Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

RANCANG BANGUN VULNERABLE WEB SIMULATION PADA REPLIKA PORTAL AKADEMIK UNTUK PENGUJIAN PENETRASI XSS DAN SQL INJECTION

Fadli Maghfirli[1], Najwa Wahyu Azzuhra[2], Muhlis Tahir[3]

[123,]Program Studi Pendidikan Informatika, Universitas Trunodjoyo Madura

Jl. Raya Telang, Telang, Kec. Kamal, Kabupaten Bangkalan, Jawa Timur 69162
[1] fadlimaghfirli5@gmail.com, [2] najwaazzuhra27@gmail.com, [3] muhlis.tahir@trunojoyo.ac.id

Abstract

Web-based application security is frequently overlooked, although input parameter validation negligence can lead to fatal consequences such as administrative access takeover and database breaches. This risk is notably high in academic platforms managing sensitive data assets. Addressing this issue, this study aims to develop a standalone *Vulnerable Web Simulation* as a university academic portal replica with *Role-Based Access Control* (RBAC) architecture, serving as a transparent and legal vulnerability evaluation environment. Testing was conducted experimentally using manual and automated *penetration testing* methods by integrating ParamSpider, DalFox, and SQLMap. Empirical results demonstrated that the absence of *Prepared Statements* triggered a critical *SQL Injection* vulnerability. Through manual testing using the authentication *bypass* payload admin' -- -, the administrator *login* gate was successfully breached. Furthermore, automated scanning using SQLMap successfully uncovered the internal structure and extracted all sensitive credentials from the users table. On the frontend interface, the absence of *output encoding* functions triggered *Cross-Site Scripting* (XSS) vulnerabilities. A JavaScript payload was permanently stored in the discussion forum (*Stored XSS*), and DalFox detected a *Reflected XSS* vulnerability on the search parameter. In conclusion, ignoring *secure coding* principles empirically collapses the entire integrity of academic access control. These findings serve as a practical guide for patching vulnerabilities and a reference for establishing virtual cybersecurity laboratories in educational environments.

Keywords : *vulnerability assessment, penetration testing, cross-site scripting, SQL injection*

Abstrak

Keamanan aplikasi berbasis web sering kali diabaikan, padahal kelalaian pada validasi parameter masukan dapat berakibat fatal seperti pengambilalihan hak akses dan kebocoran basis data. Risiko ini sangat tinggi pada platform akademik institusi pendidikan yang mengelola aset data sensitif. Berangkat dari masalah tersebut, penelitian ini bertujuan merancang bangun *Vulnerable Web Simulation* mandiri berbentuk replika portal akademik universitas berarsitektur *Role-Based Access Control* (RBAC) sebagai lingkungan evaluasi kerentanan yang transparan dan legal. Pengujian dilakukan secara eksperimental melalui metode *penetration testing* manual dan otomatis dengan mengintegrasikan ParamSpider, DalFox, dan SQLMap. Hasil pengujian empiris membuktikan bahwa ketiadaan *Prepared Statements* memicu celah *SQL Injection* kritis. Melalui pengujian manual dengan muatan *bypass* otentikasi admin' -- -, gerbang *login* administrator berhasil ditembus. Lebih lanjut, pemindaian otomatis menggunakan SQLMap berhasil membongkar struktur internal dan mengekstrak seluruh data kredensial sensitif dari tabel pengguna. Pada lapisan antarmuka, absennya fungsi *output encoding* memicu kerentanan *Cross-Site Scripting* (XSS). Penyisipan skrip JavaScript tersimpan permanen pada fitur forum diskusi (*Stored XSS*), dan DalFox mendeteksi satu titik rentan *Reflected XSS* pada parameter pencarian. Kesimpulannya, pengabaian prinsip

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

secure coding secara empiris meruntuhkan seluruh integritas kontrol akses akademik. Hasil penelitian ini berimplikasi sebagai panduan praktis penambalan celah keamanan dan rujukan pembangunan laboratorium virtual keamanan siber di lingkungan pendidikan.

Kata kunci : *vulnerability assessment, penetration testing, cross-site scripting, SQL injection*

1. PENDAHULUAN

Perkembangan teknologi sistem informasi pada era transformasi digital saat ini telah mengalami kemajuan yang sangat pesat guna memfasilitasi kebutuhan masyarakat dalam mengakses berbagai informasi secara instan melalui platform *website*. Namun, dalam pengelolaan teknologi tersebut, aspek keamanan (*web security*) sering kali ditempatkan sebagai prioritas terakhir oleh pengelola teknologi informasi dibandingkan dengan performa sistem [1]. Padahal, platform web menyimpan berbagai aset data sensitif seperti informasi pribadi dan data transaksi penting yang wajib dilindungi secara ketat dari akses pihak yang tidak bertanggung jawab. Ketidaksiapan infrastruktur digital dalam menghadapi anomali lalu lintas siber (*cyber threats*) yang bersifat dinamis dapat berakibat pada kerugian materi dan non-materi yang sangat besar bagi sebuah organisasi. Oleh karena itu, identifikasi celah keamanan secara proaktif melalui penilaian kerentanan (*vulnerability assessment*) menjadi langkah yang mendesak untuk menemukan kelemahan desain sebelum dieksploitasi oleh penyerang [2].

Kondisi tersebut diperparah oleh fakta bahwa sistem basis data sering menjadi target utama serangan siber untuk mencuri dokumen rahasia melalui teknik *SQL Injection* (SQLi) [3]. Serangan ini diakui sebagai risiko paling kritis karena penyerang memodifikasi perintah SQL pada memori *web client* guna memanipulasi logika kueri pada basis data *backend*. Karakteristik serangan ini tergolong sangat berbahaya karena tidak adanya struktur pola serangan yang standar, sehingga metode deteksi tradisional berbasis tanda tangan (*signature-based*) menghadapi tantangan besar untuk mengidentifikasinya secara efektif [4]. Selain ancaman pada basis data, terdapat pula risiko *Cross-Site Scripting* (XSS) di mana penyerang menyisipkan skrip berbahaya ke halaman web sisi klien agar dieksekusi oleh peramban pengguna lain [5]. Serangan-serangan ini umumnya berhasil melampaui kontrol akses dan memberikan dampak risiko tinggi, terutama pada parameter input seperti kolom komentar atau form pencarian yang kurang dalam pengawasan keamanan. Munculnya berbagai serangan siber ini sering kali berakar dari kelalaian dalam menyaring karakter-karakter khusus pada form input, sehingga diperlukan metode pengujian yang sistematis seperti *penetration testing* untuk mengatasinya [6].

Dalam alur pengujian yang efektif, penggunaan alat bantu otomatisasi menjadi faktor kunci untuk meningkatkan efisiensi pendeteksian celah keamanan. Sebelum eksploitasi dilakukan, tahap pengumpulan informasi (*reconnaissance*) menjadi bagian yang sangat menentukan keberhasilan pengujian [7]. Penggunaan alat bantu berbasis Python seperti ParamSpider menjadi sangat relevan dalam tahap ini untuk menggali parameter tersembunyi dari arsip web tanpa harus membangun interaksi langsung dengan *host* target [8]. Setelah parameter pintu masuk (*entry point*) ditemukan, pengujian kerentanan XSS diperkuat dengan penggunaan alat otomatisasi seperti DalFox yang memanfaatkan *fuzzy logic* untuk menyuntikkan berbagai kombinasi *payload* [8]. Di sisi lain, alat bantu otomatis seperti SQLMap telah terbukti efektif dalam mendeteksi dan mengeksploitasi celah keamanan *database* di lingkungan server berbasis sistem operasi terbuka secara otomatis [2]. Sinergi antar perangkat lunak penguji ini memungkinkan identifikasi celah keamanan pada berbagai lapisan aplikasi web secara lebih sistematis [7].

Berbagai penelitian terdahulu telah memberikan kontribusi penting dalam upaya pengamanan aplikasi web melalui beragam skenario pengujian. Teknik *brute force* digunakan untuk mendeteksi dini kerentanan XSS menggunakan puluhan jenis *payload* PortSwigger secara otomatis dengan durasi rata-rata pemindaian yang sangat efisien [9]. Selain itu, terdapat tinjauan mengenai bahaya serangan hibrida yang mengombinasikan kelemahan teknis *Stored XSS* dengan manipulasi psikologis

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

Pretexting untuk merekam *session cookie* pengguna secara otomatis [10]. Simulasi serangan gabungan antara XSS dan SQLi untuk memetakan risiko keamanan pada URL dan halaman *login* menggunakan arsitektur pertahanan *Content Security Policy* [11]. Bahkan, penggunaan teknologi masa depan seperti kecerdasan buatan (*machine learning*) berbasis model generatif dan teknik *hybrid* algoritma *Naive Bayes*, *Random Forest*, hingga *Support Vector Machine* kini mulai didalami untuk mengidentifikasi pola serangan baru dan menekan tingkat kesalahan deteksi [12]. Panduan komprehensif yang mengintegrasikan taktik mitigasi *Phishing*, XSS, dan SQLi melalui implementasi *parameterized queries* dan *Content Security Policy* menjadi referensi esensial bagi para pengembang untuk menegakkan arsitektur pengkodean yang aman [13].

Penelitian-penelitian sebelumnya berfokus pada pengujian *black-box* di sistem produksi publik atau evaluasi pertahanan seperti WAF. Namun, masih terdapat kesenjangan berupa minimnya studi yang memanfaatkan lingkungan simulasi lokal tanpa filter keamanan untuk membedah kelemahan arsitektur berbasis *Role-Based Access Control* (RBAC). Oleh karena itu, kontribusi ilmiah utama penelitian ini adalah rancang bangun *Vulnerable Web Simulation* mandiri berbentuk replika portal akademik. Pendekatan ini memberikan transparansi penuh terhadap eksekusi muatan berbahaya, sehingga batas keruntuhan sistem akibat serangan XSS dan SQLi dapat dianalisis secara mendalam, legal, dan aman. Penelitian ini bertujuan untuk merancang halaman web akademik yang sengaja dikonfigurasi memiliki celah keamanan tanpa sanitasi, menguji mekanisme *bypass* otentikasi melalui *payload* `admin' -- -`, serta mengevaluasi efektivitas alat bantu otomatisasi ParamSpider, DalFox, dan SQLMap dalam memetakan permukaan serangan secara terukur.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Berbagai penelitian terdahulu telah dilakukan untuk mengevaluasi dan mengamankan aplikasi berbasis web dari ancaman serangan siber. Penelitian yang dilakukan oleh Nelmiawati dan Dealova (2025) membahas tentang meningkatnya ancaman serangan *Cross-Site Scripting* (XSS) dan *SQL Injection* pada platform web, serta tantangan dalam menghadapi taktik *polyglot obfuscation*. Metode penyelesaian yang digunakan adalah metode eksperimental dengan menguji berbagai kombinasi *payload* terhadap *Web Application Firewall* (WAF) ModSecurity. Hasil pengujian menunjukkan bahwa seluruh *payload* berhasil dideteksi dan diblokir dengan respons HTTP 403, membuktikan bahwa aturan bawaan CRS 4.7 secara efektif melindungi sistem dari ancaman tersebut [14].

Penelitian selanjutnya oleh Yunanri dkk. (2021) mengangkat masalah kerentanan keamanan yang membahayakan aplikasi *Open Journal System* (OJS) yang beroperasi secara terbuka di internet. Menggunakan pendekatan pengujian penetrasi (*penetration testing*) *Black-Box*, hasil pengujian tersebut berhasil mengidentifikasi total 98 celah keamanan pada OJS, yang terdiri dari 1 kerentanan berisiko tinggi, 7 kerentanan menengah, dan 90 kerentanan berisiko rendah [15].

Pada tahun 2023, Syahab melakukan penelitian mengenai pentingnya mengaudit keamanan informasi pada *website* publik (Stasiun Klimatologi D.I. Yogyakarta) yang rentan terhadap eksploitasi SSL/TLS, khususnya *DROWN Attack*. Dengan memanfaatkan *tools* Network Mapper (Nmap) dan Qualys SSL, penelitian ini menghasilkan pemetaan evaluasi ketahanan website serta memberikan status metrik keamanan sebagai landasan mitigasi pelindungan data layanan instansi [16].

Penelitian lain oleh Bagaskara dkk. (2025) menyoroti tingginya potensi celah keamanan pada *website* instansi pemerintah yang rentan dimanfaatkan oleh pihak tidak bertanggung jawab. Metode penelitian yang digunakan adalah *penetration testing* dengan merujuk pada standar kerentanan global *Open Web Application Security Project* (OWASP) Top 10 guna mengidentifikasi kelemahan spesifik dan meningkatkan postur keamanan *website* [17].

Selain itu, penelitian oleh Muhammad dkk. (2023) membahas kerentanan aplikasi web sebagai media pertukaran data yang sering menjadi sasaran utama peretasan *SQL Injection* dan XSS. Hasil penelitian ini membuktikan bahwa implementasi perlindungan *Web Application Firewall* (WAF) secara efektif mampu mendeteksi serta mencegah berbagai jenis intrusi agar tidak dapat menembus basis data aplikasi [18].

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

2.2. Matriks Penelitian Terdahulu

Tabel 1. Matriks Penelitian Terdahulu

No	Judul Penelitian	Hasil Utama
1	*Analysis of Polyglot Obfuscation Techniques against ModSecurity in Preventing Cross-Site Scripting (XSS) and SQL Injection Attacks with Experimental Method*	*Payload* serangan XSS dan SQLi berhasil dideteksi dan diblokir dengan respons HTTP 403 oleh WAF ModSecurity.
2	Deteksi Serangan *Vulnerability* Pada *Open Jurnal System* Menggunakan Metode *Black-Box*	Mengidentifikasi 98 celah keamanan pada OJS tanpa merusak aset produksi.
3	Analisis Audit Kemanan Informasi *Website* Dari *Drown Attack* Menggunakan Network Mapper Dan Qualys SSL	Memetakan evaluasi ketahanan *website* instansi dalam menghadapi potensi eksploitasi DROWN.
4	Pengujian *Website* Dinas Sosial Surabaya Menggunakan Metode *Penetration Testing* Dan OWASP Top 10	Identifikasi dan Analisis kerentanan berdasarkan parameter global OWASP Top 10.
5	Pengamanan Aplikasi Web Dari Serangan *SQL Injection* Dan *Cross Site Scripting* Menggunakan *Web Application Firewall*	Penerapan WAF efektif mendeteksi dan mencegah intrusi *SQL Injection* dan XSS ke dalam basis data.

2.3. *GAP Analysis* dan *State of the Art*

Berdasarkan telaah terhadap berbagai penelitian terdahulu, dapat disimpulkan bahwa sebagian besar studi keamanan web masih menitikberatkan pada pengujian penetrasi (*black-box*) terhadap sistem produksi yang sudah berjalan di ranah publik (seperti OJS dan *website* instansi pemerintah), maupun evaluasi mekanisme pertahanan aktif seperti WAF dan enkripsi SSL. Sementara itu, penelitian yang secara khusus mengkaji pembedahan kode sumber melalui lingkungan simulasi lokal (*vulnerable lab*) tanpa filter keamanan seperti *sanitization* dan *encoding* masih sangat terbatas. Keterbatasan pada penelitian terdahulu sering kali berbenturan dengan kendala legalitas dan etika saat melakukan pengujian pada aset publik yang sedang beroperasi, sehingga batas maksimal eksploitasi dan keruntuhan sistem tidak dapat dieksplorasi secara utuh. Oleh karena itu, kebaruan (*state of the art*) dari penelitian ini adalah pengembangan *Vulnerable Web Simulation* mandiri berbentuk replika sistem akademik berarsitektur *Role-Based Access Control* (RBAC). Pendekatan mandiri ini menutupi kesenjangan penelitian sebelumnya dengan memberikan transparansi observasi 100% terhadap seluruh alur eksekusi *payload* berbahaya, sehingga evaluasi kerentanan dapat dianalisis hingga batas runtuhnya pertahanan secara legal, empiris, dan aman.

3. METODOLOGI PENELITIAN

3.1. Batasan Penelitian

Agar pengujian terfokus dan terukur, simulasi kerentanan pada *Vulnerable Web Application* ini dibatasi pada parameter berikut:

1) Lingkungan Pengujian: Eksploitasi sepenuhnya dilakukan pada lingkungan simulasi lokal (*localhost*) menggunakan *Windows Subsystem for Linux* (WSL) distribusi Ubuntu. Sistem berjalan di atas server Apache dan basis data MySQL tanpa implementasi protokol HTTPS (SSL/TLS).
2) Cakupan Kerentanan: Pengujian difokuskan secara eksklusif pada kerentanan validasi input di lapisan aplikasi (Layer 7 OSI), yaitu *SQL Injection* (SQLi) dan *Cross-Site Scripting* (XSS).
3) Absennya Mekanisme Pertahanan: Sistem sengaja dikonfigurasi tanpa menggunakan *Web Application Firewall* (WAF) maupun *Intrusion Detection System* (IDS) guna memfasilitasi observasi eksploitasi muatan (*payload*) yang murni dan tanpa intervensi.

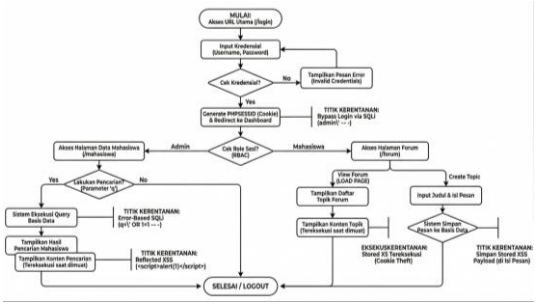
Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

3.2. Pengembangan *Vulnerable Web Application*

Proses pembangunan laboratorium simulasi (*vulnerable lab*) diawali dengan pengumpulan bahan dan dependensi perangkat lunak yang merepresentasikan arsitektur portal akademik [9]. Sistem dikembangkan sebagai aplikasi berbasis web menggunakan bahasa pemrograman PHP dan basis data MySQL sebagai *backend* penyimpanan data kredensial. Implementasi kode pemrograman pada sistem simulasi ini sengaja dirancang memiliki kelemahan struktural pada aspek otentikasi dan fungsionalitas input.

Cara perancangan simulasi kerentanan dilakukan dengan menerapkan teknik penggabungan *string* langsung (*string concatenation*) tanpa melibatkan fungsi penyaringan (*sanitization*) pada *query* SQL *backend*, serta mengabaikan fungsi *output encoding* seperti `htmlspecialchars()` pada antarmuka *web client* [19]. Sebagai basis eksekusi, simulasi ini diletakkan pada lingkungan lokal menggunakan *Windows Subsystem for Linux* (WSL) distribusi Ubuntu guna menguji ketahanan kontrol akses *Role-Based Access Control* (RBAC) secara transparan dan legal.

Visualisasi alur logika sistem dan titik kerentanan (*vulnerability points*) dipetakan untuk menunjukkan mekanisme akses pada setiap fitur portal akademik. Dimulai dari celah *bypass* pada halaman *login*, alur berlanjut ke pengecekan peran pengguna (RBAC) untuk membedakan akses administrator dan mahasiswa. Parameter pencarian diidentifikasi memiliki celah *SQL Injection*, sementara fitur forum diskusi rentan terhadap serangan *Stored XSS* akibat lemahnya validasi input. Detail alur kerja dan pemetaan titik eksploitasi sistem secara utuh dirangkum dalam Gambar 1 berikut:



Gambar 1. *Flowchart* Sistem

3.3. Skenario Pengujian Serangan

Skenario pengujian dirancang untuk mensimulasikan taktik penyerang dalam mengeksploitasi kelemahan sistem dari luar [20]. Langkah awal dilakukan dengan melakukan serangan *Authentication Bypass* pada formulir *login* utama. Struktur logika kueri SQL aslinya dimanipulasi secara manual menggunakan injeksi *payload* logika tautologi khusus. Adapun alur eksekusi *bypass login* administrator dijabarkan melalui algoritma perintah kueri berikut :

1) Sebelum Perbaikan: Kueri Rentan Terhadap Manipulasi String

```
$query = "SELECT * FROM users WHERE
username = '$username' AND password
'$password'";
```

2) Penyusupan *Input Payload* Teknis pada *Form Login*

```
$username = "admin' -- -";
$password = "bebas";
```

3) Hasil Manipulasi Kueri *Backend* Basis Data (Akses *Bypass* Terbuka)

```
SELECT * FROM users WHERE username =
'admin' -- -' AND password = 'bebas'
```

Keberhasilan naskah perintah di atas memaksa server mengabaikan validasi kata sandi karena karakter `-- -` bertindak sebagai komentar pada kueri sql di kode program [3]. Setelah penyerang mendapatkan token sesi aktif administrator (PHPSESSID), langkah pengujian dilanjutkan ke tahap *reconnaissance* menggunakan *tool* ParamSpider untuk memetakan parameter URL tersembunyi yang berpotensi menjadi titik masuk (*entry point*) serangan [8]. Berdasarkan hasil pelacakan parameter, skenario pengujian serangan dibagi menjadi dua kategori teknis:

a. Skenario Serangan *Cross-Site Scripting* (XSS)

Skenario *Stored XSS* diuji secara manual pada fitur forum diskusi mahasiswa dengan menyisipkan struktur skrip JavaScript. Tujuannya adalah memvalidasi apakah skrip berbahaya tersebut tersimpan permanen dan tereksekusi otomatis pada peramban pengguna lain untuk memicu kebocoran data sesi. Skrip yang dimasukkan pada input pesan diskusi forum sebagai berikut:

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

```
<script>alert(document.cookie)</script>
```

Sementara itu, Pengujian celah manipulasi skrip sisi klien ini menyasar dua vektor berbeda, yaitu *Reflected XSS* dan *Stored XSS*. Skenario deteksi *Reflected XSS* diotomatiskan menggunakan tool DalFox untuk menganalisis respon parameter pencarian dengan menginjeksikan penanda FUZZ [8]. Perintah eksekusi terminal yang dikonfigurasi adalah:

```
dalfox url "siakad-vuln.test" -C
"PHPSESSID=[ID]"
```

b. Skenario Serangan *SQL Injection* (SQLi)

Pemetaan kueri basis data dijalankan secara otomatis menggunakan *tool* SQLMap pada parameter pencarian halaman direktori mahasiswa. Skenario ini dijalankan melalui tiga sub-fase instruksi terminal secara berurutan:

1) Fase 1 (*Database Enumeration*): Menarik daftar nama basis data aktif menggunakan perintah :

```
sqlmap -u "siakad-vuln.test" --
cookie="PHPSESSID=[ID]" --dbs -batch
```

2) Fase 2 (*Table Enumeration*): Membongkar struktur tabel internal pada *database* siakad-vuln melalui perintah :

```
sqlmap -u "siakad-vuln.test" --
cookie="PHPSESSID=[ID]" -D siakad-
vuln -tables
```

3) Fase 3 (*Data Dumping*): Mengekstrak seluruh data kredensial akun dari tabel target menggunakan perintah :

```
sqlmap -u "siakad-vuln.test" --
cookie="PHPSESSID=[ID]" -D siakad-
vuln -T users --dump
```

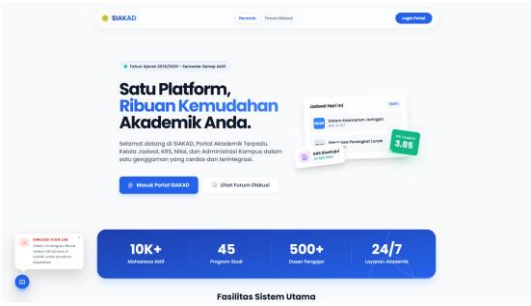
4. HASIL DAN PEMBAHASAN

Pengujian kerentanan (*vulnerability assessment*) difokuskan pada analisis dampak manipulasi parameter masukan terhadap integritas data pada sisi *frontend* maupun *backend* aplikasi akademik. Evaluasi keamanan siber ini dijalankan di bawah pengawasan lingkungan operasional yang terisolasi menggunakan serangkaian skenario eksploitasi terukur untuk membuktikan kelemahan penulisan kode program. Seluruh data teknis dikumpulkan secara berkala melalui pemindaian otomatis menggunakan perangkat lunak pihak ketiga dengan indikator efektivitas berupa kecepatan ekstraksi dokumen basis data serta keberhasilan eksekusi skrip ilegal [5].

4.1. Hasil Pengembangan *Vulnerable Web Application*

Laboratorium virtual yang dikembangkan merupakan sebuah replika Sistem Informasi Akademik (SIAKAD) yang berjalan pada klaster server Apache dengan basis data MySQL. Pembangunan antarmuka menggunakan *framework* Tailwind CSS untuk menghasilkan tampilan portal yang modern dan responsif. Fokus utama dari pengembangan sistem ini adalah menyediakan lingkungan simulasi yang secara transparan memperlihatkan titik lemah pada validasi input di setiap fiturnya.

Antarmuka beranda atau *landing page* berfungsi sebagai pusat navigasi utama yang merepresentasikan wajah publik dari sebuah portal universitas. Halaman ini memuat berbagai informasi akademik umum dan menjadi titik awal bagi pengguna untuk mengakses fitur-fitur fungsional lainnya.



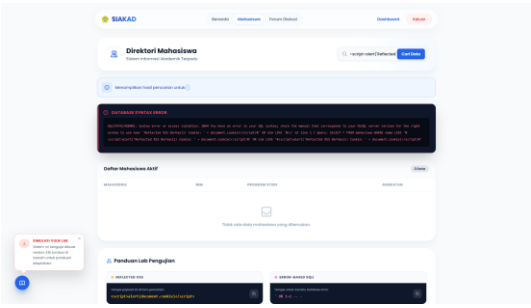
Gambar 2. Tampilan Antarmuka *Landing Page*

Halaman *login* dirancang sebagai gerbang otentikasi untuk memvalidasi hak akses pengguna berdasarkan tingkat kewenangannya. Secara teknis, formulir pada halaman ini memiliki kerentanan kritis karena mengolah *username* dan kata sandi tanpa melalui mekanisme pembersihan input pada sisi server. Ketiadaan penerapan *prepared statements* pada kueri basis data *backend* menjadikan halaman ini sebagai target utama simulasi serangan *authentication bypass*.

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

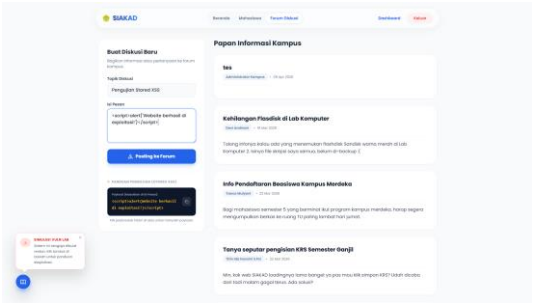
Gambar 3. Tampilan Antarmuka Halaman *Login* (Rentan *SQL Injection*)

Fitur pencarian mahasiswa diimplementasikan untuk memudahkan pengguna dalam menemukan data direktori akademik melalui kueri parameter URL. Fungsionalitas pencarian ini sengaja dibiarkan terbuka tanpa adanya filter terhadap karakter khusus yang dimasukkan oleh pengguna ke dalam kolom input. Tampilan ini memfasilitasi simulasi serangan *SQL Injection* dan *Reflected XSS* guna mengevaluasi ketahanan parameter masukan pada aplikasi.



Gambar 4. Tampilan Antarmuka Halaman Pencarian Mahasiswa (Rentan SQLi dan *Reflected XSS*)

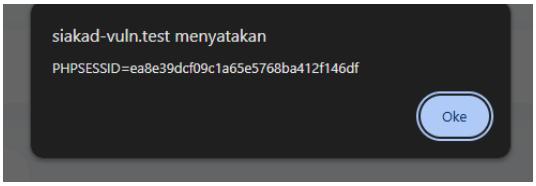
Halaman forum diskusi disediakan sebagai media interaksi sosial di mana pengguna dapat mengirimkan pesan diskusi ke dalam sistem. Kerentanan utama pada fitur ini terletak pada absennya fungsi *output encoding* yang seharusnya menyaring entitas HTML berbahaya sebelum ditampilkan kembali ke layar pengguna. Akibatnya, setiap skrip ilegal yang diinputkan akan tersimpan secara permanen di basis data dan tereksekusi secara otomatis pada setiap sesi pengguna lain yang memuat halaman ini.



Gambar 5. Tampilan Antarmuka Halaman Forum Diskusi (Rentan *Stored XSS*)

4.2. Hasil Pengujian Serangan *Cross Site Scripting* (XSS)

Evaluasi kerentanan *Cross-Site Scripting* (XSS) difokuskan pada pengujian kemampuan aplikasi dalam menangani muatan skrip berbahaya yang disisipkan melalui parameter input pengguna. Secara manual, pengujian jenis *Stored XSS* diaplikasikan pada fitur forum diskusi dengan menyisipkan struktur skrip JavaScript berupa `<script>alert(document.cookie)</script>`. Hasilnya, skrip tersebut diterima tanpa melalui tahapan validasi, tersimpan secara permanen di dalam basis data server, dan tereksekusi secara otomatis oleh peramban klien setiap kali halaman forum dimuat ulang. Fenomena ini menunjukkan adanya risiko tinggi terjadinya pencurian data sesi (*session hijacking*) terhadap pengguna lain yang mengakses forum tersebut, sebagaimana ditunjukkan pada tampilan visual Gambar 6:

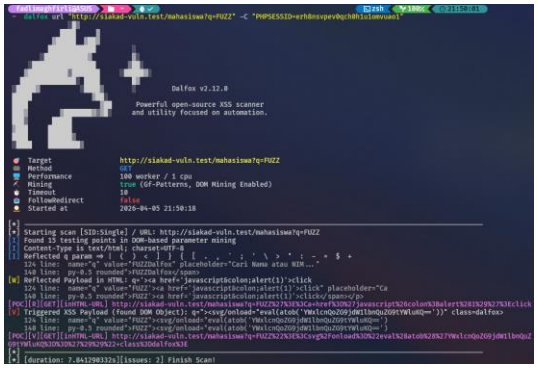


Gambar 6. Eksekusi Skrip JavaScript (*Pop-up Alert*) Pada Browser

Selain metode manual, pengujian jenis *Reflected XSS* dioptimalkan menggunakan perangkat otomatisasi DalFox untuk memetakan kelemahan pada parameter URL pencarian mahasiswa. Pengujian ini memanfaatkan fitur *fuzzy logic* DalFox dengan menargetkan parameter lokal yang rentan melalui instruksi terminal `dalfox url "http://siakad-vuln.test/mahasiswa?q=FUZZ" -C "PHPSESSID=`

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

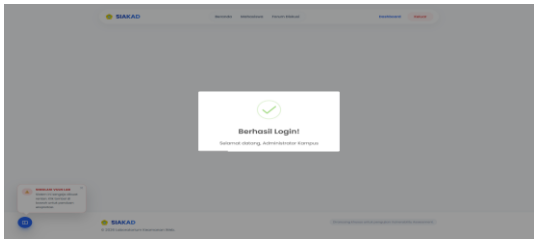
`erh8nsvpev0qch0h1u1omvuao1"`. Penyertaan parameter kustomisasi *Cookie* dengan ID sesi aktif tersebut memungkinkan DalFox untuk melakukan pemindaian komprehensif pada lapisan direktori yang memerlukan otentikasi. Hasil pemindaian otomatis yang mengonfirmasi adanya kerentanan eksekusi skrip ilegal pada sisi klien (*client-side*) terdokumentasi secara lengkap dalam laporan log pada Gambar 7 berikut:



Gambar 7. Pemindaian Kerentanan XSS menggunakan Tools DalFox

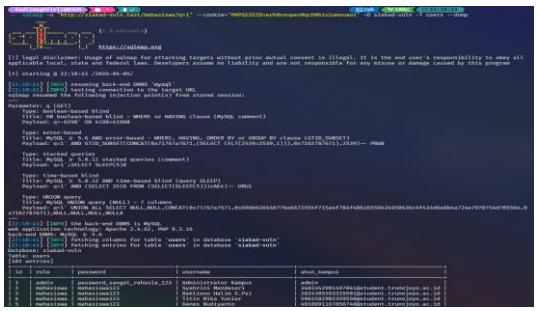
4.3. Hasil Pengujian Serangan *SQL Injection* (SQLi)

Pengujian celah keamanan *SQL Injection* dilakukan dengan memadukan teknik manipulasi kueri secara manual dan pemindaian otomatis menggunakan perangkat lunak pihak ketiga. Pada pengujian tahap pertama, dilakukan serangan *Authentication Bypass* secara manual pada formulir *login* utama administrator. Penguji menyisipkan *payload* logika tautologi khusus, yaitu `admin' -- -`, langsung ke dalam kolom nama pengguna. Injeksi ini bertujuan untuk memaksa sistem memberikan hak akses penuh tanpa memerlukan input kata sandi yang sah dari pengguna aslinya. Keberhasilan pengujian manual ini secara empiris membuktikan bahwa logika otentikasi pada *backend* dapat dimanipulasi sepenuhnya hanya dengan memanfaatkan karakter komentar sistem. Representasi visual dari keberhasilan akses ilegal melalui formulir *login* tersebut disajikan pada Gambar 8 berikut:



Gambar 8. Pengujian Manual *Bypass Login Administrator*

Tahap pengujian kedua dilakukan secara otomatis menggunakan perangkat lunak SQLMap untuk mengeksploitasi kerentanan pada parameter pencarian mahasiswa. Pengujian dieksekusi melalui antarmuka *Command Line Interface* (CLI) dengan menyertakan *cookie* sesi aktif yang didapat dari tahap *bypass* sebelumnya. Perintah spesifik yang dijalankan adalah `sqlmap -u "http://siakad-vuln.test/mahasiswa?q=1" --cookie="PHPSESSID=erh8nsvpev0qch0h1u1omvuao1" -D siakad-vuln -T users --dump`. Melalui eksekusi parameter penguraian (`--dump`) tersebut, SQLMap berhasil melakukan penetrasi ke dalam sistem dan mengekstrak seluruh data sensitif dari tabel *users* dalam format teks biasa. Rekaman aktivitas terminal yang menunjukkan keberhasilan ekstraksi data rahasia dari basis data *backend* dipaparkan pada **Error! Reference source not found.** di bawah ini:



Gambar 9. Pengujian Ekstraksi Database menggunakan Tools SQLMap

4.4. Analisis Celah Keamanan

Keberhasilan seluruh eksploitasi dalam pengujian ini bersumber dari ketiadaan implementasi pengkodean yang aman (*secure coding*) pada pengelolaan parameter masukan. Pada kasus serangan *SQL Injection*, keberhasilan *bypass* otentikasi menggunakan `admin' -- -` dan

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

ekstraksi basis data oleh SQLMap pada parameter `?q=1` diakibatkan oleh penggunaan penggabungan *string* dinamis. Ketiadaan *Prepared Statements* menyebabkan peladen langsung mengeksekusi karakter khusus dari luar sebagai bagian dari struktur instruksi logika MySQL, sehingga mengabaikan validasi otentikasi lanjutan di *backend*.

Sementara itu, kerentanan *Cross-Site Scripting* (XSS) terjadi akibat absennya fungsi sanitasi dan *output encoding* pada sisi *frontend*. Eksekusi skrip `<script>alert(document.cookie)</script>` yang tersimpan di forum diskusi, serta temuan DalFox pada parameter `?q=FUZZ`, membuktikan bahwa sistem merender data mentah dari pengguna secara langsung ke *Document Object Model* (DOM). Pengabaian fungsi pembersihan seperti `htmlspecialchars()` membuat peramban gagal membedakan antara teks biasa dengan instruksi HTML/JavaScript, sehingga mengeksekusi skrip berbahaya tersebut secara otomatis.

Secara komprehensif, akumulasi celah pada *frontend* dan *backend* ini mengakibatkan runtuhnya integritas aplikasi. Sistem *Role-Based Access Control* (RBAC) yang diterapkan menjadi tidak efektif, karena peretas dapat mengambil alih basis data dan melakukan pencurian sesi (*session hijacking*) langsung melalui kelemahan validasi input pada fitur-fitur dasar akademik.

4.5. Implikasi Keamanan dan Dampak Praktis

Eksploitasi celah SQLi dan XSS pada simulasi portal akademik ini memiliki dampak fatal jika terjadi pada sistem produksi nyata di institusi pendidikan:

1) Keruntuhan Kontrol Akses (*Confidentiality* & *Integrity*): Keberhasilan *Authentication Bypass* memungkinkan penyerang mengambil alih hak akses Administrator. Dampak praktisnya meliputi risiko manipulasi ilegal pada data nilai mahasiswa dan Kartu Rencana Studi (KRS).
2) Kebocoran Data Massal (*Confidentiality*): Ekstraksi tabel *users* oleh SQLMap memicu risiko kebocoran data privasi massal (NIM, nama, dan kata sandi). Hal ini membuka celah pengambilalihan akun dan merugikan reputasi institusi.
3) Pembajakan Sesi Pengguna (*Integrity*): Celah *Stored* XSS pada forum diskusi memungkinkan pencurian token sesi (*cookie*) pengguna lain secara pasif. Penyerang dapat bertindak atas nama korban tanpa perlu mengetahui kata sandi mereka.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil membuktikan dan memetakan kerentanan fatal pada aplikasi berbasis web yang mengabaikan prinsip *secure coding*. Melalui pengembangan lingkungan *Vulnerable Web Simulation* pada portal akademik, skenario serangan telah berhasil dieksekusi baik secara manual maupun otomatis menggunakan *tools* seperti ParamSpider, DalFox, dan SQLMap. Hasil pengujian menunjukkan kesesuaian dengan identifikasi masalah awal, di mana absennya implementasi *Prepared Statements* memicu celah *SQL Injection* yang memungkinkan *bypass* otentikasi dan ekstraksi basis data secara penuh. Lebih lanjut, ketiadaan sanitasi input dan fungsi *output encoding* mengakibatkan kerentanan *Reflected* dan *Stored XSS* yang memfasilitasi pencurian data sesi (*session hijacking*). Secara keseluruhan, akumulasi celah keamanan ini meruntuhkan integritas sistem *Role-Based Access Control* (RBAC) aplikasi.

Sebagai prospek pengembangan untuk studi selanjutnya, hasil penelitian ini dapat dijadikan landasan untuk menguji efektivitas solusi pertahanan yang lebih spesifik dan aplikatif. Beberapa rekomendasi teknis untuk penelitian mendatang antara lain:

1) Implementasi *Active Defense*: Mengintegrasikan *Web Application Firewall* (WAF) seperti ModSecurity secara berdampingan dengan *Intrusion Detection System* (IDS) seperti Suricata ke dalam lingkungan simulasi server.
2) Komparasi Arsitektur Modern: Membandingkan tingkat kerentanan pada arsitektur *web native* ini dengan sistem yang dibangun menggunakan *framework* modern seperti Laravel. Evaluasi lanjutan dapat difokuskan pada pengujian ketahanan lapisan *Object-Relational Mapping* (ORM) dan *templating engine* bawaan *framework* dalam memblokir intervensi taktis (XSS dan SQLi) secara default.

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 9, Nomor 2, Juni 2026.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i2
http://e-journal.stmiklombok.ac.id/index.php/misi

6. UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Dosen Pengampu Mata Kuliah Sistem Keamanan Jaringan pada Program Studi Pendidikan Informatika Universitas Trunodjoyo Madura yang telah memberikan bimbingan, arahan teknis, serta fondasi keilmuan teoritis maupun praktis selama masa perkuliahan, sehingga implementasi dan evaluasi celah keamanan dalam penelitian ini dapat dirancang dengan terstruktur.

DAFTAR PUSTAKA:

[1] A. Miftahun, N. Rizky, M. Tahir, T. A. Sapitri, and M. F. Islam, "Implementasi SQL Injection dalam Penetration Testing Untuk Deteksi Celah Keamanan Web," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 4, pp. 6964–6968, 2025, doi: https://doi.org/10.36040/jati.v9i4.14350.

[2] S. Praptomo, S. Suryanto, and J. Kurniawan, "Database Vulnerability Testing Memanfaatkan SQLMap Pada Ubuntu 22.04 (Studi Kasus: antaratech.net)," *Jurnal Informatika Medis (J-INFORMED)*, vol. 1, no. 2, pp. 68–72, Nov. 2023, doi: 10.52060/im.v1i2.1629.

[3] V. Abdullayev and A. S. Chauhan, "SQL Injection Attack: Quick View," *Mesopotamian Journal of CyberSecurity*, vol. 2023, pp. 30–34, 2023, doi: 10.58496/MJCS/2023/006.

[4] A. Gustiyono, E. I. Alwi, and S. M. Abdullah, "Analisa Kerentanan Website Terhadap Serangan Cross-Site Scripting (XSS) Metode Penetration Testing," *Cyber Security dan Forensik Digital*, vol. 7, no. 1, pp. 25–33, 2024, doi: https://doi.org/10.14421/csecurity.2024.7.1.4432.

[5] F. A. Maylani, M. Tahir, N. N. Juniar, D. Sari, W. A. Zulaica, and I. Ismael, "Analisis Keamanan Website E-library Kampus dengan Metode PTES (Penetration Testing Execution Standard)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 4, p. 5643, 2025, doi: https://doi.org/10.36040/jati.v9i4.13897.

[6] W. Tjahjadi and M. Hardjianto, "Deteksi Celah Keamanan SQL Injection Pada Website dengan Menggunakan Metode Penetration Testing," *Telematika MKOM*, vol. 17, no. 2, pp. 97–107, 2024, doi: https://doi.org/10.36080/telematikamkom.3889.

[7] R. S. Wiandani, M. Tahir, I. A. Dyransyha, and R. Ummah, "Identifikasi Serangan SQL Injection Berbantuan Aplikasi Pengujian Keamanan Web DVWA(Damn Vulnerable Web Application)," *Digital Transformation Technology*, vol. 5, no. 1, pp. 375–382, Jul. 2025, doi: 10.47709/digitech.v5i1.5922.

[8] M. A. Mu'min, Z. Alamin, F. Fathir, and S. Ramadhan, "Uji Penetrasi Injeksi SQL terhadap Celah Keamanan Website XYZ menggunakan Tools SQLMap," *Scientific : Journal of Computer Science and Informatics*, vol. 1, no. 2, pp. 47–52, Jul. 2024, doi: 10.34304/scientific.v1i2.330.

[9] Y. Sitorus and K. Kasmawi, "Analisis Keamanan Website XYZ Terhadap Serangan SQL Injection dan Cross Site Scripting (XSS)," *Jurnal RESTIKOM : Riset Teknik Informatika Dan Komputer*, vol. 7, no. 2, pp. 217–225, 2025, doi: https://doi.org/10.52005/restikom.v7i2.462.

[10] A. A. Chandra, A. T. Zy, and A. Nugroho, "Penerapan Teknik Penetration Testing Terhadap Cross Site Scripting (XSS) dalam Pengembangan Website," *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 9, no. 2, pp. 262–270, Jul. 2024, doi: 10.36341/rabit.v9i2.4822.

[11] R. F. G. Ananda and E. Sulistiyani, "Analisis Keamanan Terhadap Kombinasi XSS dan Social Engineering Pada Website," *Jurnal Informatika Polinema*, vol. 12, no. 2, pp. 367–374, 2026, [Online]. Available: https://portswigger.net/burp/documentation.

[12] A. Zandra and M. Hardjianto, "Deteksi Celah Keamanan XSS Pada Website dengan Metode Brute Force," *SENAFTI*, vol. 3, no. 2, pp. 66–73, 2024.

[13] M. Maulana *et al.*, "Introduction to Web Security: Detecting XSS with Dalfox and Paramspider," *Jurnal Pendidikan Tambusai*, vol. 8, no. 2, pp. 34259–34270, 2024, [Online]. Available: https://forms.gle/qH1vpvGGUvVd7Zhs5

[14] N. Nelmiawati and K. Dealova, "Analysis of Polyglot Obfuscation Techniques against ModSecurity in Preventing Cross-Site Scripting (XSS) and SQL Injection Attacks with Experimental Method," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 4, pp. 2540–2549, Sep. 2025, doi: 10.52436/1.jutif.2025.6.4.5000.

[15] Y. W, D. T. Yuwono, R. Rodianto, and Y. Yuliadi, "Deteksi Serangan Vulnerability Pada Open Jurnal System Menggunakan Metode Black-Box," *JIRE (Jurnal Informatika & Rekayasa Elektronika)*, vol. 4, no. 1, pp. 68–77, 2021, [Online]. Available: http://e-journal.stmiklombok.ac.id/index.php/jire ISSN.2620-6900

[16] A. S. Syahab, "Analisis Audit Kemanan Informasi Website Dari Drown Attack Menggunakan Network Mapper Dan Qualys SSL," *Jurnal Manajemen Informatika & Sistem Informasi (MISI)*, vol. 6, no. 1, pp. 39–47, 2023, doi: 10.36595/misi.v5i2.

[17] B. A. Bagaskara, M. Idhom, and H. E. Wahanani, "Pengujian Website Dinas Sosial Surabaya Menggunakan Metode Penetration Testing Dan OWASP Top 10," *JIRE (Jurnal Informatika & Rekayasa Elektronika)*, vol. 8, no. 1, pp. 40–50, Apr. 2025, [Online]. Available: http://e-journal.stmiklombok.ac.id/index.php/jire ISSN.2620-6900

[18] H. Haikal Muhammad *et al.*, "Pengamanan Aplikasi Web Dari Serangan SQL Injection Dan Cross Site Scripting Menggunakan Web Application Firewall," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 5, pp. 3265–3273, 2023.

[19] M. A. M. Oudah and M. F. Mahrusin, "SQL Injection Detection using Machine Learning: A Review," *Malaysian Journal of Science Health & Technology*, vol. 10, no. 1, pp. 39–49, Apr. 2024, doi: 10.33102/mjosht.v10i1.368.

[20] S. S. Nair, "Securing Against Advanced Cyber Threats: A Comprehensive Guide to Phishing, XSS, and SQL Injection Defense," *Journal of Computer Science and Technology Studies*, vol. 6, no. 1, pp. 76–93, 2024, doi: 10.32996/jcsts.2024.6.1.9.