

PERANCANGAN APLIKASI PENETRATION TESTING UNTUK KEAMANAN SISTEM INFORMASI BERBASIS WEBSITE STANDAR OWASP ZAP

Irham Muhammad Ghiffari^{1*}, Dayan Singasatia², Syariful Alam³

^{1,2,3} Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana; Jl. Cikopak No. 53, Purwakarta, Jawa Barat

Keywords:

Penetration Testing;
SQL Injection;
Cross-Site Scripting (XSS);
OWASP ZAP; Black Box
Testing.

Correspondent Email:

irham.ghiffari@sttwastukencana.
ac.id



Copyright © 2026 The Author(s),
Published by Jurnal Informatika dan
Teknik Elektro Terapan. This is an
open-access article distributed under
the terms of the Creative Commons
Attribution-NonCommercial 4.0
International License (CC BY-NC
4.0).

Abstrak. Perkembangan sistem informasi berbasis website meningkatkan risiko keamanan siber, terutama kerentanan SQL Injection dan Cross-Site Scripting (XSS) yang dapat menyebabkan kebocoran serta manipulasi data. Pengujian keamanan secara manual membutuhkan waktu lama dan keahlian teknis memadai. Penelitian ini merancang dan membangun aplikasi penetration testing berbasis web menggunakan Python untuk mendeteksi SQL Injection dan XSS mengacu standar OWASP. Metode pengembangan yang digunakan adalah Waterfall dengan perancangan UML meliputi Use Case, Activity, dan Sequence Diagram, sedangkan pengujian menggunakan Black Box Testing. Hasil penelitian menunjukkan aplikasi mampu melakukan input URL, scanning, pengujian kerentanan, dan menampilkan hasil pengujian, dengan seluruh fungsi utama berjalan sesuai kebutuhan.

Abstract. The development of website-based information systems increases cybersecurity risks, particularly SQL Injection and Cross-Site Scripting (XSS) vulnerabilities that can cause data leakage and manipulation. Manual security testing requires considerable time and adequate technical expertise. This research designs and builds a web-based penetration testing application using Python to detect SQL Injection and XSS referring to OWASP standards. The development method used is Waterfall with UML design covering Use Case, Activity, and Sequence Diagrams, while testing uses Black Box Testing. The results show the application can perform URL input, scanning, vulnerability testing, and display results, with all main functions running as required.

1. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat telah mendorong penggunaan sistem informasi berbasis website dalam berbagai bidang, seperti pendidikan, pemerintahan, bisnis, dan layanan publik. Seiring meningkatnya penggunaan aplikasi web, aspek keamanan sistem menjadi sangat penting karena website sering menyimpan data sensitif yang harus dijaga kerahasiaan, integritas, dan ketersediaannya [1]. Di sisi lain, meningkatnya penggunaan aplikasi web juga diikuti dengan meningkatnya ancaman keamanan siber. Dua jenis kerentanan yang paling umum ditemukan pada aplikasi web

adalah SQL Injection, yang memungkinkan penyerang memanipulasi perintah SQL melalui input pengguna, dan Cross-Site Scripting (XSS), yang memungkinkan penyerang menyisipkan script berbahaya pada browser pengguna sehingga berpotensi menyebabkan pencurian data maupun pengambilalihan sesi pengguna [2].

Berdasarkan OWASP Top 10:2021, kategori Injection masih menjadi salah satu risiko utama pada aplikasi web, dengan 94% aplikasi yang diuji memiliki bentuk injection, tingkat kejadian maksimum 19%, rata-rata 3,37%, dan 274 ribu occurrence pada data referensi OWASP [3]. Untuk mengurangi risiko tersebut, diperlukan proses pengujian keamanan secara

berkala, salah satunya melalui penetration testing, yaitu metode pengujian keamanan dengan mensimulasikan serangan terhadap sistem untuk mengidentifikasi kerentanan yang dapat dimanfaatkan oleh penyerang sebelum dieksploitasi [4].

Berbagai penelitian menunjukkan bahwa pengujian keamanan berbasis standar OWASP mampu membantu proses identifikasi kerentanan pada aplikasi web. Namun, sebagian besar penelitian sebelumnya masih berfokus pada penggunaan tools yang telah tersedia, seperti OWASP ZAP. Meskipun tools tersebut efektif dalam mendeteksi berbagai kerentanan keamanan, pengguna umumnya hanya memperoleh hasil pengujian tanpa memahami secara mendalam mekanisme scanning dan proses deteksi yang dilakukan; studi benchmarking terhadap versi terbaru ZAP juga menunjukkan bahwa performa scanner perlu terus dievaluasi agar tetap relevan dengan kebutuhan pengujian yang lebih luas [5]. Kondisi ini menunjukkan bahwa masih diperlukan pengembangan aplikasi penetration testing yang tidak hanya berfungsi sebagai alat deteksi, tetapi juga sebagai sarana pembelajaran untuk memahami mekanisme dasar pengujian keamanan aplikasi web.

Berdasarkan kondisi tersebut, penelitian ini mengembangkan aplikasi penetration testing berbasis web menggunakan bahasa pemrograman Python dengan fokus pada deteksi kerentanan SQL Injection dan Cross-Site Scripting (XSS). Python dipilih karena memiliki sintaks yang sederhana, mudah dipelajari, serta mendukung pengembangan aplikasi web dan otomatisasi request secara fleksibel. Pengujian dilakukan pada website uji berbasis localhost sehingga proses pengujian dapat dilakukan dalam lingkungan yang aman dan terkontrol. Rumusan masalah penelitian ini adalah bagaimana merancang dan membangun aplikasi penetration testing berbasis web menggunakan Python untuk mendeteksi SQL Injection dan XSS, serta bagaimana hasil pengujian aplikasi tersebut pada website localhost menggunakan Black Box Testing. Tujuan penelitian ini adalah merancang dan membangun aplikasi tersebut, serta menguji dan mengevaluasi kemampuannya dalam mendeteksi kerentanan pada website localhost menggunakan metode Black Box Testing.

2. TINJAUAN PUSTAKA

2.1. *Sistem Informasi Berbasis Website*

Sistem informasi berbasis website merupakan sistem yang memanfaatkan teknologi web untuk mengelola, memproses, dan menyajikan informasi kepada pengguna melalui jaringan internet, umumnya terdiri atas antarmuka pengguna, logika aplikasi, serta basis data yang saling terintegrasi melalui protokol HTTP/HTTPS, dengan pengembangan yang kini banyak mengadopsi pendekatan full stack agar sisi client dan server dapat dibangun secara terpadu [6]. Meningkatnya penggunaan sistem berbasis website juga diiringi dengan meningkatnya potensi ancaman keamanan akibat kesalahan pengembangan, kurangnya validasi input, serta konfigurasi keamanan yang tidak optimal, sehingga diperlukan mekanisme pengujian keamanan seperti penetration testing untuk mengidentifikasi potensi celah keamanan sebelum sistem digunakan secara luas [7].

2.2. *Keamanan Sistem Informasi*

Keamanan sistem informasi merupakan upaya untuk melindungi sistem komputer, jaringan, dan data dari berbagai ancaman yang dapat mengganggu kerahasiaan (confidentiality), integritas (integrity), dan ketersediaan (availability) informasi, yang dikenal dengan istilah CIA Triad. Dalam konteks aplikasi berbasis website, keamanan sistem menjadi aspek penting karena sistem dapat diakses secara luas melalui internet, sehingga rentan terhadap berbagai serangan siber seperti manipulasi input dan penyisipan script berbahaya [8].

2.3. *Penetration Testing*

Penetration testing merupakan metode pengujian keamanan sistem yang dilakukan dengan mensimulasikan serangan terhadap aplikasi atau jaringan untuk mengidentifikasi kerentanan yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab. Dalam praktiknya, penetration testing dilakukan melalui tahapan information gathering, scanning, exploitation, serta analysis dan reporting. Salah satu pendekatan yang digunakan adalah Black Box Testing, yaitu pengujian tanpa mengetahui struktur internal sistem, sehingga sesuai untuk mensimulasikan

serangan dari sudut pandang penyerang eksternal [4].

2.4. OWASP ZAP dan OWASP Top 10

OWASP ZAP (Zed Attack Proxy) adalah alat open-source yang digunakan untuk menguji keamanan aplikasi web melalui pendekatan passive scanning, active scanning, dan intercepting proxy, membantu menemukan kerentanan seperti misconfiguration, information disclosure, dan kelemahan header keamanan. Salah satu referensi utama dalam pengujian keamanan aplikasi web adalah OWASP Top 10, yaitu daftar sepuluh kerentanan keamanan aplikasi web yang paling umum dan berisiko tinggi, yang diperbarui secara berkala berdasarkan perkembangan ancaman keamanan terbaru. Kerentanan seperti SQL Injection dan Cross-Site Scripting (XSS) termasuk dalam kategori utama OWASP Top 10, sehingga pengujian terhadap kedua jenis serangan tersebut menjadi relevan dalam penelitian ini [3].

2.5. Kerentanan Keamanan Aplikasi Web

Kerentanan keamanan aplikasi web adalah kelemahan pada sistem yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab untuk memperoleh akses tidak sah, mencuri data, atau mengganggu layanan sistem, yang umumnya muncul akibat kurangnya validasi input, kesalahan konfigurasi, serta penerapan keamanan yang belum optimal. Dua jenis kerentanan yang sering ditemukan pada aplikasi web adalah SQL Injection dan Cross-Site Scripting (XSS) [9].

2.6. SQL Injection

SQL Injection merupakan salah satu jenis serangan pada aplikasi web yang memanfaatkan kelemahan dalam proses pengolahan input pengguna yang berinteraksi dengan database, di mana penyerang dapat menyisipkan perintah SQL berbahaya ke dalam query yang dijalankan oleh sistem apabila input tidak divalidasi dengan baik [10]. Salah satu contoh payload yang sering digunakan dalam pengujian SQL Injection adalah ' OR '1'='1, karena dapat mengubah logika query dan berpotensi memberikan akses tanpa autentikasi yang valid [11]. SQL Injection dapat menimbulkan berbagai dampak terhadap keamanan sistem, seperti pencurian data,

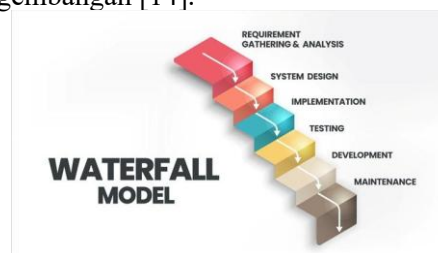
manipulasi isi database, hingga pengambilalihan akses sistem, sehingga deteksinya penting dilakukan dengan menyisipkan payload tertentu pada parameter input untuk mengidentifikasi kelemahan validasi input yang dapat dimanfaatkan untuk manipulasi query database [12].

2.7. Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) merupakan salah satu jenis serangan pada aplikasi web yang terjadi ketika aplikasi menampilkan input pengguna tanpa proses validasi atau penyaringan yang memadai, sehingga penyerang dapat menyisipkan script berbahaya yang akan dijalankan pada browser pengguna untuk mencuri data, memanipulasi tampilan, atau mengarahkan pengguna ke halaman berbahaya. XSS dibagi menjadi Reflected XSS, Stored XSS, dan DOM-Based XSS. Salah satu contoh payload yang digunakan dalam pengujian XSS adalah `<script>alert('XSS')</script>`; jika script tersebut berhasil dijalankan oleh browser, maka sistem berpotensi rentan terhadap XSS [13].

2.8. Metode Waterfall

Model Waterfall merupakan paradigma pengembangan perangkat lunak dengan pendekatan sekuensial dan sistematis, di mana setiap tahapan dilakukan secara berurutan dan harus diselesaikan sebelum melanjutkan ke tahap berikutnya, sehingga bersifat linear dan terstruktur meskipun kurang fleksibel terhadap perubahan kebutuhan selama proses pengembangan [14].



Gambar 1. Model Waterfall

Tahapan metode Waterfall terdiri dari requirement gathering and analysis untuk menganalisis kebutuhan sistem, system design untuk merancang arsitektur dan alur kerja sistem, implementation untuk mengembangkan sistem sesuai desain, testing untuk memastikan sistem berjalan sesuai kebutuhan, deployment untuk penerapan sistem, serta maintenance

untuk perbaikan dan pengembangan sistem setelah digunakan.

2.9. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk menggambarkan struktur, perilaku, dan interaksi dalam suatu sistem berbasis objek, sehingga membantu pengembang memahami kebutuhan sistem secara lebih terstruktur [15]. Use Case Diagram digunakan untuk memodelkan kebutuhan sistem berdasarkan interaksi antara aktor dan sistem serta membantu menentukan batasan sistem dan fungsi utama yang harus tersedia [16]. Activity Diagram digunakan untuk memodelkan alur aktivitas dalam suatu sistem secara terstruktur, menggambarkan bagaimana proses dimulai, keputusan diambil, dan proses berakhir [17]. Sequence Diagram menggambarkan interaksi antar objek dalam sistem berdasarkan urutan waktu melalui pertukaran pesan [18]. Class Diagram digunakan untuk menggambarkan class, atribut, metode, serta hubungan antar class dalam sebuah sistem, sehingga membantu perancang memvisualisasikan struktur aplikasi berorientasi objek secara lebih jelas dan terorganisir [19].

2.10. Pemrograman Python dan HTML

Python adalah bahasa pemrograman tingkat tinggi dengan sintaks sederhana dan mudah dipahami, mendukung berbagai paradigma pemrograman serta memiliki banyak library yang dapat digunakan dalam pengembangan aplikasi web, otomatisasi sistem, dan keamanan informasi, sehingga populer digunakan dalam pengembangan sistem informasi. HyperText Markup Language (HTML) merupakan bahasa markup standar yang digunakan untuk membangun struktur dasar halaman web, menampilkan berbagai elemen seperti teks, gambar, tabel, dan form input yang dapat diakses melalui browser, serta dapat dikombinasikan dengan CSS dan JavaScript untuk menghasilkan tampilan yang lebih interaktif dan dinamis [20]. Dalam penelitian ini, HTML digunakan sebagai bagian dari antarmuka aplikasi penetration testing yang dikembangkan menggunakan Python untuk menampilkan halaman input URL target, tombol scanning, serta hasil deteksi kerentanan.

2.11. Black Box Testing

Black Box Testing merupakan salah satu metode pengujian perangkat lunak yang berorientasi pada fungsi sistem, di mana pengujian dilakukan tanpa mengetahui struktur internal program. Metode ini bertujuan memastikan bahwa sistem dapat menerima input, memproses data, dan menghasilkan output yang sesuai dengan spesifikasi, serta digunakan untuk menemukan kesalahan pada fungsi sistem seperti kesalahan logika, kesalahan input, dan ketidaksesuaian output yang dihasilkan [21].

2.12. Penelitian Terkait

Beberapa penelitian terdahulu yang relevan dengan penelitian ini dirangkum pada Tabel 1.

Tabel 1. Penelitian Terkait

Peneliti (Tahun) & Judul	Metode & Fokus	Perbedaan dengan Penelitian Ini
Ariyadi dkk. (2024) [22] – Implementasi Secure Code pada Pengembangan Sistem Keamanan Website Menggunakan Penetration Testing dan OWASP ZAP	Penetration Testing, OWASP ZAP; identifikasi SQL Injection dan XSS	Penelitian ini mengembangkan aplikasi pentest mandiri berbasis Python, bukan hanya menggunakan tools yang telah tersedia
Hidayat & Nugroho (2025) [23] – Analisis Celah Keamanan pada Website SMA Negeri 3 Berau dengan Metode Penetration Testing	Penetration Testing, WPScan, OWASP ZAP	Penelitian ini berfokus pada perancangan aplikasi deteksi mandiri, bukan analisis satu website studi kasus dengan tools eksisting
Abdullah & Koprari (2025) [24] – Analisis Keamanan Website Melalui Penetration Testing Menggunakan Framework ISSAF dan OWASP	ISSAF Framework, OWASP ZAP	Penelitian ini menggunakan pendekatan Black Box Testing dengan aplikasi buatan sendiri, bukan framework ISSAF
Marta Dinata dkk. (2025) [25] – Analisis Keamanan Situs Web Menggunakan Metode Penetration Testing OWASP	OWASP Testing Guide	Penelitian ini menghasilkan aplikasi pengujian yang dapat digunakan ulang, bukan hanya laporan analisis satu situs
Arifai Nurrizki & Iman Heri Ujianto (2024) [26] – Analisis Keamanan Website Desa Budaya Menggunakan Metode Penetration Testing dan OWASP ZAP	Penetration Testing, OWASP ZAP	Penelitian ini berfokus pada pengembangan aplikasi deteksi SQL Injection dan XSS secara khusus, bukan audit keamanan umum satu situs

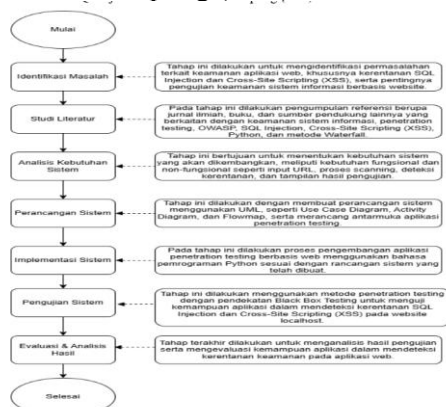
Berdasarkan penelitian terdahulu, sebagian besar penelitian masih menggunakan tools penetration testing yang telah tersedia seperti OWASP ZAP dalam proses pengujian keamanan website dan lebih berfokus pada

analisis kerentanan sistem. Penelitian ini berbeda karena tidak hanya melakukan pengujian keamanan, tetapi juga mengembangkan aplikasi penetration testing sederhana berbasis Python yang dapat digunakan sebagai media pembelajaran dalam memahami mekanisme deteksi kerentanan SQL Injection dan Cross-Site Scripting (XSS).

3. METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian dalam pengembangan aplikasi penetration testing dilakukan secara sistematis menggunakan metode Waterfall, sebagaimana disajikan pada Gambar 2. Metode ini dipilih karena memiliki alur pengembangan yang terstruktur dan berurutan, sehingga setiap tahapan dapat dilaksanakan dengan lebih terarah mulai dari analisis kebutuhan hingga pengujian sistem, dengan proses pengujian keamanan mengacu pada standar OWASP Top 10, khususnya kerentanan SQL Injection dan Cross-Site Scripting (XSS).



Gambar 2. Tahapan Penelitian

3.2. Objek Penelitian

Objek penelitian ini adalah website uji berbasis web yang digunakan sebagai media pengujian keamanan menggunakan aplikasi penetration testing yang dikembangkan. Website uji dijalankan pada lingkungan localhost dan dibangun menggunakan basis data seperti MySQL, dengan beberapa fitur yang menerima input dari pengguna, seperti form login dan form input data, sebagai titik pengujian untuk mendeteksi potensi kerentanan SQL Injection dan Cross-Site Scripting (XSS).

3.3. Metode Penelitian

Metode penelitian yang digunakan adalah penetration testing dengan pendekatan Black Box Testing serta metode Waterfall sebagai model pengembangan sistem, meliputi analisis kebutuhan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional terkait proses scanning dan deteksi kerentanan; perancangan sistem menggunakan UML yang meliputi Use Case Diagram, Activity Diagram, dan Sequence Diagram serta perancangan antarmuka; implementasi sistem berupa pembangunan aplikasi menggunakan bahasa pemrograman Python dan framework Flask sesuai rancangan; pengujian sistem menggunakan pendekatan Black Box Testing dari sisi pengguna untuk menguji kemampuan sistem mendeteksi kerentanan pada website localhost; serta evaluasi sistem untuk menganalisis hasil pengujian dan mengevaluasi kemampuan aplikasi dalam mendeteksi SQL Injection dan Cross-Site Scripting (XSS).

4. HASIL DAN PEMBAHASAN

4.1. Analisis Sistem Berjalan dan Kebutuhan Sistem

Pada sistem yang berjalan, proses pengujian keamanan website masih dilakukan secara manual oleh pentester, mulai dari identifikasi halaman web, form input, dan parameter URL, hingga pengujian menggunakan berbagai payload SQL Injection dan Cross-Site Scripting (XSS) secara manual, sebagaimana digambarkan pada flowmap sistem berjalan pada Gambar 3.



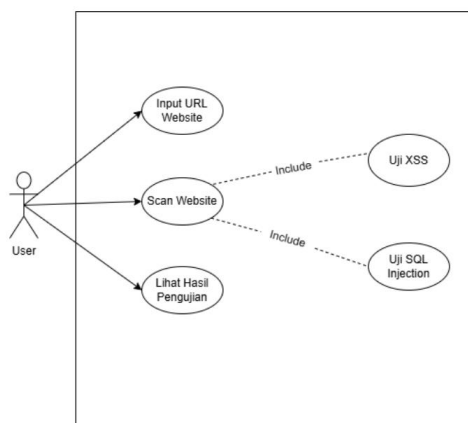
Gambar 3. Flowmap Sistem Berjalan

Berdasarkan hasil analisis, sistem yang berjalan saat ini membutuhkan waktu yang relatif lama karena identifikasi parameter,

pengiriman payload, serta analisis respons harus dilakukan secara berulang pada setiap halaman yang diuji, dan sangat bergantung pada kemampuan teknis pentester sehingga berpotensi menimbulkan kesalahan dalam proses identifikasi maupun pelaporan hasil pengujian. Berdasarkan kebutuhan fungsional yang telah diidentifikasi, sistem usulan dirancang mampu menerima input URL website target, melakukan validasi URL, melakukan proses scanning terhadap halaman website, mendeteksi kerentanan SQL Injection dan Cross-Site Scripting (XSS) menggunakan payload tertentu, serta menampilkan hasil scanning beserta status keamanan website kepada pengguna.

4.2. Perancangan Sistem

Perancangan sistem dilakukan menggunakan Unified Modeling Language (UML). Use Case Diagram pada Gambar 4 menggambarkan interaksi antara aktor User dengan sistem, di mana pengguna dapat memasukkan URL website target, menjalankan proses scanning yang mencakup pengujian SQL Injection dan Cross-Site Scripting (XSS) melalui relasi include, serta melihat hasil pengujian yang ditampilkan oleh sistem.



Gambar 4. Use Case Diagram

Selain Use Case Diagram, perancangan sistem juga meliputi Activity Diagram untuk menggambarkan alur input URL website, proses scanning, dan penampilan hasil pengujian; Sequence Diagram untuk menggambarkan interaksi antara user, aplikasi, dan website target berdasarkan urutan waktu; serta Class Diagram yang terdiri atas kelas User, ScannerApp, SQL Injection Test, XSSTest, dan Result untuk menggambarkan struktur dan hubungan antar komponen sistem.

Perancangan antarmuka meliputi halaman utama untuk konfigurasi pengujian, halaman proses scanning, serta halaman hasil pengujian yang menampilkan status keamanan website.

4.3. Implementasi Sistem

Implementasi sistem dilakukan dengan membangun aplikasi penetration testing berbasis website menggunakan bahasa pemrograman Python dengan framework Flask. Aplikasi yang dikembangkan mampu menerima input URL website target, memilih jenis pengujian keamanan, menjalankan proses scanning, serta menampilkan hasil pengujian secara otomatis. Implementasi algoritma pengujian meliputi validasi URL, identifikasi form HTML dan parameter URL, pengujian SQL Injection dengan menganalisis pola kesalahan database pada respons website, pengujian Cross-Site Scripting (XSS) dengan memeriksa apakah payload ditampilkan kembali tanpa penyaringan, hingga penyajian hasil pengujian berupa status koneksi, jumlah form dan parameter yang teridentifikasi, jumlah payload yang diuji, jumlah kerentanan yang ditemukan, dan tingkat risiko keamanan website. Tampilan implementasi halaman hasil pengujian, yang merekapitulasi hasil Black Box Testing sesuai standar OWASP Top 10, disajikan pada Gambar 5.

Tabel Hasil Black Box Testing
Rekapitulasi hasil pengujian keamanan menggunakan metode Black Box Testing sesuai OWASP Top 10

Tanggal Pengujian: 10 Juni 2024, 12:26:38 | Target: <http://demo.testfire.net/bank/login.aspx> | Metode: Black Box Testing
Standar: OWASP Top 10

NO	KASUS UI	JENIS SERANGAN	INPUT / PAYLOAD	YANG DIAMPAIKAN	HASIL PENGUJIAN	STATUS	KERENTANAN
1	SQL-01	SQL Injection	Salah payload & [uji]	Sistem menolak input dan tidak menampilkan error database	Tidak ditemukan respons error database dari payload yang dikirim	✓ BERHASIL	AMAN
2	XSS-01	Cross-Site Scripting (XSS)	<script>alert(1)</script>	Sistem menolak script dan tidak memfuskikan ke halaman	Script payload ter-reflect pada halaman target (HTTP 200) pada form: reset.php	✗ GAGAL	BERENTAN
3	XSS-02	Cross-Site Scripting (XSS)		Sistem menolak script dan tidak memfuskikan ke halaman	Script payload ter-reflect pada halaman target (HTTP 200) pada form: reset.php	✗ GAGAL	BERENTAN
4	XSS-03	Cross-Site Scripting (XSS)	<svg onmouseover=alert(1)>	Sistem menolak script dan tidak memfuskikan ke halaman	Script payload ter-reflect pada halaman target (HTTP 200) pada form: reset.php	✗ GAGAL	BERENTAN
5	XSS-04	Cross-Site Scripting (XSS)	<javascript:alert(1)>	Sistem menolak script dan tidak memfuskikan ke halaman	Script payload ter-reflect pada halaman target (HTTP 200) pada form: reset.php	✗ GAGAL	BERENTAN
6	XSS-	Cross-Site	<body onload=alert(1)>	Sistem menolak script	Script payload ter-	✗ GAGAL	BERENTAN

Gambar 5. Implementasi Halaman Hasil Pengujian

4.4. Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode Black Box Testing yang berfokus pada fungsi sistem tanpa memperhatikan struktur kode program. Pengujian dilakukan dengan memberikan berbagai masukan pada sistem dan mengamati keluaran yang dihasilkan terhadap enam fungsi utama, yaitu input URL website,

validasi URL, scan website, pengujian SQL Injection, pengujian XSS, dan penampilan hasil pengujian, sebagaimana disajikan pada Tabel 2.

Tabel 2. Black Box Testing

No	Fungsi yang Diuji	Hasil yang Diharapkan	Hasil
1	Input URL Website – memasukkan URL yang valid	Sistem menerima URL dan melanjutkan proses pengujian	Sukses
2	Validasi URL – memasukkan URL yang tidak valid	Sistem menampilkan pesan kesalahan	Sukses
3	Scan Website – menekan tombol Mulai Pengujian	Sistem menjalankan proses scanning website	Sukses
4	SQL Injection Test – memilih pengujian SQL Injection	Sistem melakukan pengujian SQL Injection	Sukses
5	XSS Test – memilih pengujian XSS	Sistem melakukan pengujian XSS	Sukses
6	Hasil Pengujian – menampilkan hasil pengujian	Sistem menampilkan hasil analisis keamanan website	Sukses

Hasil pengujian pada Tabel 2 menunjukkan bahwa seluruh enam fungsi utama aplikasi dinyatakan sukses, yang berarti sistem mampu menerima masukan dari pengguna, melakukan validasi URL website, menjalankan proses scanning, melakukan pengujian SQL Injection dan Cross-Site Scripting (XSS), serta menampilkan hasil pengujian dengan baik sesuai rancangan yang telah dibuat. Temuan ini sejalan dengan penelitian-penelitian sejenis pada Tabel 1 yang juga menunjukkan efektivitas pendekatan penetration testing berstandar OWASP dalam mengidentifikasi kerentanan aplikasi web; perbedaannya, penelitian ini menghasilkan aplikasi pengujian mandiri berbasis Python yang tidak hanya berfungsi sebagai alat deteksi, tetapi juga sebagai sarana pembelajaran mengenai mekanisme dasar penetration testing.

Dengan demikian, aplikasi penetration testing yang dikembangkan dapat digunakan sebagai alat bantu untuk melakukan identifikasi awal terhadap potensi kerentanan keamanan website secara lebih cepat dan terstruktur dibandingkan dengan proses pengujian yang dilakukan secara manual.

5. KESIMPULAN DAN SARAN

a. Penelitian ini berhasil merancang dan membangun aplikasi penetration testing berbasis website menggunakan bahasa pemrograman Python dan framework Flask yang dapat digunakan untuk membantu proses

pengujian keamanan website mengacu pada standar OWASP.

b. Aplikasi yang dikembangkan mampu melakukan pengujian keamanan website dengan mendeteksi potensi kerentanan SQL Injection dan Cross-Site Scripting (XSS) berdasarkan URL dan parameter yang dimasukkan oleh pengguna, dilengkapi fitur input URL, scanning otomatis, analisis hasil pengujian, serta penyajian informasi kerentanan yang ditemukan.

c. Hasil pengujian menggunakan metode Black Box Testing terhadap enam fungsi utama menunjukkan seluruh fungsi dapat berjalan sesuai dengan kebutuhan fungsional yang telah ditentukan, mulai dari input URL, validasi URL, scanning website, pengujian SQL Injection dan XSS, hingga penampilan hasil pengujian.

d. Pengembangan lebih lanjut disarankan mencakup penambahan jenis pengujian keamanan lain seperti Cross-Site Request Forgery (CSRF), Local File Inclusion (LFI), dan Remote File Inclusion (RFI); penambahan fitur penyimpanan riwayat hasil pengujian; pengembangan fitur laporan dalam format PDF; peningkatan kemampuan analisis untuk memberikan rekomendasi perbaikan; serta pengembangan antarmuka yang lebih interaktif bagi pengguna dengan pengetahuan terbatas mengenai keamanan website.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Dayan Singasatia, S.Kom., M.Kom. selaku Pembimbing Utama dan Bapak Syariful Alam, M.Kom. selaku Pembimbing Pendamping atas bimbingan dan arahan selama penelitian ini, serta kepada Bapak Teguh Iman Hermanto, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Wastukencana Purwakarta atas dukungan yang diberikan hingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] N. Qolbi and H. Kabetta, "Platform Edukasi Keamanan Siber Berbasis Web Berdasarkan Kerangka Global Literasi Digital UNESCO," *Jurnal INTEKNA*, vol. 24, no. 2, 2024.
- [2] M. Tarigan, "Security Analysis of Payroll System Using the Penetration Testing Execution Standard (PTES) and OWASP Top

- 10 2021,” *Jurnal Pilar Nusa Mandiri*, vol. 22, no. 1, pp. 96–101, Mar. 2026, doi: 10.33480/pilar.v22i1.8267.
- [3] OWASP, “A03:2021 – Injection,” OWASP Top 10, 2021. [Online]. Available: https://owasp.org/Top10/2021/A03_2021-Injection/
- [4] B. Kurniawan, I. Ruslianto, and S. Bahri, “Implementasi Penetration Testing pada Website Menggunakan Metode Penetration Testing Execution Standard (PTES),” *Journal of Computing Engineering, System and Science*, vol. 8, no. 2, pp. 518–528, 2023.
- [5] U.-S. Potti, H.-S. Huang, H.-T. Chen, and H.-M. Sun, “Security Testing Framework for Web Applications: Benchmarking ZAP V2.12.0 and V2.13.0 by OWASP as an Example,” 2024.
- [6] R. Beniwal, A. Sharma, S. Jain, and S. Vyas, “A Review on Full Stack Web Development,” 2023.
- [7] D. Rohmaniah, W. Miftahul Ashari, and A. Dwi Putra, “Enhancing Website Security Using Vulnerability Assessment and Penetration Testing (VAPT) Based on OWASP Top Ten,” 2025.
- [8] N. Daniasti Darno and E. Lesmana Tjong, “Analisis Celah Keamanan pada Website PDDIKTI Menggunakan Metode Penetration Testing dan Framework ISSAF,” *Jurnal Sains dan Teknologi*, vol. 12, no. 2, 2025.
- [9] B. Ariwibowo, “Analisis Keamanan Website Berbasis Black-Box Fuzzing: Studi Kasus Kerentanan XSS dan SQL Injection dalam Website X,” 2024.
- [10] K. Sanghavi and K. Samdani, “Overview of SQL Injection Types, Defense Tools and Prevention Techniques,” 2024.
- [11] M. C. Wijaya, “Security Analysis of SQL Injection Attacks on Multimedia and Journal-Services Sites Using Concatenated Input Validation and Parsing Method (CIVP),” *Ingenierie des Systemes d’Information*, vol. 29, no. 5, pp. 1915–1924, Oct. 2024, doi: 10.18280/isi.290523.
- [12] A. A. Yunanto, M. Ghazi, and A. Ghifari, “Analisis Efektivitas Parameterized Queries dalam Pencegahan Serangan SQL Injection,” *Jurnal Informatika Polinema (JIP)*.
- [13] J. Kim and J. Park, “Enhancing Security of Web-Based IoT Services via XSS Vulnerability Detection,” *Sensors*, vol. 23, no. 23, Dec. 2023, doi: 10.3390/s23239407.
- [14] N. Hikmah Maulida, “Studi Literatur Penerapan Metode Prototype dan Waterfall dalam Pembuatan Sebuah Aplikasi atau Website.” pp. 1032–1040, Jan. 2025, doi: 10.47709/digitech.v4i2.5153.
- [16] B. Alturas, “Connection Between UML Use Case Diagrams and UML Class Diagrams: A Matrix Proposal,” 2023.
- [17] F. Siewe and G. M. Ngounou, “On the Execution and Runtime Verification of UML Activity Diagrams,” *Software*, vol. 4, no. 1, p. 4, Feb. 2025, doi: 10.3390/software4010004.
- [18] Siska Narulita, Ahmad Nugroho, and M. Zakki Abdullah, “Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS),” *Bridge: Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 3, pp. 244–256, Aug. 2024, doi: 10.62951/bridge.v2i3.174.
- [19] S. W. Ramdany, S. Aulia Kaidar, B. Aguchino, C. Amelia, A. Putri, and R. Anggie, “Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web.”
- [20] H. Shah, “Advancing Web Development - Enhancing Component-Based Software Engineering and Design Systems Through HTML5 Customized Built-in Elements,” *International Journal of Web & Semantic Technology*, vol. 15, no. 1, pp. 17–31, Jan. 2024, doi: 10.5121/ijwest.2024.15102.
- [21] A. S. Wulandari, A. Saepudin, M. P. Kinanti, Z. Sudesi, A. Saifudin, and Y. Yulianti, “Pengujian Aplikasi Sistem Informasi Akademik Berbasis Web Menggunakan Metode Black Box Testing Equivalence Partitioning,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 5, no. 2, p. 102, May 2022, doi: 10.32493/jtsi.v5i2.17561.
- [22] T. Ariyadi, A. Putri Salsabila, and Y. P. Nugroho, “Implementasi Secure Code pada Pengembangan Sistem Keamanan Website Teknik Komputer Universitas Bina Darma Menggunakan Penetration Testing dan OWASP ZAP.”
- [23] N. Hidayat and M. A. Nugroho, “Analisis Celah Keamanan pada Website SMA Negeri 3 Berau dengan Metode Penetration Testing,” 2025.
- [24] A. Abdullah and M. Kopravi, “Analisis Keamanan Website pada Instansi XYZ Melalui Penetration Testing Menggunakan Framework ISSAF & OWASP,” *Digital Transformation Technology*, vol. 5, no. 1, pp. 547–555, Aug. 2025, doi: 10.47709/digitech.v5i1.6881.
- [25] R. Marta Dinata, M. Alzril, M. I. Yamin, H. Effendi, and M. Febriansyah, “Analisis Keamanan Situs Web Rumah Sakit Menggunakan Metode Penetration Testing OWASP,” *Sainstech: Jurnal Penelitian dan Pengkajian Sains dan Teknologi*, vol. 35, no. 2, pp. 99–100, Jun. 2025, doi: 10.37277/stch.v35i2.2383.
- [26] M. Arifai Nurizki and E. Iman Heri Ujianto, “Analisis Keamanan Website Desa Budaya DIY dengan Metode Penetration Testing (Pentest) dan OWASP ZAP,” 2024.