampak, serta dampak potensial yang dapat terjadi apabila kerentanan tersebut tidak diperbaiki. Selain itu, sistem juga memberikan penjelasan teknis sederhana agar mudah dipahami oleh pengguna non-teknis. Dengan adanya tampilan ini, pengguna dapat mengetahui secara spesifik bagian mana dari website yang memiliki kelemahan keamanan. Fitur Detail Temuan ini membantu proses analisis lebih mendalam sehingga pengguna dapat melakukan tindakan perbaikan yang lebih tepat dan terarah berdasarkan tingkat prioritas kerentanan yang ditemukan, yang dilampirkan pada Gambar 9.

Gambar 9 Detail Temuan

Dari hasil pemindaian keamanan pada Gambar 9, website yang dianalisis terlihat tingkat keamanan masih berada pada kategori sedang hingga rendah pada beberapa aspek penting. Website sudah dapat diakses dengan baik (status 200) dan telah menerapkan beberapa security header seperti HSTS (Strict-Transport-Security), X-Frame-Options, X-Content-Type-Options, Referrer-Policy, dan Permissions-Policy. Selain itu, sertifikat TLS masih valid dan koneksi sudah dialihkan ke HTTPS. Namun terdapat beberapa celah yang perlu diperhatikan. Content-Security-Policy belum diterapkan sehingga risiko injeksi script (XSS) masih terbuka. Masih ditemukan penggunaan resource HTTP di halaman HTTPS yang berpotensi menyebabkan mixed content. Informasi server juga masih terlihat sehingga bisa dimanfaatkan untuk fingerprinting. Selain itu, file security.txt belum tersedia dan tidak ada Set-Cookie pada response utama yang menandakan pengelolaan sesi belum optimal. Secara keseluruhan, website sudah cukup aman di level dasar, tetapi masih perlu penguatan pada kebijakan keamanan konten, manajemen cookie, dan pengurangan informasi yang terekspos untuk meningkatkan keamanan secara menyeluruh.

Tampilan Hasil Analisis

Pada tahap ini, sistem menampilkan hasil analisis keamanan website berdasarkan URL yang telah dimasukkan oleh pengguna. Hasil analisis mencakup informasi tingkat kerentanan, kategori risiko, serta rekomendasi perbaikan yang dapat dilakukan untuk meningkatkan keamanan sistem. Tampilan ini dirancang agar mudah dipahami dengan menyajikan data dalam bentuk ringkasan dan detail temuan. Pengguna dapat melihat hasil scanning secara menyeluruh, termasuk header keamanan, potensi celah, dan skor keamanan website. Dengan adanya tampilan ini, pengguna dapat mengevaluasi kondisi keamanan website secara lebih cepat dan akurat untuk mendukung pengambilan keputusan keamanan sistem secara efektif dan efisien, yang dilampirkan pada Gambar 10.


HTTP Response Headers

header	value
Content-Type	text/html; charset=UTF-8
Date	Sat, 20 Jun 2026 05:44:40 GMT
Expires	Sat, 20 Jun 2026 05:44:40 GMT
Keep-Alive	timeout=5, max=100
Link	<https://home.amikom.ac.id/wp-json/>; rel="https://api.w.org/", <https://home.amikom.ac.id/wp-json/wp/v2/pages/3313>; rel="alternate"; title="JSON
Permissions-Policy	accelerometer=(self), autoplay=(self), camera=(self), encrypted-media=(self), fullscreen=(self), geolocation=(self), gyroscope=(self), magnetometer
Referrer-Policy	origin-when-cross-origin
Server	Apache
Strict-Transport-Securi	max-age=2592000
Vary	Accept-Encoding,User-Agent
X-Content-Type-Options	nosniff
X-Frame-Options	sameorigin

Gambar 10 Hasil Analisis

Dari header HTTP pada Gambar 10, bisa disimpulkan bahwa konfigurasi keamanan website yang dianalisis sudah cukup baik di level dasar, tetapi masih ada beberapa aspek penting yang bisa ditingkatkan untuk memperkuat keamanan.

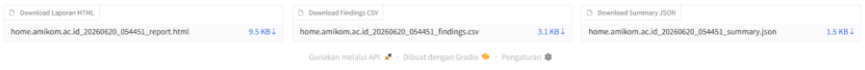
Website ini menggunakan server Apache dengan response Content-Type: text/html; charset=UTF-8 dan kompresi gzip yang membantu performa. Cache-control diset max-age=0 yang berarti halaman selalu diambil ulang (tidak di-cache), ini aman untuk konten dinamis tetapi kurang optimal untuk performa.

Dari sisi keamanan, sudah terdapat beberapa header penting seperti Permissions-Policy yang membatasi akses fitur browser (kamera, geolocation, autoplay, dll), serta Referrer-Policy: origin-when-cross-origin yang cukup aman untuk mengurangi kebocoran informasi referer. Selain itu, endpoint WordPress REST API (/wp-json/) juga terdeteksi, yang menandakan kemungkinan website berbasis WordPress.

Namun, masih terdapat beberapa kelemahan. Header seperti Content-Security-Policy (CSP) belum ada, sehingga website masih rentan terhadap serangan XSS dan injeksi script. Selain itu, tidak terlihat adanya X-Frame-Options di header ini (meskipun mungkin ada di request lain), yang penting untuk mencegah clickjacking. Server juga masih mengungkap informasi “Apache” yang dapat digunakan untuk fingerprinting.

Tampilan Unduh Laporan Hasil Analisis Website

Pada tahap ini, sistem menyediakan fitur untuk mengunduh laporan hasil analisis keamanan website yang telah dilakukan sebelumnya. Laporan disajikan dalam format file yang mudah diakses seperti PDF sehingga pengguna dapat menyimpan, mencetak, atau membagikannya untuk kebutuhan dokumentasi. Isi laporan mencakup ringkasan hasil scanning, detail temuan kerentanan, tingkat risiko, serta rekomendasi perbaikan sistem. Fitur ini memudahkan pengguna dalam melakukan evaluasi lanjutan secara offline dan menjadi bukti hasil analisis keamanan yang telah dilakukan. Dengan adanya fitur unduh laporan ini, proses pelaporan menjadi lebih praktis, terstruktur, dan mendukung kebutuhan audit keamanan website secara berkelanjutan dan profesional.



Gambar 11 Unduh Laporan Hasil Analisis

Berdasarkan Gambar 11, sistem menyediakan fitur unduh laporan hasil analisis website dalam beberapa format file, yaitu CSV, JSON, dan HTML. Ketiga format ini memungkinkan pengguna memilih jenis laporan sesuai kebutuhan, baik untuk analisis lanjutan, integrasi data, maupun dokumentasi visual. Format CSV memudahkan pengolahan data, JSON digunakan untuk kebutuhan sistem atau pengembang, sedangkan HTML menyajikan laporan dalam tampilan yang lebih rapi dan mudah dibaca. Fitur ini meningkatkan fleksibilitas penggunaan data hasil scanning serta mempermudah proses pelaporan, penyimpanan, dan pertukaran informasi hasil analisis keamanan website secara lebih efektif dan efisien.

Pengujian Sistem

Tahap pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur pada aplikasi deteksi kerentanan website berbasis Python dapat berjalan sesuai dengan kebutuhan yang

telah ditentukan. Metode pengujian yang digunakan adalah Black Box Testing, yaitu pengujian yang berfokus pada fungsi sistem berdasarkan masukan (input) dan keluaran (output) tanpa melihat struktur kode program. Pengujian dilakukan pada setiap fitur utama, meliputi input URL website, proses scanning, pemeriksaan HTTPS, validitas sertifikat SSL/TLS, security header, cookie security, directory listing, robots.txt exposure, perhitungan security score, serta pembuatan laporan hasil analisis. Hasil pengujian menunjukkan bahwa seluruh fitur dapat berjalan dengan baik dan menghasilkan output sesuai dengan yang diharapkan, yang dilampirkan pada Tabel.

Table 4 Hasil Pengujian Blackbox Testing

No	Fungsi yang Diuji	Skenario Pengujian	Hasil
1.	Input URL	Memasukkan URL website valid	Berhasil
2.	Validasi URL	Memasukkan URL tidak valid	Berhasil
3.	Proses Scanning	Menjalankan proses pemindaian website	Berhasil
4.	Pemeriksaan HTTPS	Mendeteksi penggunaan HTTPS	Berhasil
5.	SSL/TLS	Memeriksa validitas sertifikat SSL/TLS	Berhasil
6.	Security Header	Menganalisis header keamanan website	Berhasil
7.	Cookie Security	Memeriksa atribut keamanan cookie	Berhasil
8.	Directory Listing	Mendeteksi directory listing	Berhasil
9.	Robots.txt	Memeriksa paparan robots.txt	Berhasil
10.	Security Score	Menghasilkan skor keamanan website	Berhasil
11.	Laporan Analisis	Mengunduh laporan HTML, CSV, dan JSON	Berhasil

Berdasarkan hasil pengujian pada Tabel 4, seluruh fungsi sistem dapat berjalan sesuai dengan kebutuhan yang telah dirancang. Sistem berhasil melakukan pemindaian keamanan website secara otomatis, menampilkan hasil analisis, memberikan rekomendasi perbaikan, serta menghasilkan laporan dalam berbagai format. Dengan demikian, sistem dinyatakan layak digunakan sebagai alat bantu vulnerability assessment website.

Sistem Final

Tahap sistem final merupakan tahap akhir setelah seluruh proses implementasi dan pengujian berhasil dilakukan. Pada tahap ini sistem telah melalui proses evaluasi, perbaikan, dan penyempurnaan sehingga seluruh fitur dapat berjalan secara optimal sesuai kebutuhan pengguna. Sistem yang dihasilkan mampu melakukan analisis keamanan website secara otomatis dengan memeriksa berbagai aspek keamanan, seperti penggunaan HTTPS, validitas sertifikat SSL/TLS, konfigurasi security header, keamanan cookie, directory listing, robots.txt exposure, serta metode HTTP yang digunakan oleh server.

Selain melakukan deteksi kerentanan, sistem juga mampu menghitung Security Score,

mengelompokkan temuan berdasarkan tingkat risiko (Critical, High, Medium, Low, dan Info), serta memberikan rekomendasi mitigasi yang dapat digunakan sebagai acuan perbaikan keamanan website. Hasil analisis dapat disimpan dalam format HTML, CSV, dan JSON sehingga memudahkan proses dokumentasi dan evaluasi lanjutan. Dengan adanya sistem final ini, proses vulnerability assessment dapat dilakukan secara lebih cepat, efisien, dan konsisten dibandingkan metode pemeriksaan manual, sehingga dapat membantu administrator maupun pengembang website dalam meningkatkan tingkat keamanan website secara berkelanjutan.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, berhasil diimplementasikan sistem vulnerability scanner berbasis Python untuk analisis kerentanan dan rekomendasi mitigasi keamanan website menggunakan metode Prototype. Sistem yang dikembangkan mampu melakukan pemeriksaan berbagai aspek keamanan website secara otomatis, meliputi penggunaan HTTPS, validitas sertifikat SSL/TLS, security header, keamanan cookie, directory listing, robots.txt exposure, serta metode HTTP yang digunakan server. Hasil pengujian menggunakan metode Black Box Testing menunjukkan bahwa seluruh fitur sistem dapat berjalan sesuai dengan kebutuhan yang telah ditetapkan. Sistem juga mampu menghasilkan Security Score, mengelompokkan temuan berdasarkan tingkat risiko, serta memberikan rekomendasi perbaikan yang dapat digunakan sebagai acuan peningkatan keamanan website. Selain itu, sistem menyediakan fitur ekspor laporan dalam format HTML, CSV, dan JSON sehingga memudahkan proses dokumentasi dan evaluasi keamanan. Dengan demikian, sistem yang dikembangkan dapat membantu proses vulnerability assessment secara lebih cepat, efisien, dan konsisten dibandingkan pemeriksaan manual serta mendukung peningkatan keamanan website secara berkelanjutan.

REFERENSI

Ainurrohman, Muhamad, dan Yuli Nurasri. 2025. "Pengujian Keamanan Website JDIH Kab . Tegal Menggunakan Acunetix dengan Standar ISO / IEC / 27001 : 2013." 2(3):3320–23.

Akbar, Muhammad Rifky. 2026. "PENGEMBANGAN SISTEM INFORMASI CSIRT BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL DENGAN METODE PROTOTYPING PADA DISKOMINFO PURWAKARTA." 12(1):77–84.

Anugrah, Suci, Nabila Raihana Qalby, Muhammad Zaky Himawan, dan Evy Nurmiati. 2025. "Pengaruh Etika Profesi Terhadap Keamanan Informasi dalam Konteks Kebocoran Data BSI (Bank Syariah Indonesia): Studi Literatur Sistematis The Influence of Professional Ethics on Information Security in the Context of the BSI Data Breach : A Systematic Literature Study." 11:106–12.

Anwari, Zulvan Avito, I. Gede Pasek Wedana, dan Jaya Deva. 2022. "ANALISIS KERENTANAN PADA SUATU WEBSITE MENGGUNAKAN TOOLS XSPEAR, XSSCON, DAN PWNXSS." 4(4):406–12.

Ariyadi, Tamsir, dan Hidayatul Fadli. 2025. "Implementasi OWASP untuk Analisis Kerentanan dan Keamanan pada Sistem Informasi Akademik Terintegrasi Universitas Bina Darma." 4(1):1–7.

Chowdhury, Mudassor Ahmed, Mushfiqur Rahman, dan Sifatnur Rahman. 2024. "Detecting vulnerabilities in website using multiscale approaches : based on case study." 14(3):2814–21. doi:10.11591/ijece.v14i3.pp2814-2821.

Corradini, Davide. 2025. "Automated Automated Testing Testing of of Broken Broken Authentication Authentication Vulnerabilities Vulnerabilities in in Web Web APIs APIs with with AuthREST AuthREST."

Djabar, Oktavia. 2025. "Cyber Phishing dan Penegakan Hukumnya : Peran Krusial Penyidik Menurut UU ITE." 2(2):214–30.

Fauzi, Ahmad, Rohmat Taufiq, Dyas Yudi Priyanggodo, Rachmat Destriana, Teknik Informatika, Fakultas Teknik, dan Universitas Muhammadiyah Tangerang. 2026. "Pengembangan Sistem Informasi Pengelolaan Distribusi Air Bersih Berbasis Web dengan Menggunakan Metode Prototype Di Desa Cijengkol." 15(01):10–20.

Giawa, Yutesia, Sahara Abdy, dan Tomy Satria Alasi. 2026. "Jurnal Informatika Logika Perancangan Sistem Informasi Surat Masuk Dan Surat Keluar Secara Multiuser Menggunakan Metode Prototype Pada Kantor Desa Fondrakoraya Kabupaten Nias Selatan Jurnal Informatika Logika." 3:14–20.

He, Zehai, Wenyi Hong, Zhen Yang, Ziyang Pan, Mingdao Liu, Xiaotao Gu, dan Jie Tang. 2026. "Vision2Web: A Hierarchical Benchmark for Visual Website Development with Agent Verification."

Irawan, Indra, Avrillaila Akbar Harahap, Dadang Heksaputra, Tri Rochmadi, Program Studi, Sistem Informasi, Universitas Alma Ata, Black Box Testing, dan Sahabat Phone. 2026. "ELEKTRONIK BERBASIS WEBSITE PADA SAHABAT PHONE." 3:61–70. doi:10.61124/sinta.v3i1.161.

Issaf, Menggunakan Framework, dan Muhammad Koprawi. 2025. "Analisis Keamanan Website Pada Instansi XYZ Melalui Penetration Testing." 5(1):547–55.

Jayanto, Gunawan Yudi. 2025. "Analisis Celah Keamanan Website Poltekkes Kemenkes Sorong Menggunakan Metode Penetration Testing." 15(1):136–50.

Jehian, Neysa Talitha, Dedy Kiswanto, Muhammad Rizki, Andrian Fitra, Hansel Valent, Universitas Negeri Medan, dan Kabupaten Deli Serdang. 2025. "DATA WEB KOMBINASI ALGORITMA CHACHA20-POLY1305 DAN ARGON2." 13(3).

Munandar, Richard Alvin, Pratyaksa Ocsa, dan Nugraha Saian. 2026. "IT-EXPLORE Perancangan sistem manajemen session real-time menggunakan firestore , python , dan javascript." 05(52):50–64. doi:10.24246/itexplore.v5i1.2026.pp50-64.

Nainggolan, Fritti Juliana. 2026. "ANALISIS DAN PERANCANGAN UI/UX PROTOTIPE SISTEM INFORMASI PENCATATAN KEUANGAN DAN KEANGGOTAAN KOPERASI SABDA KARYA SEJAHTERA BERBASIS WEB MENGGUNAKAN METODE PROTOTYPE." 12(1).

Okonkwo, Paul C., Samuel Chukwujindu, Sunday O. Udo, Anthony Umunnakwe, Usang Nkanu, Saad S. Alarifi, Ahmed M. Eldosouky, Stephen E. Ekwok, dan Peter Andr. 2025. "Solar PV systems under weather extremes : Case studies , classification , vulnerability assessment , and adaptation pathways." 13(December 2024):929–59. doi:10.1016/j.egyr.2024.12.067.

Putrie, Dewi Sonia, Aditya Dwi, Putro Wicaksono, dan Gunawan Wibisono. 2026. "Jurnal Teknologi Informasi dan Multimedia Rancang Bangun Aplikasi Laporan Keuangan Berbasis Web Dengan Metode Prototype (Studi Kasus : Praktik Mandiri Bidan Nining)." 8(1):138–50.

Rizkayanti, Tara, dan W. Yunanri. 2023. "ANALISIS KEAMANAN WEBSITE SISTEM INFORMASI ADMINISTRASI KEPENDUDUKAN MENGGUNAKAN METODE VULNERABILITY ASSESMENT." 1(1).

Sales, ArVilane Gonçalves, dan Data Ticle. 2025. "A coastal hospitality sector database for vulnerability assessments in Veneto and." 62. doi:10.1016/j.dib.2025.111921.

Shidqi, Shafirina Arsydini, Gustin Setyaningsih, Ika Romadoni Yunita, Muhamad Awiet, Wiedanto Prasetyo, Universitas Amikom Purwokerto, Jawa Tengah, Universitas Amikom Purwokerto, Jawa Tengah, Sistem Informasi, Universitas Amikom Purwokerto, Jawa Tengah, Sistem Informasi, Universitas Telkom, dan Jawa Tengah. 2025. "PERANCANGAN WEBSITE PENJUALAN MENGGUNAKAN METODE PROTOTYPE PADA." 2(3):39–53.

Siddiq, Hasan, Sigit Auliana, Mochammad Darip, Ilmu Komputer, Ilmu Komputer, dan Universitas Bina Bangsa. 2025. "PERANCANGAN WEBSITE E-TOURISM SEBAGAI MEDIA PROMOSI DIGITAL PARIWISATA DI KECAMATAN PULOSARI MENGGUNAKAN FRAMEWORK." 11(2):192–99.