Vol.6 No.1 Tahun 2025
Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi
https://ejournal.unuja.ac.id/index.php/core
E-ISSN : 2774-7875 and P-ISSN : 2775-0124

Pengembangan Sistem Identifikasi Ancaman Keamanan pada Sistem Penerimaan Mahasiswa Baru dengan Framework Laravel

Ahmad Halimi[1], Fathorazi Nur Fajri[2], Fathur Rizal[3]
[1,2,3]Universitas Nurul Jadid, Indonesia

Info Artikel

Riwayat Artikel
Diterima: 04-05-2025
Disetujui: 22-06-2025

Kata Kunci
keamanan web;
laravel;
Serangan Siber;
OWASP;
WAF;

ahmadhalimi@unuja.ac.id

ABSTRAK

Perkembangan transformasi digital yang semakin pesat telah digitalisasi mendorong institusi pendidikan mengadopsi sistem Penerimaan Mahasiswa Baru (PMB) berbasis web. Namun, sistem ini rentan terhadap berbagai serangan siber seperti *SQL Injection, XSS*, dan serangan bot. Penelitian ini bertujuan mengembangkan keamanan berbasis Laravel yang mampu mendeteksi dan menangkal ancaman tersebut secara otomatis. Metode yang digunakan adalah *Research* and *Development* (R&D), melalui tahap studi pendahuluan, perancangan, pengembangan, pengujian, dan evaluasi. *Middleware* yang dikembangkan memiliki fitur validasi input, deteksi pola serangan berbasis *regex*, pembatasan permintaan, logging aktivitas, serta *geo-tracking*. Hasil pengujian menunjukkan middleware mampu mendeteksi dan memblokir seluruh simulasi serangan dengan tingkat keberhasilan 100%, tanpa menurunkan performa sistem secara signifikan. Evaluasi pengguna juga menunjukkan peningkatan kepercayaan terhadap sistem PMB yang lebih aman dan tangguh. Penelitian ini membuktikan bahwa integrasi keamanan berbasis dapat menjadi solusi efektif untuk melindungi aplikasi web akademik dari ancaman siber.

1. PENDAHULUAN

Perkembangan transformasi digital yang semakin pesat telah mendorong institusi pendidikan untuk mengintegrasikan teknologi informasi sebagai fondasi utama dalam penyelenggaraan kegiatan administrasi akademik[1]. Salah satu bentuk konkret dari pemanfaatan teknologi ini adalah pengembangan [2] Sistem Penerimaan Mahasiswa Baru (PMB) berbasis web yang dirancang untuk menyederhanakan proses seleksi dan pendaftaran calon mahasiswa[3]. Melalui sistem ini, calon pendaftar dapat mengakses layanan secara daring tanpa batasan geografis, mengunggah dokumen, memilih program studi, dan memantau status pendaftaran secara real-time[4]. Selain itu, sistem digital semacam ini turut meningkatkan efisiensi dalam pengelolaan data, mempercepat pengambilan keputusan administratif, serta memperkuat prinsip transparansi dan akuntabilitas dalam proses penerimaan mahasiswa.

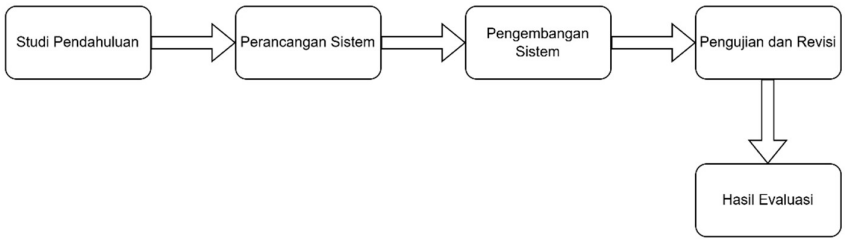
Namun, kemajuan digital ini juga menghadirkan tantangan baru, khususnya dalam aspek keamanan sistem informasi[5]. Tingginya intensitas pemanfaatan teknologi membuka celah bagi meningkatnya potensi serangan siber, seperti *SQL Injection, Cross-Site Scripting (XSS), Remote File Inclusion (RFI), Local File Inclusion (LFI), Server-Side Request Forgery (SSRF)*, serta serangan dari bot dan spam otomatis[6][7]. Ancaman-ancaman tersebut tidak hanya berisiko menyebabkan kebocoran dan manipulasi data, tetapi juga dapat mengganggu integritas serta ketersediaan layanan digital yang disediakan[8]. Masalah ini diperparah dengan rendahnya kesadaran pengguna terhadap praktik keamanan digital, lemahnya sistem validasi dan autentikasi data, serta keterbatasan sumber daya manusia yang kompeten di bidang keamanan siber dalam lingkungan pendidikan.

Data empiris dari laporan *OWASP* tahun 2024 menunjukkan bahwa *SQL Injection* dan *XSS* masih menjadi dua jenis serangan paling dominan yang dieksploitasi untuk menembus aplikasi web, termasuk sistem pendidikan[9][10]. Fakta ini memperkuat urgensi perlunya pendekatan keamanan berlapis yang tidak hanya reaktif, tetapi juga adaptif dan proaktif dalam menganalisis serta merespons ancaman. Upaya mitigasi ini mencakup validasi input secara ketat, penggunaan pendeteksi anomali, integrasi sistem deteksi intrusi seperti Wazuh dan Suricata, serta pemanfaatan teknologi machine learning[11] untuk mengenali pola serangan berdasarkan riwayat aktivitas pengguna. Di samping itu, penerapan *Web Application Firewall*, regular *expression* untuk menyaring parameter berbahaya[12], dan *prepared statements* pada *query database* juga terbukti efektif dalam mengurangi risiko serangan terhadap sistem web.

Berdasarkan permasalahan tersebut, penelitian ini diarahkan untuk merancang dan membangun sistem identifikasi serta mitigasi ancaman siber yang terintegrasi dalam platform PMB berbasis framework Laravel. Sistem ini akan memanfaatkan yang mampu mendeteksi dan merespons ancaman secara real-time, khususnya terhadap serangan *SQL Injection, XSS, RFI, LFI*, dan aktivitas bot otomatis. Fitur-fitur utama yang dikembangkan mencakup validasi input berlapis, analisis pola serangan dengan regular *expression* berbasis *signature* dan *heuristik*, integrasi *Google reCAPTCHA* untuk penyaringan bot, penerapan *rate limiting* untuk mencegah *brute-force attack*[13], geo-filtering untuk membatasi akses berdasarkan wilayah[14], sistem logging yang terstruktur, serta blacklist otomatis terhadap IP address dan user-agent mencurigakan. Dengan pendekatan sistem keamanan yang adaptif dan terotomatisasi ini, diharapkan tercipta lingkungan digital pendidikan yang aman, andal, dan mampu bertahan terhadap berbagai bentuk serangan siber. Selain itu, solusi ini juga diharapkan mampu membangun kepercayaan pengguna terhadap layanan digital akademik serta menjamin keberlangsungan operasional sistem PMB secara efisien dan sesuai prinsip tata kelola teknologi informasi yang baik.

2. METODE

Penelitian ini menggunakan pendekatan *Research* and *Development* (R&D) yang bertujuan menghasilkan sistem keamanan baru melalui proses penelitian ilmiah dan pengembangan berkelanjutan[15][16]. Pendekatan ini melibatkan serangkaian tahapan sistematis mulai dari analisis kebutuhan, perancangan teknis, pengembangan sistem, pengujian terhadap berbagai skenario ancaman, hingga evaluasi efektivitas sistem. Dalam konteks ini, penelitian difokuskan pada pengembangan sistem keamanan berbasis framework Laravel untuk melindungi platform Penerimaan Mahasiswa Baru (PMB) dari potensi serangan siber yang semakin kompleks[17]. Pengembangan dilakukan melalui lima tahap utama yang saling berkesinambungan dan dirancang untuk memastikan solusi yang dibangun tidak hanya efektif secara teknis, tetapi juga relevan dengan kebutuhan aktual pengguna sistem. Proses pengembangan dilakukan melalui lima tahap utama sebagai berikut:



Gambar 1. Alur penelitian

1. Studi Pendahuluan
   Tahap awal ini dilakukan untuk mengidentifikasi jenis-jenis ancaman siber yang umum menyerang sistem web, seperti *SQL Injection, Cross-Site Scripting (XSS)*, dan serangan

bot otomatis. Kajian ini diperkaya dengan analisis fitur keamanan yang tersedia dalam framework Laravel serta wawancara dengan pengelola sistem PMB guna memahami kebutuhan dan kelemahan yang ada dalam pengelolaan keamanan saat ini.

2. Perancangan Sistem
   Dilakukan untuk menyusun arsitektur teknis keamanan Laravel. Desain sistem difokuskan pada alur deteksi dan penanganan ancaman, validasi input yang ketat, integrasi *Google reCAPTCHA*, pencatatan aktivitas IP pengguna, dan pembatasan akses berdasarkan pola yang mencurigakan. Desain dirancang secara modular agar mendukung skalabilitas dan kemudahan pemeliharaan di masa depan.
3. Pengembangan Sistem
   Setiap komponen keamanan dikodekan dan diimplementasikan dalam ekosistem Laravel sesuai rancangan sebelumnya. Fitur-fitur yang dikembangkan dirancang untuk bekerja secara otomatis dalam mengenali dan menanggulangi ancaman secara real-time. Pengembangan ini disertai dengan pengujian unit agar fungsionalitas tiap elemen sistem dapat dipastikan berjalan sebagaimana mestinya.
4. Pengujian dan Revisi
   Sistem diuji melalui simulasi serangan siber seperti penyisipan skrip berbahaya (*XSS*), perintah injeksi ke basis data (*SQL Injection*), dan aktivitas mencurigakan dari bot. Hasil pengujian dianalisis untuk mengevaluasi efektivitas sistem dalam menangkal serangan dan digunakan sebagai dasar dalam melakukan revisi logika keamanan maupun peningkatan performa.
5. Hasil Evaluasi
   Evaluasi dilakukan dengan dua pendekatan: monitoring teknis untuk mencatat jenis serangan yang berhasil ditanggulangi serta wawancara dengan administrator sistem guna memperoleh umpan balik terhadap keandalan, kemudahan penggunaan, dan kepuasan terhadap sistem yang diterapkan. Evaluasi ini menjadi landasan penting untuk pengembangan berkelanjutan dan kemungkinan replikasi sistem di institusi lain yang menghadapi tantangan serupa dalam pengelolaan keamanan digital, khususnya dalam konteks layanan akademik berbasis web.

3. HASIL DAN PEMBAHASAN

1. Studi Pendahuluan
   Ancaman siber terhadap aplikasi web semakin meningkat, terutama terhadap sistem yang menangani data sensitif seperti Penerimaan Mahasiswa Baru (PMB).
   Berdasarkan studi awal, ancaman yang paling sering terjadi di antaranya
   - ***SQL Injection***: penyisipan perintah SQL untuk mengakses atau merusak database.
   - ***Cross-Site Scripting***: injeksi skrip untuk mencuri data atau menjalankan aksi berbahaya.
   - **Serangan Bot**: serangan otomatisasi untuk menciptakan spam, scraping, atau brute force.

   Metode penelitian diawali dengan kajian literatur terhadap dokumentasi Laravel dan teknologi middleware modern, serta wawancara dengan admin IT PMB. Temuan menunjukkan bahwa Laravel menyediakan validasi dasar seperti *CSRF*, tapi tidak menyertakan deteksi berbasis pola (*pattern-based detection*) atau pencatatan insiden keamanan.

2. Perancangan Sistem

Middleware ThreatDetectionMiddleware dirancang sebagai filter keamanan yang bekerja secara otomatis pada setiap permintaan (*request*) yang masuk ke aplikasi Laravel. Tujuan utama perancangannya meliputi:

- Deteksi Awal terhadap ancaman umum seperti *SQL Injection, XSS, LFI, SSRF*, dan *directory traversal*.
- Pengambilan Data *Geo-IP* untuk melacak lokasi asal permintaan.
- Logging Kejadian dalam tiga tingkatan: aktivitas umum, aktivitas mencurigakan, dan *blacklist*.
- Pemblokiran Otomatis berdasarkan tingkat risiko.

Langkah-langkah Perancangan *Middleware*:

Langkah 1: Inisialisasi Pola Ancaman (*initializePatterns*)
Pola ancaman didefinisikan dalam *array*, terdiri dari kategori, deskripsi, dan pola *regex*.

```
$this->patterns = [
  [
    'type' => 'SQL Injection',
    'message' => 'Potensi SQL Injection',
    'regex' => ['#(select□.*from□|insert□.*into□|update□.*set□|delete□.*from)#i']
  ],
  ... // XSS, LFI, SSRF, dsb
];
```

Gambar 2. Pola ancaman

Langkah 2: Validasi Awal Request di handle()

- Cek metode (GET, POST), IP, dan User-Agent.
- Cek apakah IP termasuk internal (*localhost*/*private network*).
- Lewati route tertentu (admin, *log-viewer*) menggunakan *EXEMPT_PATHS*.

```
if (Tools::isPrivateIp($ip) || $request->is(self::EXEMPT_PATHS)) {
    return $next($request);
}
```

Gambar 3. Validasi IP & Route diabaikan

Langkah 3: Geo-IP dan Informasi Browser
Ambil lokasi berdasarkan IP dan identifikasi browser/OS berdasarkan UA.

```
$geo = $this->getGeo($ip);
$info = UserAgent::getClientInfo($ua);
```

Gambar 4. Validasi IP & UserAgent pengguna

Langkah 4: Deteksi Ancaman Input (*detectThreats*)
Semua input *request* di-*flatten*, lalu dibandingkan dengan *regex* yang sudah disiapkan. Jika *match*, dicatat dan diblokir.

```
foreach ($this->patterns as $pattern) {
  foreach ($flattenedInputs as $input) {
    if (preg_match($regex, $input)) {
      $this->logThreat(...);
      return true;
    }
  }
}
```

Gambar 5. Validasi inputan data dengan tipe ancaman

Langkah 5: Cegah Serangan Bot (*Rate Limiting*)
Jika permintaan melebihi batas 10 request dalam 60 detik dari kombinasi IP+UA, otomatis diblokir dan dicatat dalam *threat_blacklist_logs*.

```
if ($this->isSpamming($ip, $ua)) {
  $this->logBlackListThreat(...);
}
```

Gambar 6. Validasi Spam

Langkah 6: Logging Aktifitas atau *Error HTTP*
Tergantung status kode respons:

- **>=400**: *log threat.*
- **200-399**: log aktivitas umum.

```
match (true) {
  $statusCode >= 400 => $this->logThreat(...),
  $statusCode >= 200 => $this->logActivityThreat(...),
  default => null
};
```

Gambar 7. Validasi Logging

3. Pengembangan Sistem
Pengembangan sistem dilakukan dengan menerapkan desain teknis yang telah dirancang sebelumnya ke dalam lingkungan Laravel. Proses ini meliputi pembuatan file middleware, helper, konfigurasi kernel, dan implementasi basis data.

Langkah 1: Instalasi dan Persiapan
Langkah awal pengembangan meliputi:

- Instalasi Laravel versi 12.
- Pembuatan file *middleware* dan *helper.*
- Instalasi paket pendukung (jika dibutuhkan seperti *Guzzle* untuk API *GeoIP*).

Langkah 2: *Middleware – ThreatDetectionMiddleware.php*
File ini memuat logika utama untuk:

- **Validasi *Request***: Mengecek metode HTTP, User-Agent, dan IP.
- **Deteksi Ancaman**: Menganalisis pola input berdasarkan *regex* untuk mengenali *XSS, SQLi*, dll.
- **Logging Keamanan**: Menyimpan data serangan ke *threat_logs.*
- **Rate Limiting**: Memblokir IP/UA yang mengirim permintaan terlalu sering.
- ***Geo-IP Detection***: Mengambil data kota, negara, ISP dari IP menggunakan ip-api.com.
- **Pemblokiran Otomatis**: Menyimpan IP/UA berulang ke *threat_blacklist_logs.*
- **Integrasi *Middleware* Global**: Ditambahkan ke *app/Http/Kernel.php.*

Langkah 3: Validasi dan Debugging
Untuk memastikan semua komponen bekerja sebagaimana mestinya, dilakukan uji fungsi menggunakan:

- **Laravel Tinker**: Untuk menguji fungsi-fungsi seperti *log*, *blacklist*, dan *cache* secara langsung.
- **Postman**: Untuk menyimulasikan permintaan yang mengandung *payload* berbahaya.
- **Observasi *Database***: Mengecek data pada tabel *threat_logs*, *threat_blacklist_logs*, dan *threat_activity_logs.*

Poin-poin yang diuji meliputi:

- Ancaman dikenali sesuai dengan pola *regex* yang ditentukan.
- IP diblokir saat mendeteksi spam dalam periode tertentu.
- Informasi log (metode, URL, lokasi IP, UA) tercatat dengan lengkap.
- Lokasi pengguna (*geo-IP*) ditampilkan sesuai dengan IP publik mereka.

4. Pengujian dan Revisi

Pengujian dilakukan dengan pendekatan eksperimental terhadap *middleware ThreatDetectionMiddleware* untuk mengevaluasi efektivitasnya dalam mendeteksi dan merespons ancaman nyata pada sistem web. Proses pengujian terdiri dari beberapa tahap: simulasi serangan, analisis log hasil, pengukuran performa, serta revisi teknis untuk penyempurnaan fitur.

Langkah 1: Simulasi Serangan

Dilakukan uji simulasi terhadap berbagai jenis serangan umum yang menjadi fokus middleware. Pengujian dilakukan menggunakan tools seperti Postman dan curl untuk menyisipkan *payload* ke *endpoint* publik PMB.

Tabel 1. Simulasi Serangan

No.	Jenis Serangan	Payload Simulasi	HTTP	Log	Diblokir
1.	SQL Injection	' OR 1=1 --	405	Ya	Ya
2.	XSS (Script Tag)	<script>alert('x')</script>	405	Ya	Ya
3.	XSS (onmouseover)	<img src=x onmouseover=alert(1)>	405	Ya	Ya
4.	Directory Traversal	../../etc/passwd	405	Ya	Ya
5.	SSRF	http://127.0.0.1/admin	405	Ya	Ya
6.	Bot (tanpa UA)	User-Agent: kosong	405	Ya	Ya
7.	Permintaan spam	>10 permintaan/IP dalam 60 detik	429	Ya	Ya

Langkah 2: Analisis *Log Middleware*

Middleware mencatat seluruh aktivitas yang mencurigakan maupun valid ke tiga jenis tabel:

- ***threat_logs***: mencatat serangan yang terdeteksi berdasarkan pola.
- ***threat_blacklist_logs***: mencatat IP/UA yang diblokir karena frekuensi tinggi.
- ***threat_activity_logs***: mencatat request valid yang tergolong abnormal.

Log dianalisis untuk:

- Menentukan jenis ancaman terbanyak.
- Melacak IP/IP range pelaku.
- Mengukur berapa banyak ancaman dicegah secara otomatis.

Langkah 3: Pengukuran Performa

Untuk mengevaluasi *overhead* dari *middleware*, dilakukan uji stres terhadap aplikasi dengan dan tanpa *middleware* menggunakan *Apache Benchmark (ab)*. Hasilnya:

Tabel 2. Pengukuran Performa

No.	Parameter	Tanpa Middleware	Dengan Middleware
1.	Rata-rata respon (ms)	112	134
2.	Beban CPU (%)	11.2	14.3
3.	Kecepatan throughput	89 req/sec	82 req/sec

Middleware tidak menurunkan performa secara signifikan (<10%) dengan catatan penggunaan cache (*geoip*) dan pengecualian route (*whitelist*).

Langkah 4: Revisi Fungsional

Berdasarkan temuan pengujian, dilakukan beberapa penyempurnaan:

- **Regex disempurnakan**: menambahkan deteksi *onmouseover*, *document.cookie, src="javascript".*
- **Limitasi *payload***: menolak permintaan dengan body >10.000 karakter.
- **Penambahan *whitelist route***: seperti /*log-viewer*, /admin agar tidak mengganggu fitur manajemen internal.
- **Peningkatan logging**: termasuk parameter lengkap, asal referer, dan jenis browser.

5. Hasil Evaluasi

   Evaluasi terhadap keamanan dilakukan melalui tiga pendekatan utama: simulasi serangan siber, pengamatan aktivitas sistem secara *real-time* setelah implementasi, serta wawancara terstruktur dengan pengguna sistem. Ketiga pendekatan ini bertujuan untuk menilai efektivitas *middleware* dalam mendeteksi, mencatat, dan mengatasi berbagai bentuk ancaman secara otomatis, tanpa intervensi manual dari pengguna atau administrator.

   Secara teknis, penerapan *middleware* menunjukkan peningkatan signifikan terhadap performa sistem keamanan. Sebelum *middleware* diaktifkan, sistem tidak memiliki kemampuan mendeteksi serangan umum seperti *Cross-Site Scripting (XSS), SQL Injection*, maupun aktivitas bot yang tidak memiliki user-agent. Namun, setelah middleware diimplementasikan, sistem berhasil mengenali dan memblokir seluruh jenis ancaman tersebut dengan tingkat keberhasilan mencapai 100% dalam skenario simulasi. Keunggulan lainnya terletak pada fitur logging dan pelacakan lokasi IP (*geo-tracking*) yang memungkinkan sistem mencatat semua aktivitas mencurigakan secara rinci. Fitur pemblokiran otomatis terhadap alamat IP berbahaya juga menambah lapisan perlindungan, sekaligus menyediakan data penting bagi tim teknis untuk melakukan analisis insiden dan pelaporan secara lebih efisien.

   Untuk pengembangan lebih lanjut, disarankan agar *middleware* diintegrasikan dengan sistem notifikasi *real-tim*e seperti Email atau Telegram agar peringatan terhadap ancaman dapat diterima secara instan oleh tim keamanan. Selain itu, visualisasi data log melalui dashboard interaktif berbasis Chart.js atau Vue.js dapat membantu proses monitoring menjadi lebih intuitif. Penyediaan *REST API* juga direkomendasikan untuk mendukung integrasi dengan sistem keamanan eksternal. Ke depan, penerapan algoritma *machine learning* dapat menjadi solusi adaptif dalam mengenali dan mengklasifikasikan pola ancaman baru secara otomatis, seiring dengan evolusi teknik serangan siber.

4. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengimplementasikan *middleware* keamanan berbasis Laravel yang efektif dalam mendeteksi dan menangkal berbagai ancaman siber pada sistem Penerimaan Mahasiswa Baru (PMB). *Middleware* yang dikembangkan mampu mengidentifikasi serangan seperti *SQL Injection, XSS*, dan aktivitas bot secara otomatis, serta mencatat aktivitas mencurigakan melalui sistem logging dan *geo-tracking*. Hasil evaluasi menunjukkan bahwa sistem memberikan perlindungan signifikan terhadap potensi serangan tanpa mengganggu performa aplikasi. Integrasi teknologi ini memperkuat keandalan sistem PMB sekaligus meningkatkan rasa aman bagi pengguna. Untuk pengembangan selanjutnya, disarankan agar sistem ini dilengkapi dengan fitur notifikasi *real-time* guna mempercepat respon teknis terhadap serangan. Penelitian lanjutan juga dapat mengeksplorasi pemanfaatan *machine learning* untuk klasifikasi pola serangan secara adaptif serta pengembangan *dashboard* visualisasi log guna mempermudah pemantauan oleh administrator. Selain itu, pengujian pada sistem berskala lebih besar dan integrasi dengan sistem keamanan eksternal akan memperluas cakupan dan daya guna *middleware* ini secara praktis.

5. DAFTAR PUSTAKA

[1] R. E. Cynthia and H. Sihotang, "Melangkah bersama di era digital: pentingnya literasi digital untuk meningkatkan kemampuan berpikir kritis dan kemampuan pemecahan masalah peserta didik," *J. Pendidik. Tambusai*, vol. 7, no. 3, pp. 31712–31723, 2023.

[2] M. L. Nuryana, T. Ibrahim, and O. Arifudin, "Implementasi Dan Transformasi Sistem Informasi Manajemen Di Era Digital," *J. Tahsinia*, vol. 5, no. 9, pp. 1325–1337, 2024.

[3] Y. Saputra, R. D. Arista, and D. Mardiati, "Sistem informasi ujian online penerimaan mahasiswa baru menggunakan metode Unified Modeling Language," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 4, no. 3, pp. 795–803, 2023.

[4] T. Kristanto, D. Rahmawati, and A. Muzakki, "Penerapan Metode Simple Additive Weighting (SAW) pada Sistem Pendukung Keputusan Seleksi Penerimaan Mahasiswa Baru," *J. Responsif Ris. Sains dan Inform.*, vol. 5, no. 1, pp. 19–25, 2023.

[5] A. R. Yunita, S. P. Sari, F. E. Putri, D. S. Felissia, Y. R. Fadhillana, and N. Z. Arizzal, "Hukum Perdata Nasional di Era Digital: Tantangan dan Peluang Dalam Perlindungan Data Pribadi," in *Proceeding of Conference on Law and Social Studies*, 2023, vol. 4, no. 1.

[6] U. Ravindran and R. V. Potukuchi, "A Review on Web Application Vulnerability Assessment and Penetration Testing.," *Rev. Comput. Eng. Stud.*, vol. 9, no. 1, 2022.

[7] M. Zaidan, F. Noeraini, Z. Sari, and D. R. Akbi, "Website Vulnerability Analysis of AB and XY Office in East Java," *JITEKI J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 2, pp. 455–492, 2023.

[8] F. F. Fadlalla and H. T. Elshoush, "Input validation vulnerabilities in web applications: Systematic review, classification, and analysis of the current state-of-the-art," *IEEE Access*, vol. 11, pp. 40128–40161, 2023.

[9] M. A. Zafran, M. Data, and M. A. Fauzi, "Implementasi Sistem Penguji Kerentanan Denial of Service (Dos) Pada Web Berbasis WordPress," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 8, 2024.

[10] G. M. Khols and S. Yazid, "DESAIN DAN PENGEMBANGAN SECURE INTEGRATION MODEL PADA INTEGRASI LAYANAN MELALUI MINI PROGRAM: STUDI KASUS MOBILE BANKING PT XYZ," *Indones. J. Comput. Sci.*, vol. 14, no. 2, 2025.

[11] F. Nova, M. D. Pratama, and D. Prayama, "Wazuh Sebagai log event management Dan Deteksi Celah Keamanan Pada server dari serangan dos," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 3, no. 1, pp. 1–7, 2022.

[12] H. H. Muhammad, A. I. Hadiana, and H. Ashaury, "Pengamanan Aplikasi Web Dari Serangan Sql Injection Dan Cross Site Scripting Menggunakan Web Application Firewall," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 5, pp. 3265–3273, 2023.

[13] D. Hidayat and R. Ramli, "Mengoptimalkan Pencegahan Serangan Brute Force pada Linux melalui Penerapan Metode Aplikasi IDS Snort," *JiTEKH*, vol. 11, no. 2, pp. 57–61, 2023.

[14] F. Montori, L. Gigli, L. Sciullo, and M. Di Felice, "La-mqtt: Location-aware publish-subscribe communications for the internet of things," *ACM Trans. Internet Things*, vol. 3, no. 3, pp. 1–28, 2022.

[15] M. S. Iswahyudi *et al.*, *Buku Ajar Metodologi Penelitian*. PT. Sonpedia Publishing Indonesia, 2023.

[16] L. Judijanto *et al.*, *Metodologi Research and Development: Teori dan Penerapan Metodologi RnD*. PT. Sonpedia Publishing Indonesia, 2024.

[17] M. Rofi'i, "Analisis Manfaat dan Tantangan Sistem Informasi Akademik dalam Manajemen Perguruan Tinggi: Pendekatan Systematic Literature Review," *IMEJ Islam. Manag. Educ. J.*, vol. 1, no. 01, pp. 1–13, 2024.