 ISSN:2302-8149
e-ISSN:2540-9719

Analisis dan Perbaikan Keamanan Sistem Informasi *Web* Agribisnis berbasis *Grey-Box* dan *White-Box*

Analysis and Improvement of an Agribusiness Web Information System Security using Grey-Box and White-Box Testing

[1]M. Isma Halil*, [2]Mansur
[1,2]Program Studi Keamanan Sistem Informasi, Jurusan Teknik Informatika, Politeknik Negeri Bengkalis
[1,2]Jl. Bathin Alam, Sungai Alam, Bengkalis, Riau, Indonesia
*e-mail: halelmahalil@gmail.com

(***received*:** 13 December 2025, ***revised*:** 4 January 2026, ***accepted*:** 6 January 2026)

Abstrak

Penelitian ini bertujuan untuk menganalisis dan memperbaiki keamanan Sistem Informasi Pencatatan Panen Sawit *SawitGoDigi* dengan menggunakan pendekatan grey-box dan white-box testing. Sistem ini digunakan oleh petani, agen, supir, dan admin untuk mengelola data lahan, hasil panen, distribusi, dan transaksi, sehingga memiliki risiko keamanan tinggi apabila terdapat celah kerentanan. Pengujian dilakukan mengacu pada OWASP Web Security Testing Guide (WSTG) v4.2 dan metode penilaian risiko OWASP Risk Rating Methodology. Tahap pengujian meliputi reconnaissance, pemindaian otomatis menggunakan OWASP ZAP, eksploitasi manual, evaluasi risiko, penerapan perbaikan, dan retesting. Hasil penelitian menunjukkan adanya beberapa kerentanan signifikan, termasuk SQL Injection pada fitur pencarian, kelemahan manajemen sesi melalui cookie *trusted_device*, serta tidak adanya mekanisme rate limiting yang memungkinkan brute force pada proses login. Evaluasi risiko menunjukkan bahwa *SQL Injection* dan *session hijacking* memiliki tingkat risiko *High*, sementara *brute force* berada pada kategori *Medium*. Perbaikan dilakukan melalui penerapan prepared statements, penguatan atribut cookie, penambahan security headers, serta implementasi rate limiting. Hasil retesting menunjukkan bahwa seluruh kerentanan berhasil ditutup dan berada pada tingkat risiko *Low*. Penelitian ini membuktikan bahwa penerapan pendekatan pengujian keamanan komprehensif yang mencakup eksploitasi, perbaikan, dan verifikasi ulang mampu meningkatkan keamanan aplikasi web agribisnis secara signifikan. Hasil penelitian menunjukkan bahwa sebelum perbaikan terdapat empat kerentanan dengan tingkat risiko High dan Medium, yang mencakup SQL Injection, Session Hijacking, Brute Force Login, dan Security Misconfiguration. Setelah proses remediation dan retesting, seluruh kerentanan High dan Medium berhasil diturunkan menjadi Low atau dinyatakan Closed, sehingga sistem dinyatakan aman untuk digunakan.

Kata kunci: keamanan aplikasi web, OWASP, penetration testing, grey-box testing, agribisnis digital

Abstract

This study aims to analyze and improve the security of the SawitGoDigi Palm Oil Harvest Recording Information System using grey-box and white-box testing approaches. The system is used by farmers, agents, drivers, and administrators to manage land data, harvest results, distribution, and transaction records, which makes it highly exposed to security risks if vulnerabilities are present. The security testing process was conducted based on the OWASP Web Security Testing Guide (WSTG) v4.2 and the OWASP Risk Rating Methodology. The testing stages included reconnaissance, automated scanning using OWASP ZAP, manual exploitation, risk evaluation, implementation of security improvements, and retesting. The results revealed several significant vulnerabilities, including SQL Injection in the search feature, weak session management through the trusted_device cookie, and the absence of a rate-limiting mechanism that enabled brute-force attacks during the login process. The risk assessment indicated that SQL Injection and session hijacking were classified as High risk, while brute-force attacks were categorized as Medium risk. Security improvements were implemented through the use of prepared statements, strengthening cookie attributes, adding security headers, and implementing rate limiting. Retesting results confirmed that all identified vulnerabilities were

 ISSN:2302-8149
e-ISSN:2540-9719

successfully mitigated and reduced to a Low-risk level. This study demonstrates that a comprehensive security testing approach, which includes exploitation, remediation, and verification through retesting, can significantly enhance the security of agribusiness web applications. Furthermore, the findings show that before remediation, the system contained four vulnerabilities with High and Medium risk levels, namely SQL Injection, Session Hijacking, Brute-Force Login, and Security Misconfiguration. After the remediation and retesting process, all High- and Medium-risk vulnerabilities were successfully reduced to Low risk or marked as Closed, indicating that the system is secure for operational use.

Keywords: *web application security, OWASP, penetration testing, grey-box testing, agribusiness system.*

1 Pendahuluan

Digitalisasi pada sektor agribisnis terus berkembang seiring meningkatnya kebutuhan pencatatan dan pengelolaan hasil panen secara cepat, akurat, dan terintegrasi. Pada industri kelapa sawit, sistem informasi berbasis web telah menjadi komponen penting dalam mendukung proses pencatatan hasil panen, pengelolaan data petani, pengaturan distribusi melalui supir, serta transaksi antara petani dan agen. Penelitian Patricia et al. menunjukkan bahwa pemanfaatan platform digital berbasis web dan kecerdasan buatan mampu meningkatkan kolaborasi, efektivitas komunikasi, dan kualitas informasi di kalangan petani sawit [1]. Sistem serupa juga diterapkan pada perusahaan perkebunan, seperti pada penelitian mengenai sistem pelaporan administrasi harian pekerja sawit di PT Sintang Raya yang terbukti meningkatkan efisiensi dan transparansi proses operasional [2]. Temuan tersebut menunjukkan bahwa sistem informasi berbasis web telah menjadi bagian fundamental dalam modernisasi pengelolaan perkebunan sawit di Indonesia.

Namun, meningkatnya penggunaan sistem web di sektor agribisnis juga memperbesar risiko keamanan. OWASP dalam laporan OWASP Top 10:2021 mengidentifikasi berbagai ancaman kritis seperti *Injection, Broken Authentication, Security Misconfiguration, dan Cross-Site Scripting* (XSS) yang masih menjadi ancaman dominan bagi aplikasi web modern [3]. Di sisi lain, laporan resmi Lanskap Keamanan Siber Indonesia 2023 yang diterbitkan oleh Badan Siber dan Sandi Negara (BSSN) mencatat adanya 403.990.813 trafik ancaman siber sepanjang tahun 2023, dengan puncak serangan mencapai 78.464.385 insiden pada bulan Agustus [4]. Tingginya volume ancaman ini menegaskan bahwa sistem berbasis web yang mengelola data bernilai ekonomi tinggi, termasuk data hasil panen dan transaksi agribisnis, memerlukan pengamanan yang komprehensif dan berbasis standar.

Sistem Informasi Web Agribisnis SawitGoDigi merupakan platform yang digunakan secara aktif oleh petani, agen, supir, dan admin untuk mengelola data panen, lahan, distribusi, serta transaksi. Meskipun sistem ini telah dilengkapi autentikasi *Single Sign-On* (SSO) dan *One-Time Password* (OTP), hingga kini belum terdapat dokumentasi formal yang menerapkan pengujian keamanan berbasis kerangka kerja internasional. Padahal data yang dikelola memiliki nilai ekonomi tinggi dan rentan terhadap akses tidak sah, manipulasi, maupun kebocoran apabila sistem memiliki celah keamanan.

Sementara berbagai penelitian agribisnis terdahulu berfokus pada pengembangan sistem dan peningkatan fungsionalitas, aspek keamanan masih jarang diteliti secara mendalam. Penelitian keamanan aplikasi web lain yang ada pun cenderung terbatas pada proses deteksi kerentanan tanpa menyertakan eksploitasi terkendali, penilaian risiko, perbaikan (*remediation*), dan retesting sebagai verifikasi efektivitas mitigasi [5], [6]. Hal ini menunjukkan adanya kesenjangan penelitian yang penting, yaitu belum diterapkannya pendekatan pengujian keamanan yang komprehensif pada sistem informasi agribisnis yang digunakan secara operasional.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk melakukan analisis dan perbaikan keamanan pada SawitGoDigi menggunakan pendekatan *grey-box* dan *white-box testing*, dengan mengacu pada OWASP Web Security Testing Guide v4.2 dan penilaian risiko berdasarkan OWASP Risk Rating Methodology. Proses mencakup identifikasi kerentanan, eksploitasi terkendali, perhitungan risiko, perbaikan kerentanan, serta retesting untuk memastikan seluruh celah berada pada tingkat risiko rendah (*Low*). Penelitian ini diharapkan memberikan kontribusi signifikan terhadap

 ISSN:2302-8149
e-ISSN:2540-9719

peningkatan keamanan aplikasi web agribisnis dan menjadi referensi metodologis bagi sistem serupa di sektor perkebunan sawit.

2 Tinjauan Literatur

Digitalisasi pada sektor agribisnis kelapa sawit mendorong pemanfaatan sistem informasi berbasis web untuk meningkatkan efisiensi operasional, transparansi data, serta kolaborasi antar pelaku usaha. Berbagai implementasi sistem agribisnis berbasis web telah dikembangkan untuk mendukung pencatatan hasil panen, pelaporan aktivitas pekerja, dan distribusi informasi bagi petani serta perusahaan perkebunan [1], [2]. Meskipun demikian, sebagian besar penelitian tersebut masih berfokus pada aspek fungsional dan peningkatan efisiensi operasional, tanpa membahas aspek keamanan sistem informasi yang mengelola data bernilai ekonomi tinggi seperti hasil panen, transaksi, dan distribusi.

Dari perspektif kualitas layanan, penelitian pada Website E-Makaryo menggunakan metode EUCS menekankan pentingnya evaluasi kepuasan pengguna dalam meningkatkan efektivitas sistem [7]. Namun, penelitian tersebut lebih berfokus pada aspek usability dibandingkan perlindungan keamanan. Hal ini mengindikasikan bahwa studi agribisnis maupun studi evaluasi layanan masih minim dalam membahas elemen keamanan sistem informasi sebagai komponen kritis yang harus dipenuhi dalam penerapan aplikasi digital.

Penelitian terkait keamanan aplikasi web menunjukkan bahwa pendekatan Vulnerability Assessment and Penetration Testing (VAPT) berbasis OWASP Top 10 mampu mengidentifikasi berbagai kerentanan kritis, khususnya yang berkaitan dengan Security Misconfiguration, Injection, dan Authentication Failures [5], [6], [8], [9]. Penggunaan alat pemindaian otomatis seperti OWASP ZAP juga terbukti efektif dalam mendeteksi kerentanan awal pada aplikasi web, termasuk pada sistem layanan publik dan akademik [10], [11]. Namun, sebagian besar penelitian tersebut masih berfokus pada tahap deteksi kerentanan dan belum mengombinasikan eksploitasi manual, analisis risiko terstruktur, serta proses retesting untuk memverifikasi efektivitas perbaikan yang dilakukan.

Dari sisi standar metodologi, NIST SP 800-115 memberikan kerangka teknis pengujian keamanan yang mencakup tahapan perencanaan, pelaksanaan, dan verifikasi perbaikan [12]. OWASP Web Security Testing Guide (WSTG) turut menyediakan panduan rinci dalam menguji berbagai aspek keamanan aplikasi web, termasuk authentication, authorization, input validation, hingga server configuration [13]. Selain itu, OWASP Top 10 2021 [3] serta OWASP Risk Rating Methodology [14] memberikan dasar klasifikasi kerentanan dan pendekatan penilaian risiko berbasis kombinasi kemungkinan eksploitasi dan dampak yang ditimbulkan terhadap sistem.

Pendekatan penilaian risiko (risk scoring) juga digunakan dalam penelitian [15] sebagai kerangka konseptual untuk memahami tingkat keparahan kerentanan dalam pengujian keamanan aplikasi web. Pendekatan ini menekankan bahwa tingkat risiko ditentukan oleh hubungan antara peluang terjadinya eksploitasi dan besarnya dampak terhadap kerahasiaan, integritas, serta ketersediaan sistem. Meskipun demikian, implementasi operasional penilaian risiko dapat berbeda bergantung pada metode pengujian dan konteks sistem yang dianalisis.

Penelitian lain turut menyoroti autentikasi sebagai aspek penting keamanan aplikasi web. Metode Random Forest terbukti meningkatkan efektivitas deteksi anomali dalam OTP [16], sedangkan integrasi OAuth 2.0 dan OTP dapat meningkatkan keamanan SSO tanpa mengurangi kenyamanan pengguna [17]. Namun, kedua studi tersebut masih berfokus pada rancangan mekanisme autentikasi dan tidak menguji implementasinya pada skenario serangan nyata dalam sistem agribisnis.

Kerangka metodologis terkait penggunaan pendekatan *white-box*, *black-box*, dan *grey-box* dapat dipahami melalui penelitian Tohidi et al. [8] yang menunjukkan bahwa perbedaan tingkat pengetahuan internal terhadap suatu sistem akan menghasilkan kedalaman analisis yang berbeda. Meskipun konteks penelitian tersebut bukan pada keamanan aplikasi web, konsep perbedaan akses informasi yang dijelaskan tetap relevan sebagai dasar pemahaman bagaimana tiap pendekatan memberikan sudut pandang yang berbeda dalam proses pengujian. Studi lain mengenai serangan adversarial turut menegaskan bahwa tingkat pengetahuan internal berpengaruh langsung terhadap efektivitas serangan pada skenario *white-box* dan *grey-box*[9]. Dengan demikian, kedua kajian tersebut dapat dijadikan landasan teoritis dalam memahami mekanisme pendekatan pengujian pada aplikasi web.

 ISSN:2302-8149
e-ISSN:2540-9719

Pada tingkat nasional, laporan BSSN bertajuk Lanskap Keamanan Siber Indonesia 2023 mencatat 403.990.813 trafik ancaman sepanjang tahun 2023 dengan lonjakan signifikan pada bulan Agustus [4]. Tren ancaman ini divisualisasikan pada Gambar 1 dan menunjukkan peningkatan intensitas serangan yang menyasar aplikasi web. Kondisi tersebut memperkuat urgensi perlunya pengujian keamanan yang menyeluruh bagi aplikasi agribisnis berbasis web.

Gambar 1 Tren trafik ancaman siber indonesia tahun 2023 (BSSN 2023)

Ringkasan penelitian terdahulu yang relevan disajikan pada Tabel 1. Tabel tersebut memperlihatkan bahwa studi pada sektor agribisnis masih didominasi oleh penelitian yang berfokus pada pengembangan sistem dan peningkatan efisiensi operasional, sementara aspek keamanan belum menjadi perhatian utama. Pada sisi lain, penelitian keamanan aplikasi web cenderung terbatas pada deteksi kerentanan tanpa melibatkan tahapan lanjutan seperti analisis risiko terstruktur ataupun retesting. Demikian pula, studi mengenai autentikasi dan metodologi *grey-box/white-box* belum diterapkan secara langsung pada sistem agribisnis berbasis web.

Tabel 1 Ringkasan penelitian terdahulu terkait sistem informasi agribisnis dan keamanan web

Penelitian	Fokus	Metode	Kelemahan (Gap)
[1], [2]	Sistem agribisnis	Perancangan sistem	Tidak membahas keamanan
[5], [6], [10], [11], [12]	Keamanan web	VAPT, PTES, ZAP	Tidak ada risk scoring & retesting
[13], [14]	Autentikasi	OTP/SSO	Tidak menguji implementasi pada sistem nyata
[15], [16]	Metodologi testing	Grey-box/white-box	Tidak diterapkan pada aplikasi agribisnis

Berdasarkan kajian tersebut, terlihat adanya celah penelitian yang belum terisi. Tidak ditemukan penelitian yang menerapkan pengujian keamanan komprehensif pada sistem agribisnis dengan mengombinasikan pendekatan *grey-box* dan *white-box testing*, pemetaan kerentanan menggunakan OWASP WSTG dan OWASP Top 10, analisis risiko berbasis OWASP Risk Rating, serta validasi perbaikan melalui retesting. Penelitian ini berkontribusi dengan mengisi celah tersebut melalui pendekatan menyeluruh yang tidak hanya mendeteksi kerentanan, tetapi juga memperbaiki, memverifikasi, dan memberikan rekomendasi keamanan berkelanjutan bagi sistem informasi pencatatan panen sawit.

3 Metode Penelitian (or Research Method)

Metode penelitian ini digunakan untuk mengevaluasi keamanan Sistem Informasi Pencatatan Panen Sawit melalui pendekatan *grey-box* dan *white-box testing*. Pendekatan ini dipilih karena mampu menggabungkan perspektif pengguna eksternal dan analisis internal sehingga proses identifikasi kerentanan menjadi lebih komprehensif. Penelitian mengacu pada standar OWASP Web Security Testing Guide (WSTG) v4.2 dan kategori risiko OWASP Top 10 sebagai landasan metodologis utama.

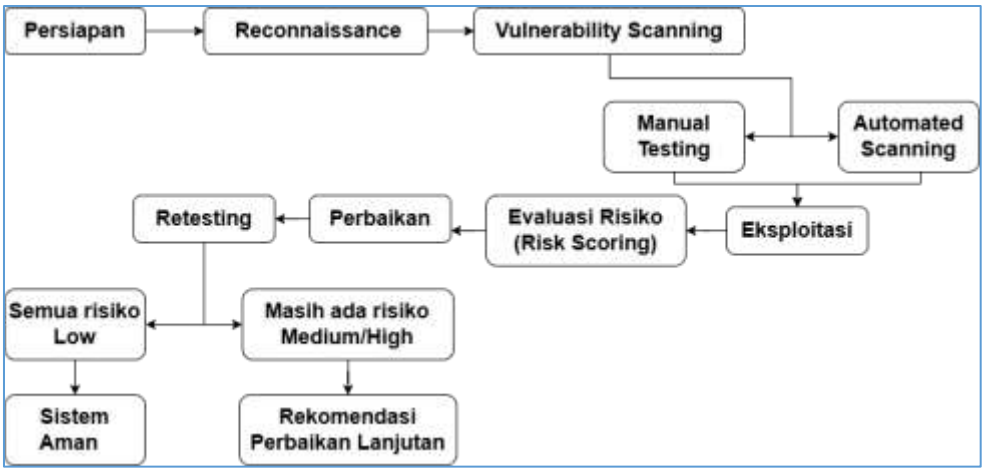
 ISSN:2302-8149
e-ISSN:2540-9719

Data penelitian terdiri atas data primer dan sekunder. Data primer diperoleh melalui proses pengujian langsung yang meliputi keluaran pemindaian otomatis, *log request–response*, struktur endpoint aplikasi, bukti eksploitasi (payload dan tangkapan layar), serta konfigurasi keamanan pada environment hosting tempat sistem berjalan. Data sekunder diperoleh dari dokumentasi aplikasi, catatan pengembang, dan literatur pendukung yang relevan dengan teknik pengujian keamanan aplikasi web.

Alat penelitian yang digunakan meliputi OWASP ZAP sebagai pemindai otomatis berbasis OWASP Top 10; Burp Suite untuk intercepting, manipulasi parameter, dan eksploitasi manual; Nmap untuk enumerasi port dan layanan pendukung; WhatWeb untuk identifikasi teknologi dan fingerprinting server; serta Chrome DevTools untuk inspeksi elemen antarmuka, cookie, API, dan perilaku client-side. Seluruh alat ini digunakan secara terintegrasi untuk meningkatkan akurasi temuan.

Ruang lingkup penelitian difokuskan pada modul aplikasi berbasis web yang digunakan oleh petani, agen, supir, dan admin dalam proses pencatatan panen, pengelolaan transaksi, serta distribusi hasil. Pengujian dilakukan atas izin resmi pemilik sistem dan tidak mencakup infrastruktur fisik, jaringan internal organisasi, maupun sistem eksternal lain di luar aplikasi. Penelitian dilakukan pada environment hosting aktif sehingga hasil pengujian merepresentasikan kondisi operasional sebenarnya.

Rancangan kegiatan penelitian mengikuti alur kerja *penetration testing* yang mencakup tahap persiapan, instalasi alat, penetapan ruang lingkup, pengumpulan informasi, pemindaian otomatis, eksploitasi kerentanan, dokumentasi temuan, evaluasi risiko, proses perbaikan, dan pengujian ulang. Alur metodologi tersebut ditunjukkan pada Gambar 2 sebagai representasi operasional dari tahapan pengujian berbasis OWASP.



Gambar 2 Tahapan penelitian

Penilaian tingkat risiko kerentanan dihitung menggunakan Persamaan (1), yang mengombinasikan parameter likelihood dan impact.

$$\boldsymbol{Risk\ Score = Likelihood \times Impact} \quad \textbf{(1)}$$

Apabila seluruh temuan pada tahap retesting berada pada tingkat risiko Low, maka sistem dinyatakan aman. Namun, apabila masih ditemukan risiko residual dengan kategori Medium atau High, penelitian dilanjutkan pada tahap rekomendasi perbaikan lanjutan sebagaimana ditunjukkan pada diagram alur penelitian.

Metode penelitian ini tidak hanya menghasilkan identifikasi kerentanan, tetapi juga memastikan efektivitas perbaikan dan memberikan rekomendasi penguatan keamanan berkelanjutan pada Sistem Informasi Pencatatan Panen Sawit. Dengan metode ini, seluruh tahapan pengujian dapat dilakukan secara sistematis dan terukur.

4 Hasil dan Pembahasan

Bagian ini menyajikan hasil pengujian keamanan Sistem Informasi Pencatatan Panen Sawit berdasarkan tahapan metodologi *penetration testing* yang telah dijelaskan sebelumnya. Penyajian hasil mengikuti alur penelitian, meliputi *reconnaissance*, pemindaian otomatis, eksploitasi kerentanan,

 ISSN:2302-8149
e-ISSN:2540-9719

evaluasi risiko, perbaikan, dan pengujian ulang. Pembahasan mengaitkan hasil temuan dengan literatur dan standar OWASP.

4.1 Hasil Tahap Persiapan dan Penetapan Ruang Lingkup

Pengujian dilakukan pada aplikasi SawitGoDigi yang berjalan di lingkungan shared hosting cPanel, menggunakan Laravel, MySQL, serta autentikasi berbasis login dan OTP. Ruang lingkup meliputi modul petani, agen, supir, dan admin, karena modul-modul ini menangani data sensitif terkait proses pencatatan panen dan transaksi agribisnis.Pengujian menggunakan pendekatan *grey-box* melalui akun pengguna yang disediakan, dan pendekatan white-box melalui akses terbatas ke file manager untuk memastikan analisis lebih mendalam pada konfigurasi aplikasi. Pengujian berlangsung pada 6 November–4 Desember dan dilaksanakan dengan izin resmi pemilik sistem.

4.2 Hasil Tahap Reconnaissance

Tahap *reconnaissance* dilakukan untuk memetakan permukaan serangan aplikasi dengan mengidentifikasi teknologi server, layanan yang terbuka, struktur endpoint, serta cookie yang digunakan dalam proses autentikasi. Informasi yang diperoleh pada tahap ini menjadi landasan dalam menentukan skenario pengujian lebih lanjut pada fase eksploitasi.

4.2.1 Identifikasi Teknologi dan Konfigurasi Server

Identifikasi komponen server dilakukan menggunakan WhatWeb dan Nmap. Hasil WhatWeb menunjukkan bahwa aplikasi berjalan pada teknologi web modern dengan konfigurasi standar pada lingkungan shared hosting. Rangkuman hasil identifikasi ditampilkan pada Tabel 2.

Tabel 2 Rekapitulasi hasil reconnaissance

Penelitian	Fokus	Metode
Web Server	Apache HTTPD	Server standar pada hosting Linux
Bahasa Pemrograman	PHP 8.2	Versi terbaru pada saat pengujian
Framework	Laravel	Teridentifikasi melalui struktur cookie dan respons server
Library Frontend	Bootstrap, Lightbox	Digunakan untuk tampilan antarmuka
Cookies	XSRF-TOKEN, laravel_session, trusted_device	Terdeteksi melalui header Set-Cookie

Selanjutnya, pemindaian port publik dilakukan menggunakan Nmap untuk mengetahui layanan yang terekspos kepada pengguna eksternal. Ringkasan hasil pemindaian ditunjukkan pada Tabel 3.

Tabel 3 Hasil pemindaian port publik menggunakan nmap

Port	Status	Layanan
80/tcp	open	HTTP
443/tcp	open	HTTPS (SSL/HTTP)

Hasil pemindaian memperlihatkan bahwa hanya dua port yang terbuka, yaitu 80/tcp dan 443/tcp. Port sensitif seperti SSH (22) dan MySQL (3306) tidak dapat dipindai atau memang tidak terekspos karena dibatasi oleh penyedia hosting. Kondisi ini menunjukkan bahwa permukaan serangan pada lapisan jaringan relatif minimal sehingga fokus analisis keamanan diarahkan pada lapisan aplikasi web yang menangani logika bisnis dan autentikasi pengguna.

4.2.2 Analisis Struktur Endpoint dan Mekanisme Autentikasi

Struktur endpoint dianalisis melalui inspeksi manual terhadap antarmuka web dan observasi respons server. Pendekatan manual ini digunakan karena mekanisme proteksi Laravel membatasi enumerasi otomatis terhadap *endpoint non-public.* Berdasarkan hasil analisis, aplikasi menggunakan dua endpoint autentikasi utama, yaitu **/login** untuk pengguna petani, agen, dan supir, serta **/admin/login** untuk admin yang memiliki hak akses lebih tinggi.

Sebagian besar endpoint fungsional lainnya hanya dapat diakses setelah proses autentikasi dilakukan. Ketika *endpoint non*-public diakses secara langsung tanpa sesi autentikasi, server merespons dengan kode status 403 (*Forbidden*) atau 405 (*Method Not Allowed*). Hal ini menunjukkan

 ISSN:2302-8149
e-ISSN:2540-9719

bahwa Laravel telah menerapkan proteksi dasar routing terhadap akses langsung yang tidak sesuai metode atau tanpa kredensial valid.

Analisis autentikasi diperkuat dengan pemeriksaan cookie melalui Chrome DevTools. Sistem menggunakan tiga cookie utama, yaitu laravel_session, XSRF-TOKEN, dan *trusted_device.* Ketiga cookie ini memiliki peran penting dalam menjaga sesi, validasi permintaan, dan *mekanisme trusted device.* Atribut keamanannya dirangkum dalam Tabel 4.

Tabel 4 Atribut keamanan cookie aplikasi sawitgodigi

Nama Cookie	HttpOnly	Secure
laravel_session	✔	✔
XSRF-TOKEN	✘	✔
trusted_device	✘	✔

Berdasarkan analisis keseluruhan, dapat disimpulkan bahwa aplikasi telah menerapkan pemisahan endpoint autentikasi sesuai role pengguna dan proteksi dasar terhadap akses langsung pada endpoint sensitif. Namun, dua cookie penting yakni XSRF-TOKEN dan *trusted_device* tidak memiliki atribut HttpOnly. Kekurangan ini meningkatkan risiko pencurian cookie melalui serangan yang melibatkan JavaScript seperti *session hijacking* atau *XSS-assisted cookie theft.* Temuan ini menjadi salah satu dasar utama dalam penyusunan skenario eksploitasi manual pada tahap berikutnya, terutama yang berkaitan dengan pengujian manajemen sesi dan autentikasi.

4.3 Hasil Tahap Vulnerability Scanning

Tahap vulnerability scanning dilakukan untuk mengidentifikasi potensi kerentanan keamanan aplikasi secara sistematis sebagai dasar penentuan prioritas pengujian lanjutan. Proses ini diawali dengan pemindaian otomatis menggunakan alat berbasis OWASP, yang kemudian dilanjutkan dengan pengujian manual terarah.

4.3.1 Automated Scanning (OWASP ZAP)

Pemindaian otomatis menggunakan OWASP ZAP menghasilkan total 30 alert yang terdiri atas kategori Medium, Low, dan Informational, tanpa adanya temuan risiko High. Berdasarkan hasil analisis, delapan temuan berada pada kategori Medium, sebelas berada pada kategori Low, dan sebelas lainnya termasuk dalam kategori Informational. Distribusi tingkat risiko tersebut divisualisasikan pada Gambar 3 sebagai gambaran umum kondisi keamanan aplikasi sebelum dilakukan pengujian manual.

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	5 (16.7%)	3 (10.0%)	0 (0.0%)	8 (26.7%)
	Low	0 (0.0%)	3 (10.0%)	7 (23.3%)	1 (3.3%)	11 (36.7%)
	Informational	0 (0.0%)	3 (10.0%)	4 (13.3%)	4 (13.3%)	11 (36.7%)
	Total	0 (0.0%)	11 (36.7%)	14 (46.7%)	5 (16.7%)	30 (100%)

Gambar 3 Ringkasan tingkat risiko hasil pemindaian OWASP ZAP

Temuan berkategori Medium umumnya berasal dari kelemahan konfigurasi keamanan (security misconfiguration), terutama tidak diterapkannya *Content Security Policy* (CSP) atau kebijakan CSP yang masih lemah. Selain itu, header anti-clickjacking seperti *X-Frame-Options* atau *frame-ancestors* tidak diaktifkan, sehingga membuka peluang terjadinya serangan framing. OWASP ZAP juga menemukan penggunaan *library JavaScript* yang rentan atau versi lama, yang berpotensi dimanfaatkan apabila terdapat celah pada komponen tersebut. Temuan lain yang cukup penting adalah

 ISSN:2302-8149
e-ISSN:2540-9719

cookie tertentu, khususnya XSRF-TOKEN dan *trusted_device*, tidak memiliki atribut HttpOnly, sehingga memungkinkan akses oleh skrip sisi klien dan meningkatkan risiko serangan pencurian sesi.

Meskipun tidak ditemukan kerentanan dengan tingkat risiko High, pola *misconfiguration* yang muncul menunjukkan bahwa aplikasi memiliki beberapa titik lemah yang dapat dimanfaatkan sebagai pintu masuk dalam proses eksploitasi manual. Kekurangan pada konfigurasi CSP, absennya proteksi *clickjacking*, penggunaan library rentan, dan lemahnya atribut cookie merupakan vektor serangan yang relevan untuk diuji lebih lanjut, terutama dalam konteks OWASP Top 10 kategori A05 (*Security Misconfiguration)* dan A07 *(Identification and Authentication Failures*). Oleh karena itu, hasil pemindaian otomatis ini menjadi dasar dalam penyusunan prioritas pengujian manual pada tahap eksploitasi berikutnya.

4.3.2 Manual Testing (Pengujian Manual Terarah)

Pengujian manual dilakukan sebagai kelanjutan dari tahap pemindaian otomatis untuk memverifikasi temuan awal serta mengidentifikasi kerentanan yang tidak dapat dideteksi secara optimal oleh alat otomatis. Pada tahap ini, pengujian difokuskan pada analisis perilaku aplikasi, manipulasi parameter, pengujian autentikasi dan manajemen sesi, serta simulasi serangan nyata sesuai skenario OWASP Top 10. Hasil dari pengujian manual ini direalisasikan dalam bentuk eksploitasi terkontrol terhadap kerentanan yang ditemukan, sebagaimana dijelaskan pada tahap eksploitasi kerentanan pada Bagian 4.4.

4.4 Hasil Eksploitasi Kerentanan

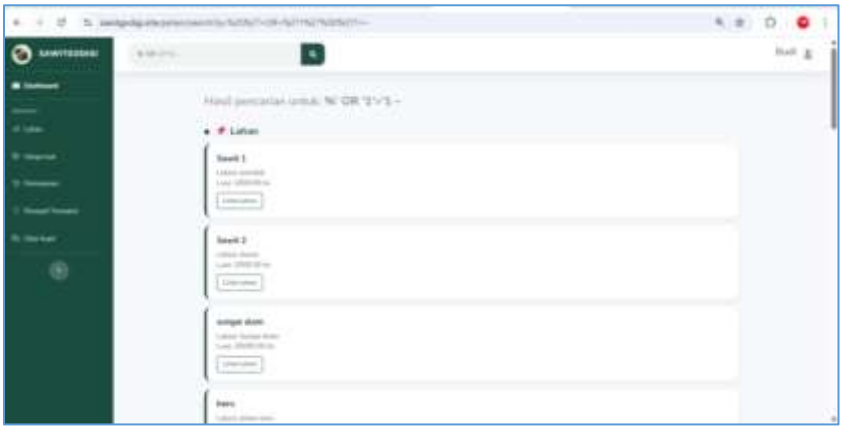
Tahap eksploitasi kerentanan dilakukan untuk memverifikasi secara langsung temuan yang diperoleh pada tahap vulnerability scanning serta menilai dampak nyata dari setiap celah keamanan terhadap sistem. Eksploitasi dilakukan secara terkontrol sesuai skenario OWASP Top 10 guna memastikan bahwa kerentanan yang diidentifikasi benar-benar dapat dimanfaatkan dalam kondisi operasional.

4.4.1 Eksploitasi SQL Injection

Pengujian SQL Injection dilakukan pada endpoint */petani/search?q=* di mana parameter *q* diketahui diproses secara langsung di dalam raw SQL tanpa sanitasi maupun parameter binding. Ketika payload

```
'%' OR '1'='1' –
```

disisipkan, sistem menghasilkan keluaran berupa seluruh data lahan yang tersimpan, meskipun pengguna yang melakukan permintaan tidak memiliki hak akses terhadap keseluruhan data tersebut. Temuan ini membuktikan bahwa mekanisme filtering input belum diterapkan pada modul pencarian, sehingga query dapat dimanipulasi secara bebas oleh attacker. Bukti visual hasil eksploitasi ditunjukkan pada Gambar 4.



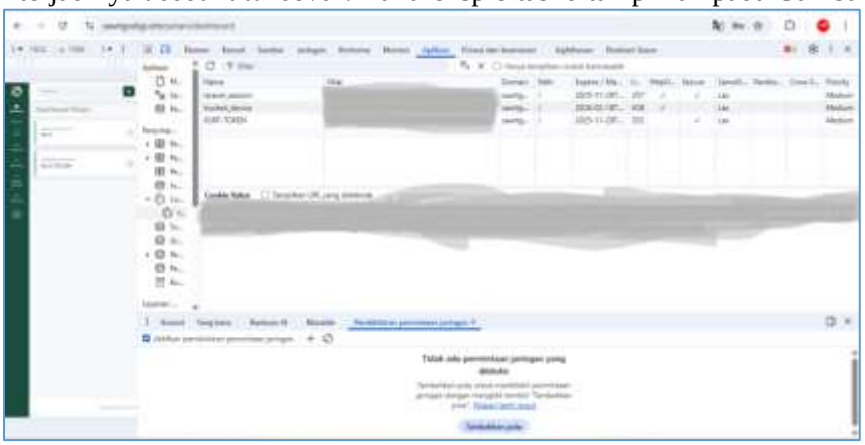
Gambar 4 Output sistem setelah penyisipan payload SQL injection

Kerentanan ini termasuk kategori OWASP A03 – Injection dengan tingkat *risiko High/Critical* karena memungkinkan pengungkapan data sensitif, pemetaan struktur database, serta potensi eskalasi serangan ke tahap credential harvesting dan dump database.

 ISSN:2302-8149
e-ISSN:2540-9719

4.4.2 Eksploitasi Session Hijacking melalui Cookie trusted_device

Hasil inspeksi menggunakan Chrome DevTools menunjukkan bahwa cookie *trusted_device* tidak dilengkapi atribut HttpOnly, sehingga nilainya dapat diakses melalui JavaScript dan berpotensi dicuri melalui serangan XSS atau teknik pencurian sesi lainnya. Eksploitasi dilakukan dengan menyalin nilai cookie dari sesi pengguna sah, kemudian menanamkannya ke browser lain dan mengakses ulang aplikasi. Sistem langsung mengarahkan attacker ke halaman dashboard tanpa proses OTP maupun autentikasi ulang. Kondisi ini menunjukkan bahwa server tidak melakukan validasi menyeluruh terhadap integritas cookie dan hanya bergantung pada kehadiran nilai tersebut, sehingga memungkinkan terjadinya *account takeover*. Bukti eksploitasi ditampilkan pada Gambar 5.



Gambar 5 Bukti kelemahan cookie trusted_device (disamarkan)

Kerentanan ini termasuk kategori OWASP A07 – *Identification* and *Authentication Failures* dengan tingkat risiko *High*, karena memberikan akses penuh terhadap data panen, transaksi, serta fungsi lain sesuai role pengguna yang diserang.

4.4.3 Pengujian Brute Force pada Mekanisme Login (Hydra & Burp Suite Intruder)

Pengujian brute force dilakukan melalui dua pendekatan, yaitu Hydra sebagai brute force otomatis dan Burp Suite Intruder sebagai brute force manual terarah. Seluruh percobaan dilakukan menggunakan kombinasi username valid dari ketiga role (petani, agen, dan supir) serta sekitar 20 password umum dan password salah untuk mengidentifikasi pola respons server.

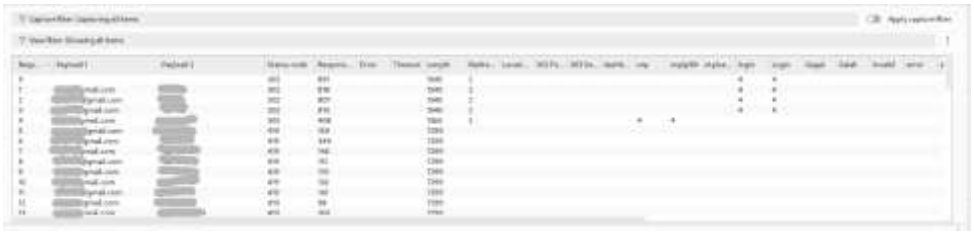
Hasil pengujian menggunakan Hydra menunjukkan bahwa sistem tidak memiliki mekanisme rate limiting maupun *account lockout*. Seluruh permintaan login diterima tanpa pembatasan, sehingga sistem berpotensi dieksploitasi melalui serangan password spraying apabila pengguna memilih password lemah. Meskipun tidak ditemukan password yang valid selama pengujian, absennya kontrol pertahanan dasar menunjukkan bahwa mekanisme autentikasi belum memenuhi standar OWASP ASVS.

Pengujian menggunakan Burp Suite Intruder memberikan gambaran lebih rinci mengenai perilaku respons autentikasi. Pada skenario login tanpa *trusted device*, permintaan yang dikirim untuk role petani dan agen sebagian besar menghasilkan HTTP 419 karena Burp Intruder mengirim permintaan POST berulang tanpa pembaruan token CSRF. Kondisi ini menyebabkan respons server tidak dapat digunakan untuk membedakan password benar atau salah sehingga teknik enumerasi password tidak dapat dilakukan. Berbeda dengan kedua role tersebut, role supir menampilkan pola anomali: baik password benar maupun salah menghasilkan HTTP 302 yang mengarah ke halaman OTP. Hal ini menunjukkan adanya praktik premature OTP issuance, yakni OTP dikirim berdasarkan email valid sebelum dilakukan verifikasi password, sehingga desain autentikasi pada role ini dinilai lemah.

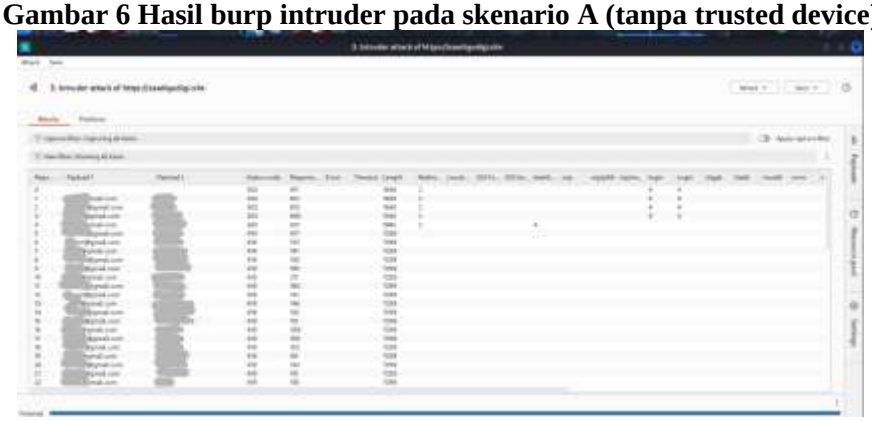
Pada skenario kedua, yaitu setelah perangkat ditetapkan sebagai *trusted device*, pola respons berubah secara signifikan. Password benar tetap menghasilkan HTTP 302 yang mengarah ke proses OTP, sedangkan password salah menghasilkan HTTP 419 yang menunjukkan request invalid atau sesi tidak sinkron. Perbedaan pola ini memungkinkan attacker melakukan enumerasi password dengan cukup mudah. Selain itu, tidak adanya mekanisme rate limiting menjadikan serangan brute force tetap

 ISSN:2302-8149
e-ISSN:2540-9719

dapat dilakukan tanpa batas. Bukti hasil uji Burp Intruder pada kedua skenario ditampilkan pada Gambar 6 dan Gambar 7.



Gambar 6 Hasil burp intruder pada skenario A (tanpa trusted device)



Gambar 7 Hasil burp intruder pada skenario B (trusted device)

Hasil keseluruhan menunjukkan bahwa aplikasi memiliki kelemahan penting pada mekanisme autentikasi, yaitu absennya pembatasan percobaan login, desain OTP yang tidak konsisten, serta perbedaan respons yang dapat dimanfaatkan untuk password enumeration. Kombinasi faktor tersebut menempatkan kerentanan brute force dalam kategori OWASP A07 – Identification and Authentication Failures dengan tingkat risiko Medium–High**.**

4.5 Evaluasi Risiko (Risk Scoring)

Penilaian tingkat risiko dilakukan menggunakan OWASP Risk Rating Methodology, di mana setiap kerentanan dievaluasi berdasarkan dua parameter utama, yaitu Likelihood (kemungkinan eksploitasi) dan Impact (dampak terhadap kerahasiaan, integritas, dan ketersediaan sistem). Nilai risiko dihitung berdasarkan Persamaan (1) yang telah dijelaskan pada bagian metodologi penelitian.

Tabel 5 Evaluasi risiko berdasarkan OWASP risk rating methodology

Kerentanan	Likelihood	Impact	Risk Score	Level
SQL Injection	3	3	9	High/Critical
Session Hijacking	3	3	9	High
Brute Force	2	2	4	Medium
Security Misconfiguration	2	1	2	Low

Hasil evaluasi risiko untuk setiap kerentanan yang ditemukan disajikan secara ringkas pada Tabel 5**,** yang memperlihatkan nilai *Likelihood*, *Impact*, *Risk Score*, serta tingkat risiko dari masing-masing kerentanan. Berdasarkan Tabel 5, kerentanan *SQL Injection* dan *Session Hijacking* memiliki nilai *Risk Score* tertinggi, masing-masing sebesar 9, yang dikategorikan sebagai risiko High hingga *Critical***.**

Kerentanan *SQL Injection* dan *Session Hijacking* tersebut berpotensi menyebabkan pengambilalihan akun, kebocoran data sensitif, manipulasi data transaksi, serta kompromi penuh terhadap sistem. Oleh karena itu, kedua kerentanan ini ditetapkan sebagai prioritas utama dalam proses perbaikan (*remediation*). Sementara itu, kerentanan *Brute Force* berada pada kategori risiko *Medium*, dan *Security Misconfiguration* termasuk dalam kategori risiko *Low*, sebagaimana ditunjukkan pada Tabel 5**.**

Secara keseluruhan, kombinasi kerentanan pada kategori *High* dan *Medium* menunjukkan bahwa sistem memiliki risiko keamanan yang signifikan pada lapisan aplikasi, meskipun permukaan serangan pada lapisan jaringan relatif minimal. Tingkat risiko yang ditunjukkan pada Tabel 5 menjadi

 ISSN:2302-8149
e-ISSN:2540-9719

dasar dalam penyusunan strategi mitigasi serta penentuan prioritas penanganan pada tahap perbaikan berikutnya.

4.6 Hasil Perbaikan (Remediation)

Tahap remediation dilakukan setelah eksploitasi berhasil memverifikasi adanya beberapa kerentanan pada aplikasi. Perbaikan dilakukan menggunakan pendekatan *white-box* dengan meninjau langsung kode sumber, konfigurasi server, serta mekanisme autentikasi. Hasil perbaikan dijelaskan sebagai berikut.

1) Perbaikan SQL Injection

Kerentanan SQL Injection disebabkan oleh penggunaan raw SQL yang menyisipkan variabel pengguna secara langsung tanpa validasi. Contoh kode sebelum perbaikan adalah:

```
$lahan = DB::select("
    SELECT * FROM sw_lahans
    WHERE id_petani = $petaniId
    AND (nama LIKE '%$q%' OR lokasi LIKE '%$q%')
");
```

Input *q* tidak divalidasi dan tidak menggunakan parameter binding sehingga penyerang dapat memasukkan payload seperti:

```
%' OR '1'='1 --
```

yang menghasilkan seluruh data lahan tanpa pembatasan akses. Perbaikan dilakukan dengan mengganti *raw query* menjadi *Query Builder* yang mendukung *prepared statements*. Kode setelah perbaikan:

```
$lahan = DB::table('sw_lahans')
    ->where('id_petani', $petaniId)
    ->where(function ($w) use ($q) {
        $w->where('nama', 'LIKE', "%{$q}%")
          ->orWhere('lokasi', 'LIKE', "%{$q}%");
    })
    ->get();
```

Dengan perubahan ini, input tidak lagi disisipkan langsung ke SQL, setiap parameter diproses melalui mekanisme sanitasi Laravel, dan risiko SQL Injection dapat dieliminasi sepenuhnya.

2) Perbaikan Session Hijacking pada Cookie trusted_device

Sebelum perbaikan, cookie *trusted_device* tidak memiliki atribut HttpOnly sehingga dapat dibaca oleh JavaScript dan berpotensi dicuri melalui serangan XSS. Perbaikan dilakukan dengan menambahkan atribut HttpOnly dan *Secure*, melakukan regenerasi token pada setiap sesi login, serta menambahkan validasi *server-side* untuk memastikan token hanya berlaku pada perangkat yang sah. Setelah perbaikan, nilai cookie tidak dapat lagi diakses dari sisi klien sehingga risiko *session hijacking* menurun secara signifikan.

3) Perbaikan Missing Security Headers

Pemindaian OWASP ZAP menemukan beberapa header keamanan yang belum diterapkan, termasuk HSTS, CSP, *X-Frame-Options*, dan *X-Content-Type-Options*. Perbaikan dilakukan melalui penambahan konfigurasi pada berkas *.htaccess* sebagai berikut:

```
Header always set X-Frame-Options "SAMEORIGIN"
Header always set X-Content-Type-Options "nosniff"
Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains"
Header always set Content-Security-Policy "default-src 'self';"
```

Penambahan header ini memperkuat perlindungan terhadap *clickjacking*, *MIME sniffing*, *downgrade attack*, serta reduksi risiko XSS pasif.

4) Perbaikan Mekanisme Brute Force (Rate Limiting)

Sebelum perbaikan, endpoint login tidak memiliki pembatasan percobaan login sehingga memungkinkan serangan brute force dan password spraying. Middleware *Throttle Requests* ditambahkan pada route login:

 ISSN:2302-8149
e-ISSN:2540-9719

```
Route::post('/login', [AuthController::class, 'login'])
    ->middleware('throttle:5,1');
```

Konfigurasi ini membatasi percobaan login maksimum lima kali per menit per alamat IP. Jika melebihi batas, server menolak permintaan dengan HTTP 429 (*Too Many Attempts*). Perbaikan ini mengurangi risiko *brute force* secara signifikan. Tabel 6 menunjukkan ringkasan perubahan sebelum dan sesudah remediation.

Tabel 6 Perbandingan kondisi kerentanan sebelum dan sesudah perbaikan

Kerentanan	Sebelum	Setelah
SQL Injection	Query raw tanpa sanitasi	Prepared statements + validasi input
trusted_device	Tidak ada HttpOnly	HttpOnly + regen token
Missing Headers	HSTS, CSP, X-Frame-Options tidak ada	Seluruh header ditambahkan
Brute Force	Tidak ada rate limit	Penambahan throttle + blok IP

Perbaikan ini diverifikasi melalui tahap *retesting*, dan seluruh kerentanan dinyatakan telah tertutup atau turun ke tingkat risiko rendah (*Low*).

4.7 Hasil Pengujian Ulang (Retesting)

Tahap retesting dilakukan setelah seluruh proses mitigasi diterapkan pada aplikasi, mencakup perbaikan kode program, penyesuaian konfigurasi server, serta penguatan mekanisme autentikasi dan manajemen sesi. Tujuan retesting adalah memastikan bahwa setiap kerentanan yang sebelumnya berhasil dieksploitasi tidak lagi dapat dimanfaatkan dengan metode yang sama. Hasil retesting disajikan pada Tabel 7, yang menunjukkan perubahan tingkat risiko sebelum dan sesudah perbaikan.

Tabel 7 Hasil retesting kerentanan

Kerentanan	Status Sebelum	Status Sesudah	Hasil Retesting
SQL Injection	High	Closed	Query tidak dapat dimanipulasi, payload ditolak
Session Hijacking	High	Closed	Cookie tidak dapat digunakan ulang; HttpOnly aktif
Brute Force Login	Medium	Low	Rate limiting aktif; percobaan berulang diblokir
Security Misconfiguration	Medium	Low	Header keamanan lengkap dan tervalidasi

Berdasarkan hasil *retesting*, seluruh kerentanan pada tingkat risiko *High* dan *Medium* berhasil diturunkan menjadi *Low* atau dinyatakan *Closed.* Tidak ada skenario eksploitasi sebelumnya yang dapat direplikasi setelah perbaikan diterapkan. Dengan demikian, aplikasi dinyatakan telah memenuhi standar keamanan minimum dalam konteks penelitian ini dan siap digunakan dalam lingkungan operasional.

4.8 Rekomendasi Perbaikan Keamanan (Security Improvement Recommendations)

Selain mitigasi yang telah dilakukan pada tahap remediation, penelitian ini memberikan rekomendasi penguatan keamanan lanjutan untuk memastikan sistem tetap terlindungi dari potensi ancaman di masa mendatang. Rekomendasi disusun berdasarkan OWASP ASVS, OWASP Top 10, serta praktik terbaik keamanan aplikasi web.

(1) Penguatan Autentikasi dan Mekanisme Login

Rekomendasi meliputi penerapan CAPTCHA pada form login untuk mencegah brute force berbasis bot, penggunaan *rate limiting* adaptif untuk membatasi percobaan autentikasi, serta penerapan mekanisme *account lockout* setelah sejumlah percobaan gagal. Kebijakan password perlu diperketat sesuai OWASP *Password Policy*, dan proses pengiriman OTP harus dilakukan hanya setelah verifikasi password berhasil untuk menghindari premature OTP *issuance.*

(2) Penguatan Manajemen Sesi

Seluruh cookie, termasuk trusted_device, perlu menerapkan atribut HttpOnly dan Secure, serta SameSite=Strict untuk menekan risiko CSRF. Regenerasi sesi harus dilakukan setelah login atau perubahan hak akses. *Cookie trusted*_device disarankan memiliki masa berlaku terbatas dan dikaitkan dengan device fingerprint untuk mengurangi risiko penyalahgunaan.

(3) Validasi Input dan Penanganan Query

 ISSN:2302-8149
e-ISSN:2540-9719

Prepared statements harus digunakan secara konsisten di seluruh modul untuk mencegah *SQL Injection.* Validasi *server-side* perlu diterapkan pada setiap parameter input, khususnya pada fitur pencarian dan filter dinamis. Karakter berbahaya yang tidak relevan dengan fungsi input harus ditolak untuk memperkuat keamanan.

(4) Hardening Konfigurasi Server

Server perlu mengaktifkan header keamanan seperti *Content-Security-Policy*, *X-Frame-Options*, *X-Content-Type-Options*, dan HSTS. Informasi *server signature* dan versi Apache/PHP harus dinonaktifkan untuk mengurangi informasi yang dapat membantu attacker. Firewall aplikasi (WAF) direkomendasikan untuk mendeteksi pola serangan–termasuk manual injection.

(5) Pengurangan Risiko Akses Tidak Sah

Aplikasi disarankan menerapkan kontrol akses berbasis peran (RBAC) secara ketat serta mencatat aktivitas pengguna secara komprehensif, mencakup percobaan login gagal, perubahan data kritis, dan akses ke endpoint sensitif.

(6) Pengujian Keamanan Berkala

Penelitian merekomendasikan pelaksanaan *penetration testing* secara rutin untuk mengidentifikasi kerentanan baru. Pemindaian otomatis menggunakan OWASP ZAP atau Burp Suite perlu dilakukan pada setiap pembaruan (per-deploy). Selain itu, penerapan *Secure Software Development Lifecycle (Secure SDLC)* diperlukan sebagai pendekatan berkelanjutan yang memastikan aspek keamanan dipertimbangkan sejak fase perencanaan, pengembangan, pengujian, hingga pemeliharaan aplikasi.

Temuan penelitian ini sejalan dengan hasil studi Diana et al. [5], yang menunjukkan bahwa aplikasi publik umumnya masih menghadapi kerentanan pada aspek *Security Misconfiguration* dan *Authentication Failures*. Namun, penelitian ini memiliki nilai lebih karena tidak berhenti pada tahap identifikasi, tetapi melanjutkan hingga proses remediation dan retesting untuk memastikan setiap kerentanan berhasil ditutup. Hal ini berbeda dengan penelitian Sebrina [6] dan Putra [10], yang fokus pada pemetaan kerentanan tanpa melakukan verifikasi ulang setelah perbaikan. Dengan demikian, penelitian ini memberikan kontribusi tambahan berupa bukti empiris mengenai efektivitas mitigasi pada aplikasi web berbasis Laravel yang digunakan dalam konteks operasional agribisnis.

Keunikan penelitian ini terletak pada penggunaan kombinasi pendekatan *grey-box* dan *white-box* yang memungkinkan analisis keamanan dilakukan secara lebih menyeluruh terhadap struktur aplikasi, mekanisme autentikasi, serta konfigurasi server. Tidak seperti penelitian sebelumnya yang hanya berfokus pada pemetaan kerentanan, penelitian ini secara langsung menerapkan proses remediation di lingkungan produksi untuk memastikan bahwa setiap temuan dapat diperbaiki secara nyata dan terukur. Selain itu, penelitian ini memetakan seluruh kerentanan berdasarkan kategori OWASP Top 10 dan mengevaluasi tingkat risikonya menggunakan *OWASP Risk Rating Methodology*, sehingga hasil analisis tidak hanya bersifat kualitatif tetapi juga kuantitatif.

Aspek kebaruan lainnya adalah pelaksanaan retesting setelah proses mitigasi, yang memberikan bukti empiris bahwa langkah perbaikan benar-benar menutup celah yang sebelumnya dapat dieksploitasi. Pendekatan ini jarang ditemui dalam penelitian serupa yang biasanya berhenti pada tahap identifikasi kerentanan tanpa verifikasi ulang. Penelitian ini juga memberikan kontribusi pada literatur keamanan aplikasi web di sektor agribisnis—bidang yang masih minim eksplorasi—khususnya terkait pengujian keamanan pada aplikasi pencatatan panen berbasis Laravel yang digunakan dalam operasi nyata.

5 Kesimpulan

Penelitian ini bertujuan untuk mengevaluasi dan meningkatkan keamanan Sistem Informasi Pencatatan Panen Sawit SawitGoDigi melalui pendekatan grey-box dan white-box testing berbasis standar OWASP. Hasil penelitian menunjukkan bahwa beberapa kerentanan kritis berhasil diidentifikasi dan diverifikasi, meliputi SQL Injection pada fitur pencarian, session hijacking melalui cookie trusted_device, serta kelemahan autentikasi berupa ketiadaan mekanisme pembatasan percobaan login dan premature OTP issuance pada salah satu role pengguna. Penilaian risiko menggunakan OWASP Risk Rating Methodology menunjukkan bahwa SQL Injection dan session hijacking berada pada tingkat risiko High, sedangkan brute force dan security misconfiguration berada pada kategori Medium dan Low, yang selanjutnya berhasil diturunkan ke tingkat risiko Low setelah dilakukan proses remediation dan retesting. Temuan ini mengindikasikan bahwa penerapan

 ISSN:2302-8149
e-ISSN:2540-9719

metodologi pengujian keamanan yang komprehensif mencakup eksploitasi, perbaikan, dan verifikasi ulang efektif dalam meningkatkan keamanan aplikasi web agribisnis secara terukur. Namun, penelitian ini memiliki keterbatasan karena pengujian hanya difokuskan pada aplikasi web dan belum mencakup aspek keamanan jaringan, infrastruktur fisik, serta pengujian keamanan berkelanjutan. Oleh karena itu, penelitian selanjutnya disarankan untuk memperluas ruang lingkup pengujian ke lapisan infrastruktur, menerapkan continuous security testing, serta mengintegrasikan keamanan ke dalam Secure Software Development Lifecycle (Secure SDLC) guna memperkuat ketahanan sistem agribisnis berbasis web dalam jangka panjang.

Referensi

[1] O. Patricia, A. B. Wahabbi, E. Syafrianto, F. K. Putra, and C. L. Andesti, "Sawit Kita *WebApp Development: Artificial-based E-Learning Intelligence and Community to Drive Actual Information Collaboration and Innovation of Palm Oil Farmers In Indonesia*," *Indonesian Journal of Artificial Intelligence and Data Mining*, Vol. 7, No. 2, p. 460, Jul. 2024, DOI: 10.24014/ijaidm.v7i2.31366.

[2] I. Nursaada, "Perancangan Sistem Informasi Pelaporan Administrasi Harian Pekerja Sawit PT. Sintang Raya berbasis *Web Design of Web-based Daily Administration Reporting Information System for Palm Oil Workers at PT. Sintang Raya*," *Jurnal Intelek Insan Cendikia*, Vol. 2, pp. 3047–7824, May 2025. [Online]. Available: https://jicnusantara.com/index.php/jiic

[3] OWASP Foundation, "*OWASP Top 10 - 2021*," OWASP. [Online]. Available: https://owasp.org/Top10/2021/A00_2021_Introduction/index.html. Accessed: Dec. 10, 2025.

[4] BSSN, "Lanskap Keamanan Siber Indonesia," Jakarta, 2023.

[5] D. Rohmaniah, W. M. Ashari, and A. Dwi Putra, "*Enhancing Website Security using Vulnerability Assessment and Penetration Testing (VAPT) based on OWASP Top Ten*," *Journal of Applied Informatics and Computing (JAIC)*, Vol. 9, No. 2, p. 404, Apr. 2025. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[6] A. F. Sebrina, "Pengujian Celah Keamanan *Website* Posketanmu dengan *Google Penetration Testing* dan *OWASP Top 10*," Skripsi, Univ. Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia, 2024.

[7] F. Indryani, I. Susanto, and D. M. Kusumawardani, "Rekomendasi Perbaikan *Website E-Makaryo* berdasarkan Analisis Kepuasan Pengguna menggunakan Metode *End User Computing Satisfaction (EUCS)*," *Remik*, Vol. 6, No. 3, pp. 465–474, Aug. 2022, DOI: 10.33395/remik.v6i3.11629.

[8] Z. Faizi, A. A. Ridha, U. Singaperbangsa Karawang, J. HSRonggo Waluyo, T. Timur, and J. Barat, "Analisis *Web Security Hole* menggunakan Metode *Penetration Testing Execution and Standard* (Studi Kasus : Universitas Singaperbangsa Karawang)," *Jurnal informasi dan Komputer*, Vol. 11, No. 2, p. 2023.

[9] F. Fachri, "Optimasi Keamanan *Web Server* Terhadap Serangan *Brute-Force* menggunakan *Penetration Testing*," *Jurnal Teknologi Informasi dan Ilmu Komputer*, Vol. 10, No. 1, pp. 51–58, Feb. 2023, DOI: 10.25126/jtiik.2023105872.

[10] G. H. A. Kusuma, "Implementasi *OWASP ZAP* untuk Pengujian Keamanan Sistem Informasi Akademik," *Jurnal Teknologi Informasi*, Vol. 16, No. 2, pp. 2656–0321, Aug. 2022.

[11] Nurasmawati, Mansur, and Nurmi Hidayasari, "Analisis Kerentanan Keamanan pada *Website* Kelurahan Rimba Sekampung dengan menggunakan *Framework OWASP ZAP*," *Jurnal Teknik Industri Terintegrasi*, Vol. 8, No. 4, pp. 3848–3861, 2025.

[12] K. A. Scarfone, M. P. Souppaya, A. Cody, and A. D. Orebaugh, "*Technical Guide to Information Security Testing and Assessment*.," Gaithersburg, MD, 2008. DOI: 10.6028/NIST.SP.800-115.

[13] OWASP Foundation, "*OWASP Web Security Testing Guide*," OWASP. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide. Accessed: Dec. 10, 2025.

[14] OWASP Foundation, "*OWASP Risk Rating Methodology*," OWASP. [Online]. Available: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology. Accessed: Dec. 10, 2025.

[15] C. C. Echefunna, J. Osamor, C. Iwendi, P. Owoh, M. Ashawa, and A. Philip, "*Evaluation of Information Security in Web Application Through Penetration Testing Techniques using OWASP Risk Methodology*," in *2024 International Conference on Advances in Computing Research on*

 ISSN:2302-8149
e-ISSN:2540-9719

Science Engineering and Technology, ACROSET 2024, Institute of Electrical and Electronics Engineers Inc., 2024. DOI: 10.1109/ACROSET62108.2024.10743903.

[16] A. N. Riswanto and D. J. Lubis, "Implementasi *One-Time Password (OTP)* menggunakan *Random Forest*," Jurnal Ilmiah Informatika dan Komputer, Vol. 1, No. 1, pp. 23–29, Jun. 2024.

[17] I. Gumeraruloh Arianto, W. Witanti, H. Ashaury, T. Informatika, and U. Jenderal Achmad Yani, "Sistem Keamanan Otentikasi Pengguna pada Modul *Single Sign On* menggunakan *OAuth 2.0* dan *One Time Password*," 2025. [Online]. Available: http://creativecommons.org/licences/by/4.0/

[18] S. S. Tohidi, D. Cali, M. Tamm, J. Ortiz, J. Salom, and H. Madsen, "*From White-Box to Grey-Box Modelling of the Heat Dynamics of Buildings*," in *E3S Web of Conferences*, EDP Sciences, Dec. 2022. doi: 10.1051/e3sconf/202236212002.

[19] H. Feng, S. Li, H. Shi, and Z. Ye, "*A Comparative Analysis of White Box and Gray Box Adversarial Attacks to Natural Language Processing Systems*," 2024, pp. 640–646. DOI: 10.2991/978-94-6463-540-9_65.