Jurnal Minfo Polgan (JMP)
Volume 15, Nomor 2, Mei-Agustus 2026
DOI : https://doi.org/10.33395/jmp.v15i2.16567
e-ISSN : 2797-3298
p-ISSN : 2089-9424

Analisis Kerentanan Keamanan pada Integrasi API Generative AI dalam Aplikasi Berbasis Web

Jimmy
Fakultas Sains dan Teknologi, Teknik Informatika, Universitas IBBI, Medan, Indonesia
*Korespondensi: jim8470@gmail.com

Submit : 28 Mei 2026 | **Diterima** : 03 Juli 2026 | **Terbit** : 09 Juli 2026

ABSTRACT

The integration of Generative Artificial Intelligence (GenAI) Application Programming Interfaces (APIs) in web-based applications introduces new attack surfaces not fully addressed by conventional web security practices. This research aims to analyze security vulnerabilities emerging from GenAI API integration in web applications through a Systematic Literature Review (SLR) combined with an experimental framework. The SLR methodology, guided by Kitchenham and Charters, identified, evaluated, and synthesized 42 primary articles from an initial pool of 487 publications across IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, arXiv, and Google Scholar. The analysis reveals ten primary vulnerability categories based on the OWASP Top 10 for LLM Applications 2025, including prompt injection, sensitive information disclosure, supply chain risks, data and model poisoning, improper output handling, excessive agency, system prompt leakage, vector and embedding weaknesses, misinformation, and unbounded consumption. Each vulnerability was analyzed within the context of API integration in web applications, accompanied by attack scenarios, potential impacts, and mitigation strategies. A five-phase Security Testing Framework (STF) is proposed, encompassing planning, reconnaissance, vulnerability assessment, exploitation, and reporting phases tailored to GenAI API integration, with the framework being conceptual in nature. The primary contribution is a comprehensive mapping of the GenAI security threat landscape in the web context, providing a reference for developers, security practitioners, and researchers in building more secure GenAI-based web applications.

Keywords: *API security; generative AI; LLM applications; OWASP LLM Top 10; prompt injection; security vulnerabilities; web applications*

ABSTRAK

Integrasi Application Programming Interface (API) Generative Artificial Intelligence (GenAI) dalam aplikasi berbasis web menciptakan permukaan serangan baru yang belum sepenuhnya teratasi oleh praktik keamanan web konvensional. Penelitian ini bertujuan untuk menganalisis kerentanan keamanan yang muncul pada integrasi API GenAI dalam aplikasi berbasis web melalui pendekatan Systematic Literature Review (SLR) yang dikombinasikan dengan kerangka kerja eksperimental. Metodologi SLR mengikuti pedoman Kitchenham dan Charters, mengidentifikasi dan mensintesis 42 artikel primer dari 487 publikasi yang diperoleh dari IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, arXiv, dan Google Scholar. Hasil analisis mengungkapkan sepuluh kategori kerentanan utama berdasarkan OWASP Top 10 for LLM Applications 2025, meliputi prompt injection, kebocoran informasi sensitif, rantai pasok, peracunan data dan model, penanganan output yang tidak tepat, otoritas berlebihan, kebocoran prompt sistem, kelemahan vektor dan embedding, misinformasi, serta konsumsi tanpa batas. Setiap kerentanan dianalisis dalam konteks integrasi API pada aplikasi web, disertai skenario serangan, dampak potensial, dan strategi mitigasi. Penelitian ini juga mengusulkan Security Testing Framework (STF) lima fase yang mencakup perencanaan, pengintaian, penilaian kerentanan, eksploitasi, dan pelaporan yang disesuaikan untuk integrasi API GenAI. Kontribusi utama penelitian ini adalah pemetaan komprehensif lanskap ancaman keamanan GenAI dalam konteks web, yang dapat menjadi acuan

bagi pengembang, praktisi keamanan, dan peneliti dalam membangun aplikasi web berbasis GenAI yang lebih aman.

Kata Kunci: AI generatif; Aplikasi LLM; Aplikasi Web; Keamanan API; Kerentanan Keamanan; OWASP LLM Top 10; *Prompt Injection*

PENDAHULUAN

Perkembangan Generative Artificial Intelligence (GenAI) telah mentransformasi lanskap pengembangan aplikasi web secara fundamental. Model bahasa besar (Large Language Model/LLM) seperti OpenAI GPT-4, Google Gemini, Anthropic Claude, dan Meta Llama telah menjadi komponen integral dalam arsitektur aplikasi web modern, memungkinkan fungsionalitas canggih seperti chatbot cerdas, pembuatan konten otomatis, analisis data natural language, dan personalisasi pengalaman pengguna (Brown et al., 2020). Akses terhadap kemampuan GenAI ini umumnya difasilitasi melalui API yang disediakan oleh penyedia layanan cloud AI. Menurut laporan Gartner tahun 2024, lebih dari 80% perusahaan teknologi telah mengintegrasikan setidaknya satu layanan GenAI API ke dalam produk dan layanan mereka, dengan proyeksi adopsi mencapai 95% pada tahun 2026 (Gartner, 2024). Di Indonesia, tingginya penetrasi internet nasional yang terus meningkat (APJII, 2024) memberikan fondasi bagi adopsi teknologi AI generatif dalam aplikasi web pada sektor e-commerce, fintech, layanan kesehatan digital, dan pemerintahan elektronik.

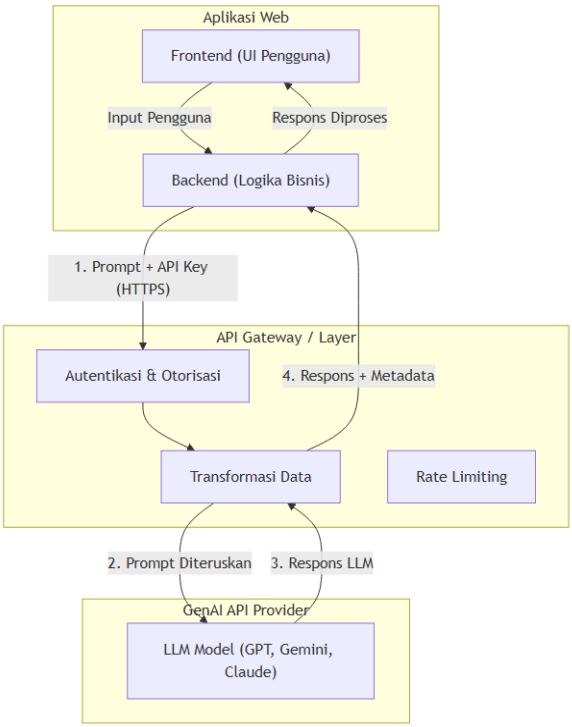
Namun, di balik percepatan adopsi ini, muncul tantangan keamanan yang signifikan. Integrasi API GenAI dalam aplikasi web menciptakan permukaan serangan baru yang tidak tercakup dalam praktik keamanan aplikasi web tradisional (Chen et al., 2024). Berbeda dengan integrasi API konvensional, API GenAI memiliki karakteristik unik yang memperkenalkan vektor ancaman spesifik: model bahasa dapat dimanipulasi melalui prompt injection untuk menghasilkan output berbahaya, data sensitif dapat bocor melalui respons model, dan kredensial API yang terekspos dapat disalahgunakan untuk akses tidak sah (OWASP Foundation, 2025). Open Web Application Security Project (OWASP) pada tahun 2025 merilis pembaruan daftar Top 10 for LLM Applications yang mengidentifikasi sepuluh risiko keamanan paling kritis pada aplikasi berbasis LLM (OWASP Gen AI Security Project, 2025). Liu et al. (2023) mendemonstrasikan serangan prompt injection terhadap 36 aplikasi web yang menggunakan API GPT-4 dengan tingkat keberhasilan 86,1%. Truffle Security (2025) menemukan lebih dari 12.000 kredensial API yang masih aktif dalam data pelatihan model DeepSeek, sementara GitGuardian (2026) melaporkan 29 juta kebocoran kredensial pada tahun 2025 dengan kredensial agen AI sebagai kontributor signifikan.

Meskipun literatur terkait keamanan GenAI berkembang pesat, sebagian besar penelitian yang ada berfokus pada aspek keamanan model secara terisolasi—seperti adversarial attack pada model, jailbreaking, dan alignment—tanpa secara spesifik membahas kerentanan yang muncul dari integrasi API GenAI dalam arsitektur aplikasi web (Branch et al., 2022; Chao et al., 2023). Terdapat kesenjangan penelitian dalam memahami bagaimana kerentanan spesifik GenAI berinteraksi dengan kerentanan aplikasi web tradisional, serta bagaimana pengembang dapat secara sistematis mengidentifikasi dan memitigasi risiko tersebut. Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengidentifikasi dan mengklasifikasikan kerentanan keamanan pada integrasi API GenAI dalam aplikasi web, menganalisis karakteristik dan dampak setiap kerentanan, merumuskan strategi mitigasi yang komprehensif, serta mengusulkan kerangka kerja pengujian keamanan untuk integrasi API GenAI. Penelitian ini dibatasi pada kerentanan yang terkait dengan integrasi API GenAI eksternal dalam aplikasi web, menggunakan OWASP Top 10 for LLM Applications 2025 sebagai kerangka acuan utama, dengan kerangka eksperimental yang diusulkan bersifat konseptual.

TINJAUAN PUSTAKA

Generative AI merujuk pada sistem kecerdasan buatan yang mampu menghasilkan konten baru—termasuk teks, gambar, kode, dan multimedia—berdasarkan pola yang dipelajari dari data pelatihan. LLM seperti GPT (Generative Pre-trained Transformer), Gemini, dan Claude merupakan implementasi GenAI yang paling banyak diadopsi dalam aplikasi web, dengan arsitektur transformer sebagai fondasi teknisnya (Vaswani et al., 2017; Brown et al., 2020). Arsitektur

integrasi API GenAI dalam aplikasi web umumnya mengikuti pola client-server dengan tiga komponen utama: aplikasi web (frontend dan backend) yang memproses input dan output pengguna, API gateway/layer sebagai middleware autentikasi dan transformasi data, serta GenAI API provider seperti OpenAI API atau Google Vertex AI yang menyediakan akses ke model LLM melalui endpoint REST atau gRPC (OpenAI, 2025). Gambar 1 mengilustrasikan arsitektur tiga lapis tersebut beserta alur komunikasi antara komponen.



Gambar 1. Arsitektur Integrasi API GenAI dalam Aplikasi Web
Arsitektur Integrasi API GenAI

Alur komunikasi tipikal: aplikasi web menerima input pengguna, backend memproses dan menggabungkannya dengan prompt sistem, mengirim permintaan ke API GenAI melalui HTTP request dengan API key, model LLM memproses prompt, dan respons dikembalikan ke backend untuk post-processing sebelum disajikan kepada pengguna (Liu et al., 2023).

Keamanan aplikasi web secara tradisional difokuskan pada perlindungan terhadap kerentanan yang tercantum dalam OWASP Top 10 for Web Applications, meliputi Broken Access Control, Cryptographic Failures, Injection, Insecure Design, Security Misconfiguration, Vulnerable and Outdated Components, Identification and Authentication Failures, Software and Data Integrity Failures, Security Logging and Monitoring Failures, dan Server-Side Request Forgery (OWASP Foundation, 2021). Meskipun kerentanan tradisional ini tetap relevan, integrasi GenAI memperkenalkan dimensi keamanan baru yang tidak sepenuhnya tertangani oleh kontrol keamanan web konvensional (Das et al., 2025). Sebagai contoh, Web Application Firewall (WAF) tradisional tidak dapat mendeteksi serangan prompt injection karena payload disampaikan dalam format natural language yang legitimate, berbeda dengan pola serangan SQL injection atau XSS yang memiliki signature teknis yang dapat dikenali (Greshake et al., 2023).

OWASP merilis daftar Top 10 for LLM Applications pertama kali pada tahun 2023 dan memperbaruinya secara signifikan pada tahun 2025. Pembaruan ini mencerminkan pergeseran fokus dari risiko pada level model ke risiko pada level aplikasi dan ekosistem yang lebih luas. Tabel 1 menyajikan ringkasan OWASP Top 10 for LLM Applications 2025 beserta deskripsi singkatnya.

Tabel 1. OWASP Top 10 for LLM Applications 2025

ID	Kategori	Deskripsi
LLM01:2025	Prompt Injection	Manipulasi input pengguna untuk mengubah perilaku LLM
LLM02:2025	Sensitive Information Disclosure	Kebocoran informasi sensitif melalui respons model
LLM03:2025	Supply Chain	Kerentanan pada komponen pihak ketiga dalam rantai pasok LLM
LLM04:2025	Data and Model Poisoning	Manipulasi data pelatihan untuk menanamkan kerentanan
LLM05:2025	Improper Output Handling	Validasi output LLM tidak memadai sebelum digunakan downstream
LLM06:2025	Excessive Agency	Pemberian otonomi berlebihan kepada LLM dalam menjalankan aksi
LLM07:2025	System Prompt Leakage	Kebocoran prompt sistem berisi instruksi dan logika bisnis
LLM08:2025	Vector and Embedding Weaknesses	Kerentanan pada sistem RAG dan basis data vektor
LLM09:2025	Misinformation	Generasi informasi tidak akurat oleh LLM
LLM10:2025	Unbounded Consumption	Konsumsi sumber daya tidak terbatas menyebabkan DoS

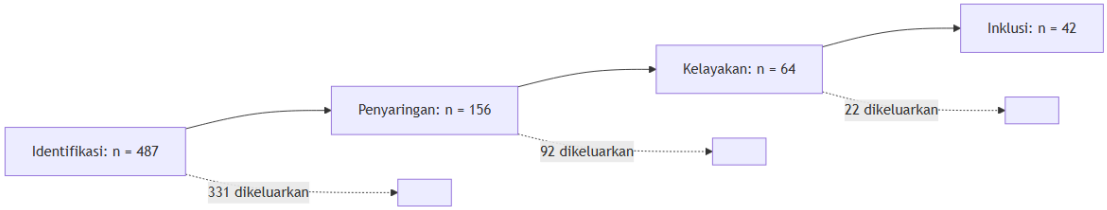
Penelitian terdahulu telah mendokumentasikan berbagai aspek kerentanan keamanan GenAI. Survei komprehensif oleh Das et al. (2025) dan Yao et al. (2024) mengklasifikasikan ancaman keamanan dan privasi LLM ke dalam beberapa kategori, termasuk serangan pada tahap pre-training, fine-tuning, deployment, dan LLM-based agents. Liu et al. (2023) melakukan investigasi terhadap serangan prompt injection pada 36 aplikasi terintegrasi LLM menggunakan toolkit HouYi dan mencapai tingkat keberhasilan 86,1% dalam mencuri prompt asli. Chen et al. (2024) mengidentifikasi tiga kategori utama tantangan keamanan GenAI: model sebagai target serangan, ketergantungan berlebihan pada output GenAI, dan keterbatasan pendekatan keamanan tradisional. Penelitian oleh Gupta et al. (2023) dan Wu et al. (2023) mendokumentasikan risiko kebocoran data di mana model LLM dapat secara tidak sengaja mengungkapkan Personally Identifiable Information (PII) dan informasi bisnis rahasia. Nasr et al. (2023) lebih jauh mendemonstrasikan bahwa serangan ekstraksi data pelatihan dapat secara sistematis mengekstrak informasi sensitif dari model GenAI. Terkait kerentanan pada level API dan agen AI, Liu et al. (2024) mengembangkan kerangka formal untuk benchmarking serangan prompt injection pada sistem terintegrasi LLM, sementara Das et al. (2025) mengidentifikasi risiko kebocoran kredensial dan eskalasi hak akses pada agen AI sebagai area yang memerlukan perhatian lebih lanjut. Namun, sebagian besar penelitian terdahulu membahas kerentanan GenAI secara terpisah—dari perspektif keamanan model, API, atau aplikasi web—tanpa mengintegrasikan ketiga dimensi dalam satu analisis komprehensif. Penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan menyediakan analisis terpadu yang menghubungkan kerentanan spesifik GenAI dengan konteks integrasi API dalam arsitektur aplikasi web.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan mixed-method yang menggabungkan Systematic Literature Review (SLR) dengan pengembangan kerangka kerja eksperimental. Desain ini dipilih untuk memberikan landasan teoretis yang kuat melalui SLR sekaligus menghasilkan instrumen praktis yang dapat digunakan untuk pengujian keamanan di masa mendatang (Creswell & Creswell, 2018).

SLR dilakukan mengikuti pedoman Kitchenham dan Charters (2007) yang terdiri dari tiga tahap utama: perencanaan, pelaksanaan, dan pelaporan. Pada tahap perencanaan, dirumuskan tiga pertanyaan penelitian: kerentanan keamanan apa yang muncul pada integrasi API GenAI dalam aplikasi web, bagaimana karakteristik dan dampak masing-masing kerentanan, serta strategi mitigasi apa yang telah diusulkan dalam literatur. Protokol pencarian diterapkan pada

database akademik bereputasi meliputi IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, arXiv, dan Google Scholar, dengan kombinasi kata kunci "generative AI" atau "LLM" atau "large language model" yang dikombinasikan dengan "API security" atau "security vulnerability" atau "prompt injection" dan "web application" atau "web-based." Kriteria inklusi mencakup publikasi tahun 2022–2025 yang membahas keamanan GenAI/LLM dalam konteks aplikasi atau API, tersedia dalam bahasa Inggris atau Indonesia, dan bersifat peer-reviewed atau preprint bereputasi. Kriteria eksklusi meliputi artikel yang fokus pada keamanan model tanpa konteks aplikasi/API, artikel opini tanpa dasar empiris, dan artikel tanpa teks lengkap.
Tahap pelaksanaan dilakukan dalam empat fase seleksi yang diilustrasikan pada Gambar 2.



Gambar 2. Diagram Alir Seleksi Artikel SLR

Diagram Alir Seleksi Artikel SLR
Fase identifikasi menghasilkan 487 artikel dari pencarian awal. Penyaringan berdasarkan judul dan abstrak mengurangi jumlah menjadi 156 artikel. Pemeriksaan teks penuh menghasilkan 64 artikel yang memenuhi kriteria kelayakan. Penilaian kualitas akhir menghasilkan 42 artikel primer yang dimasukkan dalam sintesis. Data diekstraksi dari setiap artikel menggunakan formulir terstandarisasi yang mencakup metadata publikasi, jenis kerentanan, konteks aplikasi, metodologi penelitian, temuan utama, dan strategi mitigasi yang diusulkan. Data disintesis menggunakan analisis tematik untuk mengidentifikasi kategori kerentanan, pola serangan, dan strategi mitigasi (Braun & Clarke, 2006).

Sebagai pelengkap SLR, penelitian ini mengusulkan Security Testing Framework (STF) untuk pengujian keamanan integrasi API GenAI pada aplikasi web. Kerangka kerja ini dirancang berdasarkan temuan SLR dan diadaptasi dari metodologi penetration testing standar, termasuk *OWASP Testing Guide* (OWASP Foundation, 2024), *PTES* (Penetration Testing Execution Standard, 2014), dan *OSSTMM* (Institute for Security and Open Methodologies, 2010) dengan penambahan komponen spesifik GenAI. STF terdiri dari lima fase. Fase Perencanaan mencakup identifikasi endpoint API GenAI, pemetaan arsitektur integrasi, inventarisasi data flow, dan threat modeling menggunakan STRIDE-LM—ekstensi STRIDE dengan kategori LLM-specific (prompt injection, system prompt leakage) dan Model-specific (data poisoning, embedding attacks). Fase Pengintaian meliputi identifikasi penyedia API dan versi model, analisis konfigurasi keamanan API (CORS headers, mekanisme autentikasi), serta pemindaian endpoint API dan API key dalam kode sumber publik. Fase Penilaian Kerentanan meliputi pengujian prompt injection, information disclosure, API key exposure, output validation, rate limiting, dan kontrol autentikasi/otorisasi. Fase Eksploitasi mencakup simulasi serangan prompt injection untuk ekstraksi prompt sistem, penyalahgunaan API key yang terekspos, privilege escalation melalui otoritas berlebihan, dan simulasi serangan denial of service. Fase Pelaporan menghasilkan klasifikasi kerentanan berdasarkan CVSS v4.0 (FIRST, 2023), rekomendasi mitigasi, dan roadmap perbaikan. Untuk mendukung analisis, dikembangkan matriks pemetaan yang menghubungkan setiap kategori OWASP LLM Top 10 dengan vektor serangan, komponen arsitektur terdampak, kontrol keamanan tradisional, dan strategi mitigasi.

HASIL DAN PEMBAHASAN

Hasil SLR mengonfirmasi bahwa sepuluh kategori OWASP Top 10 for LLM Applications 2025 sepenuhnya relevan dalam konteks integrasi API GenAI pada aplikasi web. Analisis mengidentifikasi bahwa setiap kategori memiliki manifestasi spesifik yang terkait erat dengan arsitektur web, protokol HTTP, dan pola integrasi API. Tabel 2 menyajikan pemetaan kerentanan terhadap komponen arsitektur web yang terdampak.

Tabel 2. Pemetaan Kerentanan OWASP LLM Top 10 terhadap Arsitektur Aplikasi Web

Kerentanan	Frontend	Backend	API Layer	Database	Jaringan
LLM01: Prompt Injection	✓	✓✓✓	✓✓	-	-
LLM02: Sensitive Info Disclosure	✓	✓✓	✓✓✓	✓✓	-
LLM03: Supply Chain	-	✓✓	✓✓✓	-	-
LLM04: Data/Model Poisoning	-	✓	✓✓	✓✓✓	-
LLM05: Improper Output Handling	✓✓✓	✓✓	-	-	-
LLM06: Excessive Agency	-	✓✓✓	✓✓✓	✓	✓
LLM07: System Prompt Leakage	✓	✓✓	✓✓✓	-	-
LLM08: Vector/Embedding Weaknesses	-	✓✓	✓	✓✓✓	-
LLM09: Misinformation	✓✓✓	✓	-	-	-
LLM10: Unbounded Consumption	-	✓✓	✓✓✓	-	✓

Keterangan: ✓ = terdampak rendah, ✓✓ = terdampak sedang, ✓✓✓ = terdampak tinggi

Analisis per kategori kerentanan memberikan gambaran mendalam tentang masing-masing risiko. Prompt Injection (LLM01) merupakan kerentanan di mana input pengguna yang tidak terpercaya dimasukkan ke dalam prompt LLM dan menyebabkan model mengabaikan instruksi sistem atau menjalankan tindakan yang tidak diinginkan. Terdapat dua jenis: Direct Prompt Injection di mana penyerang langsung memasukkan instruksi berbahaya melalui antarmuka pengguna, dan Indirect Prompt Injection di mana konten berbahaya disisipkan melalui sumber eksternal seperti halaman web atau dokumen. Vektor serangan mencakup formulir input pengguna, upload dokumen dengan teks tersembunyi, URL parameter, cookie dan header HTTP, serta webhook yang memasukkan data tidak terpercaya. Dampaknya meliputi ekstraksi prompt sistem proprietary, bypass filter konten, eksekusi fungsi tidak sah melalui tool calling, dan penyalahgunaan komputasi LLM. Strategi mitigasi meliputi input sanitization (RSE, XML Tagging), prompt hardening, penggunaan model LLM terpisah untuk evaluasi upaya injeksi, least privilege pada tool calling, serta output encoding sebelum rendering.

Sensitive Information Disclosure (LLM02) terjadi ketika informasi sensitif—PII, data keuangan, rahasia dagang, kredensial API—bocor melalui interaksi dengan LLM. Kebocoran dapat terjadi melalui data pengguna dalam log API provider, API key tertanam di frontend JavaScript, data pelatihan terekstrak, dan informasi sesi dalam respons LLM. Dampaknya mencakup pelanggaran privasi dan regulasi (UU PDP, GDPR), penyalahgunaan kredensial, eksfiltrasi data proprietary, dan kerusakan reputasi. Mitigasi mencakup data minimization, penyimpanan API key di environment variable atau secrets manager, implementasi Data Loss Prevention (DLP), enkripsi TLS 1.3 dan AES-256, serta audit logging.

Supply Chain (LLM03) mencakup risiko dari ketergantungan pada komponen pihak ketiga: model pre-trained, dataset fine-tuning, library integrasi API, dan plugin. Skenario serangan meliputi library client versi rentan, model fine-tuned dengan backdoor, dataset terkontaminasi, dan dependensi transitif rentan. Mitigasi meliputi pemeliharaan Software Bill of Materials (SBOM), dependency scanning, verifikasi provenance model, vendor security assessment, dan pembaruan berkala terhadap dependensi.

Data and Model Poisoning (LLM04) terjadi ketika penyerang memanipulasi data pelatihan, fine-tuning, atau embedding untuk menanamkan kerentanan atau backdoor. Serangan melalui injeksi data ke dataset fine-tuning, manipulasi embedding RAG, poisoning via user-generated content, dan serangan pipeline data. Dampaknya: degradasi output, backdoor teraktivasi trigger, bias, dan output berbahaya sulit dideteksi. Mitigasi mencakup validasi dan verifikasi integritas data, adversarial training, pemantauan perilaku model berkelanjutan, kontrol akses ketat pada pipeline data, dan pelaksanaan fine-tuning dalam lingkungan terisolasi.

Improper Output Handling (LLM05) terjadi ketika output LLM tidak divalidasi atau disanitasi dengan benar sebelum diteruskan ke komponen downstream, memungkinkan serangan XSS, SSTI, SQL injection, atau command injection. Skenario serangan meliputi output LLM yang mengandung JavaScript berbahaya dirender tanpa sanitasi, ekspresi template yang dimasukkan ke template engine, markdown dengan skrip tersembunyi, dan output yang digunakan sebagai parameter query database. Mitigasi kritis meliputi contextual output encoding, Content Security Policy (CSP), sanitasi HTML/Markdown (DOMPurify), parameterized queries, dan validasi output terhadap skema yang diharapkan.

Excessive Agency (LLM06) terjadi ketika LLM diberikan otonomi berlebihan untuk menjalankan aksi atas nama pengguna atau sistem, termasuk akses ke API eksternal, database, atau fungsi sistem tanpa kontrol memadai. Dampaknya mencakup akses tidak sah ke data, modifikasi data tidak sah, eskalasi hak akses, dan eksekusi transaksi finansial tidak sah. Mitigasi mencakup principle of least privilege pada tool LLM, human-in-the-loop untuk aksi berisiko tinggi, verifikasi otorisasi per aksi, rate limiting per aksi, dan audit trail.

System Prompt Leakage (LLM07) merupakan kebocoran prompt sistem yang berisi instruksi proprietary, logika bisnis, dan konfigurasi. Kebocoran dapat terjadi melalui prompt injection bertarget, analisis diferensial respons, eksploitasi fitur "repeat" atau "summarize", dan error message yang mengandung fragmen prompt. Mitigasi meliputi minimalisasi informasi sensitif dalam prompt, prompt hardening, model LLM guard terpisah, dynamic prompt assembly, dan output filtering.

Vector and Embedding Weaknesses (LLM08) berkaitan dengan kelemahan pada sistem RAG, basis data vektor, dan komponen embedding. Serangan mencakup injeksi dokumen berbahaya ke vector database, akses tidak sah ke embedding, manipulasi similarity search, dan adversarial embedding. Mitigasi mencakup kontrol akses pada vector database, validasi dokumen sebelum insertion, enkripsi embedding, audit berkala, dan rate limiting operasi.

Misinformation (LLM09) merujuk pada generasi output faktual tidak akurat atau halusinasi oleh model. Dampaknya mencakup pengambilan keputusan keliru, kerusakan reputasi, risiko hukum, dan degradasi kepercayaan. Mitigasi: grounding dengan RAG, sitasi dan sumber, factual verification layer, komunikasi ketidakpastian, dan domain-specific fine-tuning.

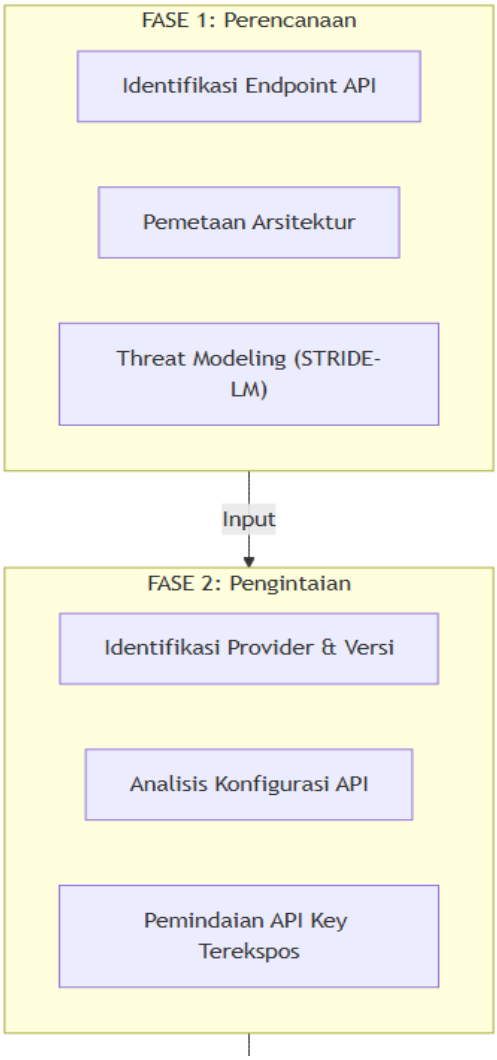
Unbounded Consumption (LLM10) terjadi ketika tidak ada batasan pada penggunaan sumber daya, menyebabkan DoS, pembengkakan biaya, atau degradasi kinerja. Vektor: token bombing, permintaan berulang masif, infinite loop agent AI, dan DDoS. Mitigasi: rate limiting, token budget, batasan panjang input/output, cost monitoring real-time, dan request throttling.

Berdasarkan analisis di atas, Tabel 3 menyajikan matriks mitigasi komprehensif yang menghubungkan setiap kerentanan dengan kontrol keamanan yang direkomendasikan.

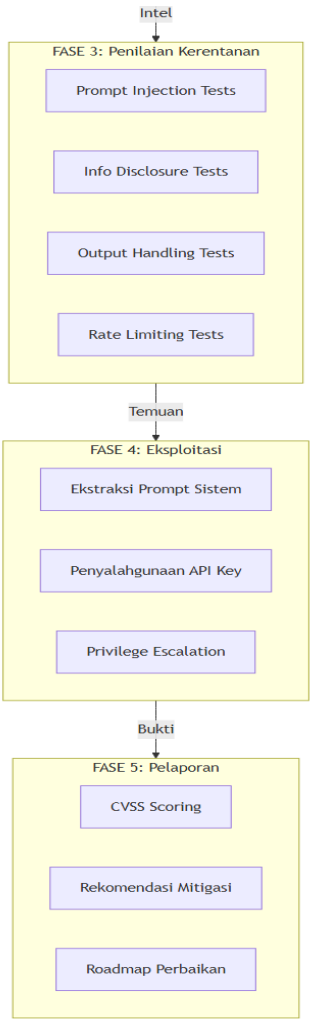
Tabel 3. Matriks Mitigasi Kerentanan Integrasi API GenAI

Kerentanan	Input Validation	Output Sanitization	Access Control	Monitoring	Architecture
LLM01	✓✓✓	✓✓	✓	✓✓	✓✓
LLM02	✓✓	✓✓✓	✓✓✓	✓✓✓	✓✓
LLM03	-	-	✓✓	✓✓✓	✓✓✓
LLM04	✓✓✓	-	✓✓✓	✓✓✓	✓✓
LLM05	-	✓✓✓	-	✓	✓
LLM06	-	-	✓✓✓	✓✓✓	✓✓✓
LLM07	✓✓	✓✓	✓	✓✓	✓✓✓
LLM08	✓✓	✓	✓✓✓	✓✓	✓✓
LLM09	✓	✓✓✓	-	✓✓	✓✓
LLM10	✓✓	-	✓✓	✓✓✓	✓✓✓

Sebagai kontribusi praktis, Security Testing Framework (STF) yang diusulkan menyediakan metodologi terstruktur untuk pengujian keamanan integrasi API GenAI pada aplikasi web. Gambar 3 mengilustrasikan arsitektur STF dengan kelima fase dan modul pengujiannya.



Gambar 3. Arsitektur Security Testing Framework (STF) untuk Integrasi API GenAI



Gambar 4 Arsitektur Security Testing Framework

Temuan penelitian ini memiliki implikasi penting bagi ekosistem pengembangan aplikasi web di Indonesia. Pertumbuhan adopsi GenAI di Indonesia menunjukkan tren positif seiring dengan ekspansi ekosistem digital nasional. Kepatuhan terhadap UU PDP No. 27 Tahun 2022 menjadi faktor kritis karena kebocoran data akibat integrasi API GenAI yang tidak aman dapat berujung pada sanksi signifikan (Republik Indonesia, 2022). Direkomendasikan agar perusahaan mengadopsi STF dalam Secure Software Development Lifecycle (SSDLC), memasukkan pelatihan keamanan GenAI dalam program awareness siber, serta mendorong BSSN mengembangkan panduan keamanan untuk integrasi AI. Penelitian ini mengakui beberapa keterbatasan, termasuk sifat konseptual dari kerangka eksperimental yang belum divalidasi secara empiris pada aplikasi nyata, fokus pada API GenAI berbasis teks tanpa mencakup modalitas lain seperti image generation dan multimodal, serta lanskap keamanan GenAI yang berevolusi sangat cepat sehingga memerlukan pembaruan berkala.

KESIMPULAN

Penelitian ini telah menganalisis kerentanan keamanan pada integrasi API Generative AI dalam aplikasi berbasis web melalui Systematic Literature Review (SLR) terhadap 42 artikel primer, mengonfirmasi relevansi sepuluh kategori OWASP Top 10 for LLM Applications 2025 dalam konteks integrasi API web. Setiap kerentanan menunjukkan manifestasi spesifik yang berbeda dari kerentanan aplikasi web tradisional, dengan Prompt Injection (LLM01) teridentifikasi paling

kritis (tingkat keberhasilan 86,1% pada aplikasi nyata menurut Liu et al., 2023), diikuti oleh Sensitive Information Disclosure (LLM02) yang berdampak langsung pada kepatuhan regulasi. Strategi mitigasi efektif memerlukan pendekatan pertahanan berlapis pada level input, output, akses, arsitektur, dan monitoring. Kontribusi utama meliputi pemetaan komprehensif lanskap ancaman GenAI dalam konteks API web, analisis mendalam karakteristik dan mitigasi setiap kerentanan, Security Testing Framework (STF) lima fase untuk evaluasi postur keamanan, serta matriks mitigasi yang menghubungkan kerentanan dengan kontrol keamanan spesifik. Penelitian lanjutan direkomendasikan untuk validasi empiris STF pada studi kasus aplikasi web Indonesia, perluasan cakupan ke modalitas GenAI lain, dan pengembangan automated security testing tools untuk integrasi API GenAI.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi/Fakultas dan institusi atas fasilitas dan sumber daya yang disediakan selama penelitian ini. Ucapan terima kasih juga ditujukan kepada para rekan peneliti dan praktisi keamanan siber yang telah memberikan masukan berharga terkait lanskap ancaman GenAI. Penghargaan khusus disampaikan kepada komunitas OWASP Gen AI Security Project atas kerangka kerja dan sumber daya terbuka yang menjadi fondasi analisis dalam penelitian ini.

REFERENSI

APJII. (2024). *Survei penetrasi internet Indonesia 2024*. Asosiasi Penyelenggara Jasa Internet Indonesia.

Branch, H. J., Cefalu, J. R., McHugh, J., Hujer, L., Bahl, A., del Castillo Iglesias, D., … Darwishi, R. (2022). Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples. *arXiv preprint* arXiv:2209.02128.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc. https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., & Wong, E. (2023). Jailbreaking black box large language models in twenty queries. *arXiv preprint* arXiv:2310.08419.

Chen, S., Liu, Y., He, Y., Li, B., & Song, D. (2024). Generative AI security: Challenges and countermeasures. *arXiv preprint* arXiv:2402.12617.

Creswell, J. W., & Creswell, J. D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). SAGE Publications.

Das, B. C., Amini, M. H., & Wu, Y. (2025). Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, *57*(6), Article 152, 1–39. https://doi.org/10.1145/3712001

Forum of Incident Response and Security Teams [FIRST]. (2023). *Common vulnerability scoring system version 4.0*. https://www.first.org/cvss/v4.0/specification-document

Gartner. (2024, January 17). *Gartner predicts 80% of enterprises will have used GenAI APIs or deployed GenAI-enabled applications by 2026*. Gartner Research. https://www.gartner.com/en/newsroom/press-releases/2024-01-17-gartner-predicts-80-percent-of-enterprises-will-have-used-generative-ai-apis-or-deployed-generative-ai-enabled-applications-by-2026

GitGuardian. (2026, April 14). 29 million leaked secrets in 2025: Why AI agents credentials are out of control. *Help Net Security*. https://www.helpnetsecurity.com/2026/04/14/gitguardian-ai-agents-credentials-leak/

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt

injection. In *Proceedings of the Black Hat USA Conference*. https://www.blackhat.com/us-23/

Gupta, M., Akiri, C., Aryal, K., Parker, E., & Praharaj, L. (2023). Privacy risks of large language models: A comprehensive survey. *arXiv preprint* arXiv:2310.12825.

Institute for Security and Open Methodologies [ISECOM]. (2010). *Open source security testing methodology manual* (Version 3.02). https://www.isecom.org/OSSTMM.3.02.pdf

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (EBSE Technical Report EBSE-2007-01). Keele University and Durham University. https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

Liu, Y., Deng, G., Li, Y., Wang, K., Wang, Z., Wang, X., … Liu, Y. (2023). Prompt injection attack against LLM-integrated applications. *arXiv preprint* arXiv:2306.05499.

Liu, Y., Jia, Y., Geng, R., Jia, J., & Gong, N. Z. (2024). Formalizing and benchmarking prompt injection attacks and defenses. In *Proceedings of the 33rd USENIX Security Symposium* (pp. 1831–1847). USENIX Association. https://www.usenix.org/conference/usenixsecurity24/presentation/liu-yupei

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., … Lee, K. (2023). Scalable extraction of training data from (production) language models. *arXiv preprint* arXiv:2311.17035.

OpenAI. (2025). OpenAI API documentation: Security best practices. *OpenAI Platform*. https://platform.openai.com/docs/guides/safety-best-practices

OWASP Foundation. (2021). *OWASP Top 10:2021*. https://owasp.org/www-project-top-ten/

OWASP Foundation. (2024). *OWASP testing guide* (Version 4.2). https://owasp.org/www-project-web-security-testing-guide/

OWASP Foundation. (2025). *OWASP Top 10 for large language model applications*. https://owasp.org/www-project-top-10-for-large-language-model-applications/

OWASP Gen AI Security Project. (2025). *OWASP Top 10 for LLM applications 2025*. https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/

Penetration Testing Execution Standard [PTES]. (2014). *Penetration testing execution standard*. http://www.pentest-standard.org/

Republik Indonesia. (2022). *Undang-Undang Republik Indonesia Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi*. Lembaran Negara RI Tahun 2022 Nomor 136. Sekretariat Negara.

Truffle Security. (2025). Research finds 12,000 ‘live’ API keys and passwords in DeepSeek’s training data. *Truffle Security Blog*. https://trufflesecurity.com/blog/research-finds-12-000-live-api-keys-and-passwords-in-deepseek-s-training-data

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998–6008). Curran Associates, Inc. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Wu, R., Li, J., Chen, X., Zhang, Y., & Wang, S. (2023). Data leakage in large language models: Analysis and mitigation. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery. https://doi.org/10.1145/3576915.3623178

Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, *4*(2), 100211. https://doi.org/10.1016/j.hcc.2024.100211