DESAIN UI/UX DAN PENGUATAN KEAMANAN WEBSITE MANAJEMEN INVENTORI MENGGUNAKAN WAF DAN HTTPS

1st Alif Al Ghifari
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
elfghfr@student.telkomuniversity.ac.id

2nd Prajna Deshanta Ibnugraha
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
prajna@tass.telkomuniversity.ac.id

3rd Setia Juli Irzal Ismail
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
julismail@telkomuniversity.ac.id

Abstrak

Penelitian ini mengevaluasi prototipe sistem manajemen inventori berbasis web untuk PT Kereta Cepat Indonesia China (KCIC) dari aspek penerimaan pengguna dan keamanan. Proses perancangan menggunakan pendekatan *User-Centered Design* (UCD) untuk menghasilkan antarmuka yang responsif, yang kemudian divalidasi melalui *User Acceptance Testing*(UAT). Hasil menunjukkan tingkat penerimaan pengguna yang sangat positif, sekaligus mengidentifikasi beberapa area perbaikan pada alur interaksi. Dari sisi keamanan, *Web Application Firewall* (WAF) terbukti memblokir 100% simulasi serangan, dan konfigurasi HTTPS yang diperkuat berhasil mengamankan data saat transit sesuai dengan standar keamanan modern. Prototipe ini memiliki antarmuka yang diterima baik oleh pengguna serta lapisan keamanan dasar yang efektif, namun memerlukan iterasi desain dan optimasi lanjutan sebelum diimplementasikan secara penuh.

Kata kunci: sistem manajemen inventori, *User-Centered Design, User Acceptance Testing, Web Application Firewall, HTTPS*, keamanan web.

I. PENDAHULUAN

Perubahan digital mendorong PT KCIC untuk mengadopsi sistem manajemen inventori berbasis web dalam rangka meningkatkan kecepatan operasional sekaligus memperkuat aspek keamanan data [1][2]. Sebagai operator infrastruktur penting, KCIC membutuhkan sistem yang tidak hanya efisien, tetapi juga tangguh dalam menghadapi ancaman siber.

Pengembangan sistem ini berawal dari fokus pada rancangan antarmuka (UI/UX) yang mudah dipahami agar pengguna merasa nyaman[3]. Namun, karena aplikasi yang terhubung ke internet sangat rentan terhadap serangan, aspek keamanan menjadi prioritas lain yang harus diintegrasikan. Ancaman utama yang umum dijumpai adalah SQL Injection (SQLi) dan Cross-Site Scripting (XSS), dua jenis serangan yang masuk dalam daftar OWASP Top 10 [4][5]. Selain itu, tanpa adanya enkripsi, informasi yang ditransmisikan sangat berisiko untuk dicuri [6].

Atas dasar tersebut, penelitian ini mengembangkan prototipe sistem inventori yang mengombinasikan prinsip desain berpusat pada pengguna dengan penerapan dua mekanisme keamanan dasar, yaitu penggunaan WAF dan pengaktifan HTTPS. Tujuan akhirnya adalah menghadirkan aplikasi yang fungsional, ramah pengguna, serta memiliki lapisan perlindungan yang kuat terhadap ancaman.

II. KAJIAN TEORI

A. Sistem Manajemen Inventori

Sistem informasi manajemen merupakan perangkat penting bagi organisasi modern karena mendukung integrasi data, efisiensi pengelolaan, dan pengambilan keputusan yang lebih akurat [2][7].

B. Desain Antarmuka Pengguna (UI/UX)

Antarmuka pengguna (UI) dan pengalaman pengguna (UX) menjadi faktor kunci dalam keberhasilan aplikasi. UCD menekankan peran pengguna dalam proses desain, sedangkan Design Thinking berfungsi untuk memecahkan masalah dengan solusi kreatif [3][8]. Evaluasi dapat dilakukan melalui UAT, SUS, serta UEQ yang terbukti efektif menilai tingkat kegunaan sistem [9].

C. Keamanan Aplikasi Web

SQLi adalah teknik injeksi yang memanfaatkan celah input untuk mengakses atau memanipulasi basis data [10]. XSS, di sisi lain, memungkinkan eksekusi skrip berbahaya di browser pengguna sehingga data sensitif dapat dicuri [11][12]. Kedua jenis serangan ini dikategorikan sebagai ancaman prioritas oleh OWASP [13][14][15].

D. Web Application Firewall (WAF)

WAF bertindak sebagai penjaga lalu lintas HTTP/S antara klien dan server. Implementasi dengan ModSecurity dan OWASP Core Rule Set (CRS) terbukti mampu mencegah serangan SQLi maupun XSS [16][11][10].

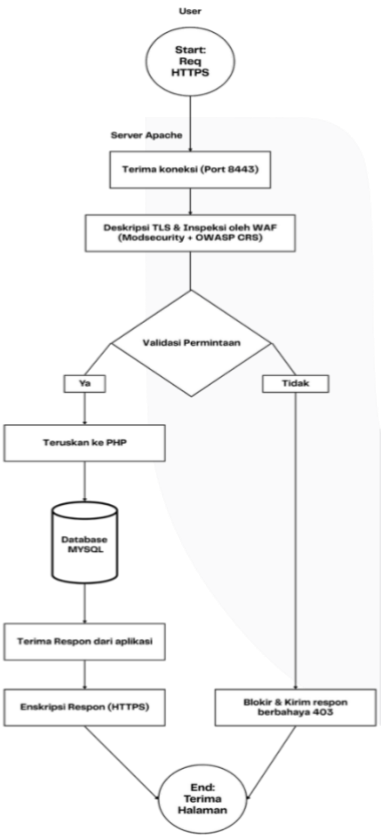
E. Enkripsi HTTPS dan Keamanan TLS

HTTPS adalah protokol komunikasi yang mengandalkan SSL/TLS untuk mengamankan data selama pengiriman. Versi terbaru, TLS 1.3, menggunakan algoritma enkripsi modern seperti AES-GCM dan ChaCha20-Poly1305 yang lebih tahan terhadap serangan [17][18][19][20].

III. METODE

Penelitian ini dilaksanakan melalui tiga tahapan utama. Pertama, analisis kebutuhan dilakukan untuk mengidentifikasi kelemahan sistem awal, termasuk kurangnya validasi input dan komunikasi yang tidak terenkripsi. Kedua, pengembangan prototipe difokuskan pada perancangan UI/UX berbasis UCD menggunakan Figma serta penguatan keamanan dengan ModSecurity WAF dan konfigurasi TLS. Ketiga, pengujian dilakukan untuk mengevaluasi dua aspek sekaligus, yaitu usability dan keamanan.

UAT dilaksanakan menggunakan Maze agar interaksi pengguna terhadap prototipe dapat dianalisis secara visual dan kuantitatif. Sementara itu, uji keamanan dilakukan dengan pendekatan *white-box testing*, di mana prototipe diuji menggunakan serangan SQLi dan XSS, lalu dievaluasi efektivitas WAF. Selain itu, konfigurasi HTTPS divalidasi melalui serangkaian alat seperti Wireshark, nmap, openssl, dan curl. Di bawah ini ditampilkan flowchart sistem keamanan pada Gambar 1.



Gambar 1 Flowchart sistem keamanan

IV. HASIL DAN PEMBAHASAN

Bagian ini menyajikan dan menganalisis hasil-hasil yang diperoleh dari pengujian sistem manajemen inventori berbasis web, meliputi aspek pengalaman pengguna dan keamanan. Temuan disajikan secara deskriptif dan sistematis untuk memudahkan pemahaman.

A. Analisis Pengujian Pengalaman Pengguna (UAT)

UAT menghasilkan bukti bahwa prototipe dapat digunakan dengan baik oleh pengguna. Heatmap login menunjukkan pola klik yang sesuai dengan alur interaksi yang dirancang. Skor SUS mencapai 73 untuk administrator dan 83 untuk pelanggan, yang menandakan tingkat penerimaan yang tinggi. Hasil UEQ juga menegaskan semua dimensi berada dalam kategori *excellent* [3][21].

Gambar 2 Heatmap Tampilan Login

B. Analisis Pengujian Keamanan

Pengujian keamanan memperlihatkan bahwa WAF dapat mendeteksi sekaligus memblokir seluruh 10 payload SQLi dan XSS. Semua percobaan serangan menghasilkan nilai anomali di atas ambang batas sehingga langsung ditolak[14]. Dibawah ini ditampilkan hasil payload pada Tabel 1

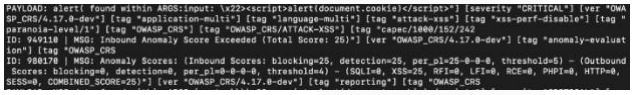
Tabel 1 Hasil Evaluasi Payload Serangan

No	*Payload*	*Jenis*	*Status HTTP*	*Terdeteksi Di Log*	*ID Rule*
1	' OR '1'='1	SQL	403	YA	942100 (libinjection)
2	admin' --	SQL	403	YA	942100 (libinjection)
3	' UNION SELECT null,null,null--	SQL	403	YA	942100
4	'; DROP TABLE users;--	SQL	403	YA	Terdeteksi parsial
5	1 AND SLEEP(5)--+	SQL	403	YA	942100
6	<script>alert(1)</script>	XSS	403	YA	941100, 941110
7	<img src=x onerror=alert(1)>	XSS	403	YA	941110
8	<svg onload=alert(1)>	XSS	403	YA	941110
9	<body onload=alert(1)>	XSS	403	YA	941110
10	"><script>alert(document.cookie)</script>	XSS	403	YA	941110

Dari 10 *payload* pengujian, seluruhnya mendapatkan respons blokir dan dapat diverifikasi secara langsung di log audit ModSecurity, menunjukkan efektivitas pemblokiran 100% terhadap skenario serangan yang diuji. OWASP Core Rule Set (CRS) menggunakan sistem *anomaly scoring* untuk menilai tingkat bahaya dari setiap permintaan. Ambang batas (*threshold*) untuk pemblokiran permintaan berbahaya ditetapkan sebesar 5. Hasil analisis log ModSecurity menunjukkan bahwa seluruh *payload* serangan menghasilkan *anomaly score* ≥ 5, sehingga secara otomatis diblokir oleh WAF. Pengujian ini membuktikan bahwa konfigurasi WAF telah bekerja secara efektif dan dapat diandalkan sebagai garis depan pertahanan.
Beberapa contoh *payload* SQL Injection dan XSS yang berhasil dideteksi dan diblokir oleh WAF, beserta log auditnya, ditunjukkan pada gambar-gambar berikut:

Gambar 3 Log Deteksi Payload SQL Injection '1' OR '1'='1'

Jenis Serangan: Gambar 3 ini mendeteksi serangan SQL Injection (*attack-sqli*). Penyerang menyuntikkan *payload* '1' OR '1'='1' pada parameter ID untuk memanipulasi kueri *database* agar selalu bernilai benar. Mekanisme Pemblokiran: WAF menganggap serangan ini *severity* "CRITICAL" dan memberinya total skor anomali 5. Karena skor ini sama dengan *threshold* 5, WAF memblokir permintaan tersebut.



Gambar 4 Log Deteksi Payload XSS
"><script>alert(document.cookie)</script>

Jenis Serangan: Gambar 4 ini mendeteksi serangan XSS yang mencoba mencuri *cookie* pengguna. Penyerang menggunakan pengkodean (\x22) untuk menyamarkan *payload* <script>alert(document.cookie)</script>. Mekanisme Pemblokiran: WAF berhasil mendekode dan mengidentifikasi *payload* tersembunyi tersebut sebagai *attack-xss* dengan *severity* "CRITICAL". Serangan ini menghasilkan skor anomali 25, yang menyebabkan pemblokiran.

Gambar 5 Log Deteksi Payload XSS <body onload=alert(1)>

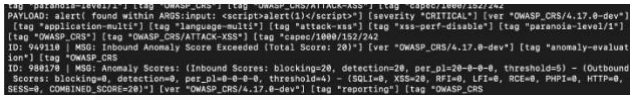
Jenis Serangan: Gambar 5 ini adalah serangan XSS yang menggunakan *tag* <body> dengan *event handler* onload. Skrip alert(1) akan dieksekusi saat halaman web selesai dimuat. Mekanisme Pemblokiran: WAF mengenali penggunaan onload pada *tag* <body> sebagai pola serangan XSS, memberikan *severity* "CRITICAL" dan skor anomali 15.Permintaan ini diblokir.

Gambar 6 Log Deteksi Payload XSS <svg onload=alert(1)>

Jenis Serangan: Gambar 6 ini mendeteksi serangan XSS yang disembunyikan di dalam *tag* <svg>, format gambar vektor. Skrip alert(1) dieksekusi melalui *event handler* onload pada *tag* tersebut. Mekanisme Pemblokiran: WAF mengklasifikasikan ini sebagai *attack-xss* dengan *severity* "CRITICAL". Serangan ini menghasilkan skor anomali 15, yang secara otomatis memicu pemblokiran.

Gambar 7 Log Deteksi Payload XSS <img src=x onerror=alert(1)>

Jenis Serangan: Gambar 7 adalah serangan XSS yang menggunakan *tag* <img>. Skrip alert(1) dieksekusi melalui *event handler* onerror, yang terpicu ketika src (sumber gambar) tidak dapat ditemukan. Mekanisme Pemblokiran: WAF mengenali pola ini sebagai *attack-xss* dengan *severity* "CRITICAL". Serangan ini menghasilkan skor anomali 15, yang menyebabkan pemblokiran otomatis.

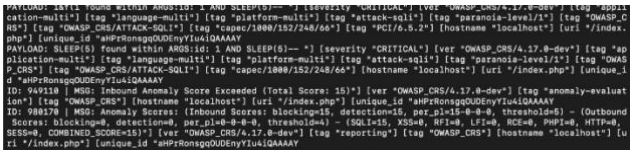


Gambar 8 Log Deteksi Payload XSS <script>alert(1)</script>

Jenis Serangan: Gambar 8 ini mendeteksi serangan Cross-Site Scripting (XSS). Penyerang menyuntikkan skrip <script>alert(1)</script> untuk mengeksekusi kode di sisi peramban pengguna. Mekanisme Pemblokiran: WAF mengidentifikasi *tag* <script> sebagai indikasi serangan XSS (*attack-xss*), memberinya *severity* "CRITICAL" dan skor anomali 20. Skor ini jauh melampaui *threshold* 5, sehingga permintaan diblokir.

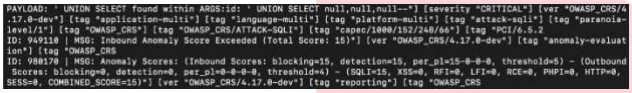
Gambar 9 Log Deteksi Payload SQL Injection '; DROP TABLE users;--

Jenis Serangan: Gambar 9 ini adalah serangan SQL Injection yang sangat berbahaya, menggunakan *payload* '; DROP TABLE users;--. Serangan ini berpotensi menghapus tabel users dari *database*. Mekanisme Pemblokiran: WAF mengidentifikasi *payload* destruktif ini sebagai *severity* "CRITICAL" dengan *tag attack-sqli*. Deteksi ini cukup untuk mengindikasikan bahwa WAF telah mengidentifikasi ancaman serius ini.



Gambar 10 Log Deteksi Payload SQL Injection 1 AND SLEEP(5)--+

Jenis Serangan: Gambar 10 ini mendeteksi SQL Injection berbasis waktu (1 AND SLEEP(5)--). Penyerang menggunakan perintah SLEEP(5) untuk mengukur respons server dan menyimpulkan kebenaran kondisi. Mekanisme Pemblokiran: Serangan ini dianggap *severity* "CRITICAL" dan menghasilkan skor anomali 15, yang menyebabkan pemblokiran otomatis karena melebihi *threshold* 5.



Gambar 11 Log Deteksi Payload SQL Injection ' UNION SELECT null,null,null—

Jenis Serangan: Gambar 11 ini mendeteksi serangan SQL Injection yang lebih canggih, menggunakan *payload* UNION SELECT null,null,--. Tujuannya adalah menggabungkan hasil dari dua kueri *database* untuk mengekstrak data yang tidak seharusnya diakses. Mekanisme Pemblokiran: Serangan ini diklasifikasikan sebagai *severity* "CRITICAL" dan menghasilkan total skor anomali 15, jauh di atas *threshold* 5. Permintaan ini langsung diblokir.

Gambar 12 Log Deteksi Payload SQL Injection admin' –

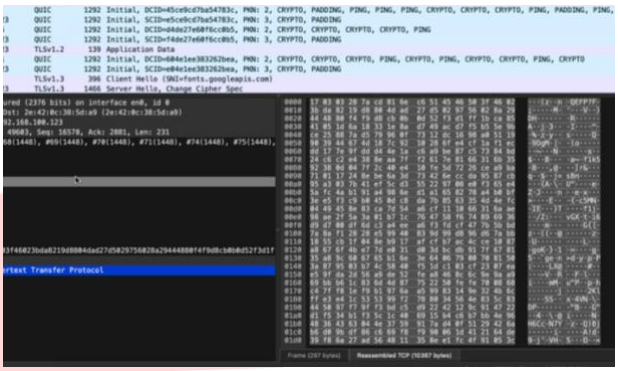
Jenis Serangan: Gambar 12 ini mendeteksi serangan SQL Injection yang menargetkan parameter *username* dengan *payload* admin--. Simbol -- digunakan untuk mengomentari sisa kueri SQL, sehingga penyerang dapat melewati pemeriksaan kata sandi (*password*). Mekanisme Pemblokiran: WAF menganggap serangan ini *severity* "CRITICAL" dan memberinya skor anomali 5. Karena skor ini sama dengan *threshold* 5, WAF segera memblokir permintaan tersebut. Analisis Tambahan: Meskipun *payload* ini terlihat sederhana, WAF berhasil mengidentifikasi pola (admin--) yang umum digunakan dalam upaya *login bypass*. Ini menunjukkan efektivitas WAF dalam melindungi aplikasi dari berbagai teknik serangan, bahkan yang paling dasar.

Pengaturan *threshold* untuk *anomaly scoring* pada WAF juga dikonfirmasi, seperti yang terlihat pada Gambar 13.

```
SecAction \
    "id:900110,\
    phase:1,\
    pass,\
    t:none,\
    nolog,\
    tag:'OWASP_CRS',\
    ver:'OWASP_CRS/4.17.0-dev',\
    setvar:tx.inbound_anomaly_score_threshold=5,\
    setvar:tx.outbound_anomaly_score_threshold=4"
```

Gambar 13 Pengaturan Threshold Anomaly Scoring

Validasi HTTPS & TLS Hardening: Pengujian ini bertujuan untuk memverifikasi kekuatan konfigurasi enkripsi pada server.Selain itu, konfigurasi HTTPS berhasil divalidasi. Data yang ditangkap melalui Wireshark muncul dalam bentuk terenkripsi, nmap mengonfirmasi dukungan TLS 1.2 dan TLS 1.3 dengan cipher suite modern, sedangkan curl membuktikan bahwa header HSTS aktif [22][17][18]. Dibawah ini adalah hasil pengujian wireshark pada Gambar 14.



Gambar 14 Analisis Paket Jaringan dengan Wireshark

Hasil Pengujian Terukur Konfigurasi HTTPS: Pengujian terukur merangkum kekuatan konfigurasi HTTPS berdasarkan parameter standar keamanan modern pada Tabel 2.

Tabel 2 Hasil Pengujian Terukur Konfigurasi HTTPS

No	*Parameter*	*Status*	*Hasil Teknis*	*Keterangan*
1	Versi TLS Modern	Aktif	TLS 1.2 dan TLS 1.3 tersedia	TLSv1.0 dan SSLv3 tidak didukung, sesuai standar keamanan saat ini
2	Cipher Suite Aman	Aman	TLS_AES_256_GCM_SHA384, CHACHA20, ECDHE_RSA_AES_128_GCM	Semua *cipher* mendapatkan nilai Grade A, mendukung Forward Secrecy
3	Validitas Sertifikat	Self-signed	Sertifikat dari mkcert, valid secara lokal	TLSv1.0 dan SSLv3 tidak didukung, sesuai standar keamanan saat ini
4	Header HSTS (HTTPS Policy)	Aktif	Strict-Transport-Security: max-age=63072000; preload	Semua *cipher* mendapatkan nilai Grade A, mendukung Forward Secrecy
5	Renegosiasi TLS	Tidak Didukung	TLS 1.3 tidak mendukung renegosiasi sesi	TLSv1.0 dan SSLv3 tidak didukung, sesuai standar keamanan saat ini

C. Kemungkinan Pengembangan dan Penelitian Ke Depan

Berdasarkan hasil penelitian ini, terdapat beberapa arah pengembangan dan penelitian lanjutan yang dapat dieksplorasi:

1. Penyempurnaan UI/UX Berkelanjutan: Melakukan iterasi desain berdasarkan umpan balik UAT yang lebih mendalam, termasuk pengujian A/B untuk membandingkan berbagai alternatif desain dan mengoptimalkan alur interaksi yang masih menunjukkan hambatan.
2. Penguatan Keamanan Adaptif: Mengembangkan WAF dengan kemampuan deteksi anomali berbasis *machine learning* yang lebih canggih untuk mengidentifikasi serangan *zero-day* atau pola serangan yang belum dikenal.
3. Integrasi Keamanan Otomatis: Mengeksplorasi implementasi alat keamanan otomatis dalam siklus pengembangan perangkat lunak (DevSecOps) untuk memastikan kerentanan diidentifikasi dan diperbaiki sejak dini.
4. Analisis Performa Sistem dengan Keamanan Aktif: Melakukan studi lebih lanjut mengenai dampak kinerja (misalnya, latensi dan *throughput*) dari implementasi WAF dan HTTPS pada skala produksi, serta mencari strategi optimasi untuk meminimalkan *overhead* tanpa mengorbankan keamanan.
5. Penerapan Keamanan Lapisan Lanjut: Mempertimbangkan implementasi kontrol keamanan tambahan seperti *Runtime Application Self-Protection* (RASP) atau *API Security Gateway* untuk perlindungan yang lebih komprehensif.

V. KESIMPULAN

Berdasarkan keseluruhan analisis dan pengujian yang telah dilaksanakan, penelitian ini berhasil mencapai tujuannya dalam mengevaluasi secara komprehensif prototipe sistem manajemen inventori dari dua aspek utama: penerimaan pengguna dan keamanan. Evaluasi pengalaman pengguna melalui User Acceptance Testing (UAT) menunjukkan tingkat penerimaan yang sangat positif dari responden, namun juga berhasil mengidentifikasi beberapa area perbaikan spesifik pada alur interaksi untuk meningkatkan efisiensi. Dari sisi keamanan, implementasi dan validasi kontrol fundamental menunjukkan hasil yang solid. Web Application Firewall (WAF) terbukti efektif dalam memblokir 100% simulasi serangan, dan konfigurasi HTTPS yang diperkuat berhasil mengamankan data saat transit sesuai dengan standar keamanan modern. Dengan demikian, dapat disimpulkan bahwa prototipe aplikasi ini memiliki antarmuka yang diterima dengan baik oleh pengguna dan lapisan keamanan dasar yang efektif, namun memerlukan iterasi desain dan optimasi lebih lanjut sebelum siap untuk implementasi penuh.

Sebagai kontribusi, penelitian ini menghadirkan bukti konsep penerapan WAF dan HTTPS dalam sistem inventori yang dapat dijadikan acuan praktis bagi organisasi serupa. Meskipun demikian, diperlukan iterasi desain dan optimasi lebih lanjut sebelum sistem ini siap diimplementasikan secara penuh.

REFERENSI

[1] Irianto, “Pengantar Sistem Informasi Manajemen.” Accessed: Jun. 18, 2025. [Online]. Available: https://stieamm.ac.id/wp-content/uploads/2017/07/Irianto.pdf

[2] R. Kurniawan Ritonga and R. Firdaus, “JICN: Jurnal Intelek dan Cendikiawan Nusantara PENTINGNYA SISTEM INFORMASI MANAJEMEN DALAM ERA DIGITAL THE IMPORTANCE OF MANAGEMENT INFORMATION SYSTEMS IN THE DIGITAL ERA,” vol. 1, no. 3, 2024, [Online]. Available: https://jicnusantara.com/index.php/jicn

[3] N. D. T. Putri, P. Putra, N. R. Oktadini, A. Meiriza, and P. E. Sevtiyuni, “Penerapan Metode Eye Tracking dalam Evaluasi Pengalaman Pengguna pada Website Traveloka,” *Jurnal Algoritma*, vol. 21, no. 2, pp. 112–123, Nov. 2024, doi: 10.33364/algoritma/v.21-2.2098.

[4] P. Kampanakis and W. Childs-Klein, “The impact of data-heavy, post-quantum TLS 1.3 on the Time-To-Last-Byte of Web connections ,” Internet Society, Feb. 2025. doi: 10.14722/madweb.2024.23010.

[5] S. Toprak and A. Yavuz, “Web Application Firewall Based on Anomaly Detection using Deep Learning,” *Acta Infologica*, vol. 0, no. 0, pp. 0–0, Oct. 2022, doi: 10.26650/acin.1039042.

[6] Anisa Puspita and Muhammad Irwan Padli Nasution, “Manfaat Implementasi Sistem Informasi Manajemen di Organisasi Bisnis,” *Jurnal Penelitian Ekonomi Manajemen dan Bisnis*, vol. 3, no. 1, pp. 153–158, Dec. 2023, doi: 10.55606/jekombis.v3i1.3035.

[7] F. M. Alotaibi and V. G. Vassilakis, “Toward an SDN-Based Web Application Firewall: Defending against SQL Injection Attacks,” *Future Internet*, vol. 15, no. 5, May 2023, doi: 10.3390/fi15050170.

[8] G. Kurniawan, F. Adnan, and J. A. Putra, “Perancangan User Interface dan User Experience Aplikasi E-Commerce Kain Batik pada UMKM Rezti’s Batik Menggunakan Pendekatan Design Thinking,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 3, pp. 551–560, Jul. 2023, doi: 10.25126/jtiik.2023106733.

[9] S. Fleury and N. Chaniaud, “Multi-user centered design: acceptance, user experience, user research and user testing,” *Theor Issues Ergon Sci*, vol. 25, no. 2, pp. 209–224, 2024, doi: 10.1080/1463922X.2023.2166623.

[10] M. Hasan, A. Al-Maliki, and N. Jasim, “Review of SQL injection attacks: Detection, to enhance the security of the website from client-side attacks,” *Int. J. Nonlinear Anal. Appl*, vol. 13, pp. 2008–6822, 2022, doi: 10.22075/ijnaa.2022.6152.

[11] J. R. Tadhani, V. Vekariya, V. Sorathiya, S. Alshathri, and W. El-Shafai, “Securing web applications against XSS and SQLi attacks using a novel deep learning approach,” *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-023-48845-4.

[12] A. Almadira, Y. Pratama, F. Purwani, and K. Kunci, “MELINDUNGI DATA DI DUNIA DIGITAL:

PERAN STATEGIS ENKRIPSI DALAM KEAMANAN DATA PROTECTING DATA IN THE DIGITAL WORLD: THE STRATEGIC ROLE OF ENCRYPTION IN DATA SECURITY INFO ARTIKEL ABSTRAK," *Journal of Scientech Research and Development*, vol. 6, no. 2, 2024, [Online]. Available: https://idm.or.id/JSCR/index.php/JSCR

[13] D. Supriadi, E. Suryadi, R. Muslim, L. Delsi Samsumar, and U. Teknologi Mataram, "IMPLEMENTASI VULNERABILITY ASSESSMENT OWASP (OPEN WEB APPLICATION SECURITY PROJECT) PADA WEBSITE UNIVERSITAS TEKNOLOGI MATARAM."

[14] OWASP Foundation, "OWASP® ModSecurity Core Rule Set." Accessed: Jul. 08, 2025. [Online]. Available: https://owasp.org/www-project-modsecurity-core-rule-set/

[15] Jakobsson and I. Häggström, "Study of the techniques used by OWASP ZAP for analysis of vulnerabilities in web applications," 2022.

[16] M. Idris, I. Syarif, and I. Winarno, "Web Application Security Education Platform Based on OWASP API Security Project," *EMITTER International Journal of Engineering Technology*, pp. 246–261, Dec. 2022, doi: 10.24003/emitter.v10i2.705.

[17] A. Setiawan, M. A. Satrio, I. Madani, R. D. Rachmat, and S. H. Sukma, "Meningkatkan Keamanan Sertifikat Digital dengan Pengaktifan HTTPS," *Journal of Internet and Software Engineering*, vol. 1, no. 4, p. 9, Aug. 2024, doi: 10.47134/pjise.v1i4.3170.

[18] A. Fazlu Rahman, "Hardening A Web Server Infrastructure : An Applied Study of TLS, Reverse Proxy Security, and Attack Simulations," 2025.

[19] Australian Cyber Security Centre, " Implementing Certificates, TLS, HTTPS and Opportunistic TLS." Accessed: Jul. 08, 2025. [Online]. Available: https://www.cyber.gov.au/resources-business-and-government/maintaining-devices-and-systems/system-hardening-and-administration/web-hardening/implementing-certificates-tls-https-and-opportunistic-tls

[20] R. Serrano, C. Duran, M. Sarmiento, C. K. Pham, and T. T. Hoang, "ChaCha20–Poly1305 Authenticated Encryption with Additional Data for Transport Layer Security 1.3," *Cryptography*, vol. 6, no. 2, Jun. 2022, doi: 10.3390/cryptography6020030.

[21] T. Daffa, A. Dakhilullah, and B. Suranto, "Penerapan Metode User Centered Design Pada Perancangan Pengalaman Pengguna Aplikasi I-Star."

[22] A. Haikal, S. D. Putra, and N. Nelmiawati, "Analisis terhadap Enkripsi Data SSL di MySQL: Menguji Keamanan In-Transit," *ROUTERS: Jurnal Sistem dan Teknologi Informasi*, pp. 79–84, Jan. 2024, doi: 10.25181/rt.v2i2.3446.