SAIN TEKOM
ISSN 2088-1770 (Print)
ISSN 2503-3247 (Online)
Vol. 15 No.2 (2025) 141-152

JURNAL
SAINTEKOM
Sains, Teknologi, Komputer, dan Manajemen

Penerapan Laravel untuk Mengatasi Kelemahan Keamanan WordPress pada Website Badan Layanan Umum Daerah

Tegar Satria Iman Saputra[1], Suraya[2], *Muhammad Sholeh[3]
[1,3)]Informatika, Universitas AKPRIND Indonesia
[2)]Rekayasa Sistem Komputer, Universitas AKPRIND Indonesia
Jl. Kalisahak No.28, Kota Yogyakarta, Daerah Istimewa Yogyakarta
Email: [1]imaniashiren6@gmail.com, [2] suraya@akprind.ac.id,[3]muhash@akprind.ac.id

ABSTRACT

The BLUD.co.id website was previously developed using the Content Management System (CMS) WordPress*, but it encountered security issues in the form of* URL redirect *attacks that directed users to untrusted sites. This study aims to redesign the information system using the Laravel* framework *to enhance security and flexibility. The system was developed locally by applying the* Model-View-Controller (MVC) *architectural pattern and a relational database structure designed independently. The research employed a* Research and Development (R&D) *approach through the stages of design, implementation, and security testing. Tests were conducted on several scenarios, including* SQL Injection*,* Cross-Site Request Forgery (CSRF)*, and login bypass. The results indicate that the Laravel-based system was able to reduce the risk of attacks by 80–82% compared to* WordPress*, while also demonstrating greater stability when handling invalid inputs. These findings highlight that Laravel can provide a significant contribution to delivering a more secure digital information platform for government services, while also opening opportunities for further research to evaluate its application in similar organizational contexts.*

Keywords : laravel; information system; web security; WordPress; R&D

ABSTRAK

Situs web BLUD.co.id sebelumnya dibangun dengan *Content Management System (CMS) WordPress*, tetapi menghadapi masalah keamanan berupa serangan *URL redirect* yang mengarahkan pengguna ke situs tidak terpercaya. Penelitian ini bertujuan merancang ulang sistem informasi menggunakan *framework Laravel* untuk meningkatkan aspek keamanan dan fleksibilitas. Sistem dikembangkan secara lokal dengan menerapkan pola arsitektur *Model-View-Controller (MVC)* serta basis data relasional yang dirancang mandiri. Metode penelitian mengikuti pendekatan *Research and Development (R&D)* melalui tahapan perancangan, implementasi, dan pengujian keamanan. Uji coba dilakukan pada beberapa skenario, termasuk *SQL Injection*, *Cross-Site Request Forgery (CSRF)*, dan bypass login. Hasil penelitian menunjukkan bahwa sistem berbasis *Laravel* mampu menurunkan risiko serangan hingga 80–82% dibandingkan *WordPress*, sekaligus menunjukkan kestabilan yang lebih baik ketika menerima input tidak valid. Temuan ini memperlihatkan bahwa *Laravel* dapat memberikan kontribusi nyata dalam penyediaan platform informasi digital yang lebih aman bagi layanan pemerintahan, sekaligus membuka peluang penelitian lanjutan untuk mengevaluasi penerapannya pada organisasi serupa.

Kata kunci : laravel; sistem informasi; keamanan web; *WordPress*; *R&D*

Diterima Redaksi: 19-07-2025 | Selesai Revisi: 7-09-2025 | Diterbitkan Online: 30-09-2025

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat mendorong banyak instansi pemerintah untuk memanfaatkan *Content Management System* (CMS) sebagai solusi praktis dalam pengelolaan konten web. Salah satu CMS paling populer adalah WordPress, yang dikenal karena antar mukanya yang ramah pengguna dan ribuan plugin pendukung. WordPress digunakan secara luas oleh berbagai organisasi (Anggraeni et al., 2025), termasuk Badan Layanan Umum Daerah (BLUD), untuk memfasilitasi pelayanan publik secara digital. Namun, di balik kemudahan penggunaannya, WordPress menyimpan tantangan serius dalam aspek keamanan. Banyak serangan siber, seperti *malicious URL redirection*, terjadi akibat kerentanan plugin pihak ketiga atau pengaturan default yang tidak optimal (Setya Putra & Santoso, 2025). Masalah ini telah dialami langsung oleh BLUD, di mana situs web yang dikelola sempat dialihkan secara otomatis ke halaman pihak ketiga yang berpotensi membahayakan data dan reputasi instansi.

Penelitian sebelumnya menunjukkan bahwa CMS sumber terbuka memiliki keterbatasan dalam mengakomodasi kebutuhan keamanan tingkat lanjut, terutama pada sistem informasi publik yang menangani data sensitif (Feri Setyawan & Agustin, 2024). Upaya penanganan melalui konfigurasi tambahan seringkali tidak cukup efektif, sehingga diperlukan pendekatan yang lebih mendasar. Beberapa *framework* seperti *CakePHP* dan *CodeIgniter* juga digunakan dalam pengembangan web, namun penelitian menunjukkan bahwa *Laravel* memiliki fleksibilitas serta dukungan keamanan bawaan yang lebih baik dibandingkan *framework* sejenis (Endra et al., 2021). Adapun *framework* berbasis *JavaScript* lebih banyak dipakai untuk aplikasi berskala besar dengan kebutuhan *real-time*, sehingga kurang sesuai dengan fokus pengembangan BLUD.co.id yang berbasis *PHP*. Dengan pertimbangan tersebut, *Laravel* dipilih karena menawarkan fitur keamanan yang lengkap, dokumentasi kuat, serta dukungan komunitas yang luas.

Laravel sebagai *framework* PHP modern hadir dengan fitur bawaan yang mendukung pengembangan sistem yang aman dan fleksibel, seperti perlindungan terhadap *Cross-Site Request Forgery* (CSRF), validasi input yang

DOI: https://doi.org/10.33020/saintekom.v15i2.938

ketat untuk mencegah serangan *SQL Injection*, dan manajemen otentikasi berbasis *middleware* (Abutaleb et al., 2021). Penelitian oleh (Luh Gede Pivin Suwirmayanti et al., 2023) juga membuktikan bahwa penggunaan *Laravel* pada sistem informasi akademik memberikan peningkatan signifikan dari sisi keamanan dan modularitas dibandingkan CMS WordPress.

Celah penelitian (*research gap*) yang teridentifikasi adalah masih terbatasnya kajian penerapan *Laravel* dalam konteks pengembangan sistem informasi di Indonesia. Sebagian besar layanan digital pemerintah masih menggunakan *CMS WordPress*, yang secara global menguasai lebih dari 40% *website* di dunia (Setya Putra & Santoso, 2025), tetapi rawan serangan keamanan. Penelitian terdahulu lebih banyak berfokus pada optimalisasi WordPress atau penerapan *Laravel* di bidang akademik, sementara kajian pada layanan pemerintah seperti BLUD.co.id masih jarang dilakukan. Oleh karena itu, penelitian ini diarahkan pada perancangan dan implementasi sistem berbasis *Laravel* dengan menekankan fitur keamanan bawaan seperti validasi input untuk mencegah *SQL Injection*, perlindungan *Cross-Site Request Forgery (CSRF)*, serta *middleware* untuk kontrol otentikasi. Kontribusi penelitian ini terletak pada integrasi fitur keamanan tersebut sesuai kebutuhan operasional BLUD.co.id dan evaluasi efektivitas penggunaan melalui pengujian terhadap serangan umum, termasuk *SQL Injection* dan *URL redirection*.

2. METODE

Penelitian ini menggunakan pendekatan *Research and Development (R&D)* untuk mengembangkan sistem informasi berbasis web yang aman dan fleksibel. Pendekatan ini dipilih karena memungkinkan peneliti untuk merancang, mengimplementasikan, dan mengevaluasi sistem secara sistematis hingga menghasilkan produk yang sesuai dengan kebutuhan pengguna (Iftitah & Nuryasin, 2022; Imtihan et al., 2022).

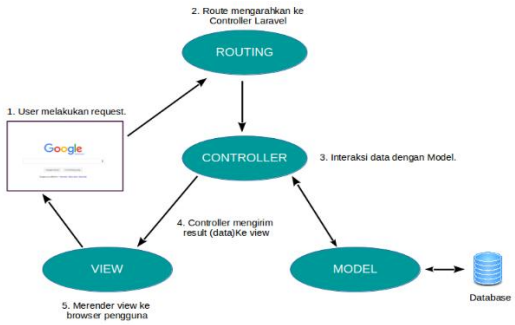
2.1. Model Arsitektur MVC

Pengembangan sistem dilakukan secara lokal menggunakan *framework Laravel* dengan menerapkan pola arsitektur *Model-View-Controller (MVC)*. Arsitektur ini memisahkan logika bisnis *(Model),* antarmuka pengguna *(View),* dan alur kontrol *(Controller),* sehingga meningkatkan

DOI: https://doi.org/10.33020/saintekom.v15i2.938

modularitas dan keamanan sistem. Basis data dirancang dengan pendekatan relasional menggunakan fitur *migration* dan *Eloquent ORM Laravel* untuk menjaga integritas data sekaligus mempermudah manipulasi data tanpa harus menulis query mentah (Muthia Kansha et al., 2023).

Alur pengembangan sistem berbasis MVC ditunjukkan pada Gambar 1.

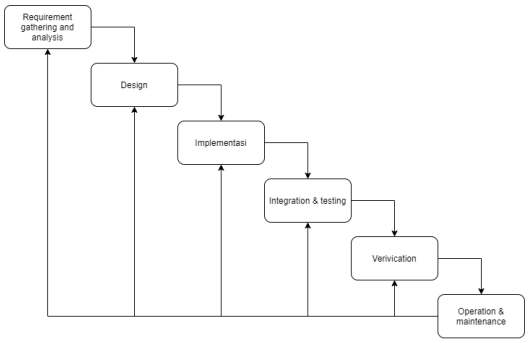


Gambar 1. Alur desain MVC BLUD
(Sumber : sulhi.id, 2021)

2.2. Model Pengembangan *Waterfall*

Tahapan pengembangan sistem mengikuti model *Waterfall* yang terdiri dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan (Abdul Wahid, 2020). Model ini dipilih karena sesuai untuk proyek dengan kebutuhan yang sudah jelas sejak awal, serta mendukung dokumentasi yang sistematis. Dibandingkan model lain seperti *Spiral, RAD,* atau *Prototype, Waterfall* lebih sederhana, terstruktur, dan meminimalisasi perubahan berulang yang berpotensi memperpanjang waktu pengembangan.

Alur pengembangan *Waterfall* ditunjukkan pada Gambar 2.



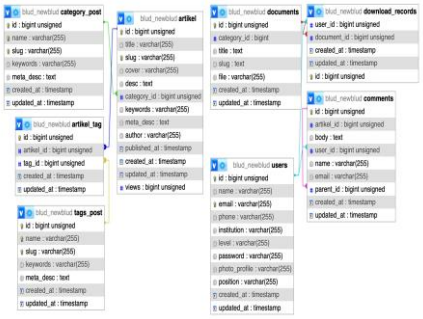
Gambar 2. Alur model *Waterfall*
(Sumber : dicoding, 2021)

2.3. Desain Basis Data

Perancangan basis data dilakukan untuk menggantikan struktur data WordPress yang bergantung pada plugin pihak ketiga, yang sering menjadi titik lemah keamanan. *Database* pada sistem baru dirancang secara relasional agar setiap entitas saling terhubung secara konsisten, mendukung integritas data, serta mempermudah proses audit. Fitur *migration Laravel* digunakan untuk mencatat perubahan skema, sedangkan *Eloquent ORM* dimanfaatkan untuk mengurangi risiko kesalahan sintaks maupun serangan *SQL Injection* karena pengembang tidak perlu menulis query mentah (Prastiawan et al., 2023).

Desain basis data sistem informasi BLUD ditampilkan pada Gambar 3.

DOI: https://doi.org/10.33020/saintekom.v15i2.938



Gambar 3. Desain basis data BLUD

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sistem informasi berbasis Laravel yang dikembangkan sebagai pengganti CMS WordPress pada BLUD. Sistem lama yang dibangun dengan WordPress menghadapi berbagai masalah keamanan, terutama serangan *malicious URL redirection* dan ketergantungan pada plugin pihak ketiga yang rentan dieksploitasi. Sistem baru berbasis *Laravel* dirancang untuk meningkatkan keamanan, modularitas arsitektur, serta kemudahan pengelolaan oleh administrator (Subiksa et al., 2023).

Laravel dipilih karena menyediakan fitur keamanan bawaan seperti proteksi *Cross-Site Request Forgery* (CSRF), validasi input untuk mencegah *SQL Injection*, serta *middleware* autentikasi pengguna. Dengan arsitektur *Model-View-Controller* (MVC), sistem ini lebih mudah dikembangkan sekaligus mendukung operasional digital BLUD secara lebih andal.

3.1. Tampilan Sistem

a. Halaman Login

Halaman login merupakan titik masuk utama ke dalam sistem informasi BLUD. Pada halaman ini, pengguna diminta untuk memasukkan email dan kata sandi sebagai kredensial otentikasi. Validasi kredensial dilakukan di sisi server untuk memastikan bahwa input yang diberikan sudah sesuai dengan standar keamanan yang ditetapkan. Selain itu, sistem juga dilengkapi dengan fitur keamanan tambahan berupa *Google reCAPTCHA* untuk memverifikasi bahwa permintaan login benar-benar berasal dari manusia dan bukan dari program otomatis (*bot*).

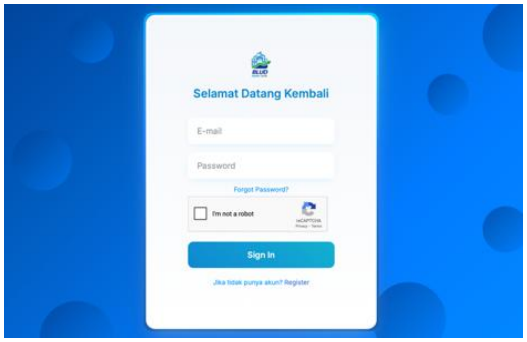
Implementasi *Laravel* pada halaman login ini memanfaatkan fitur *form request validation* untuk memastikan semua input telah memenuhi syarat sebelum diproses lebih lanjut oleh server. Fitur ini tidak hanya memvalidasi format data, tetapi juga mampu menolak input yang berpotensi berbahaya, seperti skrip berbahaya (*malicious scripts*). Dengan adanya validasi yang ketat di tahap awal, sistem dapat meminimalisir risiko serangan

DOI: https://doi.org/10.33020/saintekom.v15i2.938

injeksi data dan meningkatkan keandalan proses otentikasi.

Berbeda dengan sistem WordPress yang memiliki konfigurasi *login default* dan sering kali menjadi target serangan *brute-force, Laravel* memungkinkan penyesuaian rute *login* untuk meningkatkan keamanan. Misalnya, jalur *login* dapat disembunyikan atau diubah sehingga tidak mudah ditebak oleh penyerang. Selain itu, Laravel mendukung penerapan *middleware* otentikasi yang secara otomatis memfilter permintaan yang tidak sah. Pembatasan laju (rate limiting) juga diterapkan untuk mencegah percobaan login berulang-ulang dalam waktu singkat, yang biasanya dilakukan oleh penyerang dengan teknik brute-force.

Halaman login ini ditampilkan pada Gambar 3, yang menunjukkan antarmuka dengan fitur *reCAPTCHA* yang telah terintegrasi.



Gambar 1. Tampilan Halaman Login

Husain et al., (2024) menemukan bahwa penerapan validasi *input* dan *reCAPTCHA* terbukti mengurangi risiko serangan *login* otomatis (*automated login attacks*) hingga 75% pada sistem berbasis web. Dalam studi mereka, integrasi *reCAPTCHA* di tahap otentikasi dapat secara signifikan mempersulit *bot* untuk mengakses sistem, sekaligus melindungi server dari beban berlebih akibat permintaan palsu (*fake requests*). Hal ini sejalan dengan implementasi pada sistem BLUD, yang menunjukkan bahwa fitur keamanan modern seperti ini sangat krusial dalam menjaga kestabilan layanan dan mencegah kompromi terhadap akun pengguna.

b. Dashboard

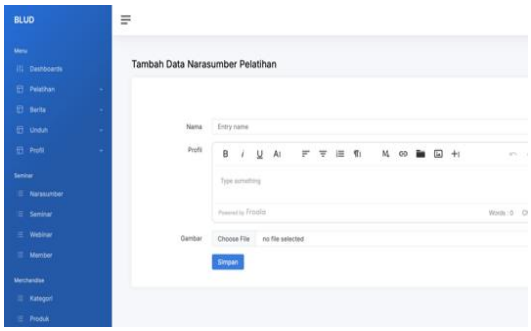
Dashboard pada sistem informasi BLUD dirancang sebagai pusat kontrol utama yang memfasilitasi admin dan operator dalam mengelola berbagai data penting, termasuk data narasumber pelatihan. Dashboard ini dibangun dengan fitur CRUD (*Create, Read, Update, Delete*) yang memungkinkan pengguna untuk melakukan penambahan, pembaruan, penghapusan, dan penelusuran data dengan cepat dan efisien. Setiap tindakan CRUD yang dilakukan pada dashboard dilengkapi dengan lapisan validasi input di sisi klien dan server untuk memastikan bahwa data

DOI: https://doi.org/10.33020/saintekom.v15i2.938

yang disimpan memenuhi standar integritas dan keamanan.

Penyimpanan data dalam sistem ini memanfaatkan model *Laravel* yang terhubung langsung dengan tabel-tabel pada basis data relasional. Integrasi ini memungkinkan proses pengolahan data berlangsung dengan lancar, meminimalisir terjadinya inkonsistensi, serta mengurangi risiko manipulasi data yang biasanya terjadi jika validasi hanya dilakukan di sisi klien. Laravel memanfaatkan *Eloquent ORM* untuk mengelola hubungan antar entitas, sehingga memudahkan pengembang dalam menangani data berskala besar.

Selain itu, dashboard ini memiliki antarmuka pengguna (*user interface*) yang dirancang dengan prinsip kesederhanaan dan kejelasan. Desain yang bersih dan terstruktur membuat admin/operator dapat berfokus pada tugas utama tanpa terganggu elemen visual yang tidak perlu. Seluruh kontrol pada dashboard diatur agar mudah diakses, sehingga mempercepat proses pengelolaan data.



Gambar 2. Tampilan Dashboard

Gambar 4 menampilkan tampilan dashboard pada sistem BLUD dengan tata letak yang intuitif dan dukungan keamanan yang telah diintegrasikan ke dalam setiap modul.

Azhar et al., (2023) menyebutkan bahwa penggunaan *framework Laravel* dengan validasi berlapis sangat efektif dalam mengurangi celah keamanan pada sistem manajemen konten. Dalam penelitian mereka, sistem yang dibangun menggunakan *Laravel* mampu memblokir upaya eksploitasi yang sering terjadi pada CMS konvensional seperti WordPress, terutama serangan injeksi data melalui *form input*. Keunggulan *Laravel* dalam memisahkan logika aplikasi dan lapisan data juga mempermudah pengelolaan keamanan tanpa mengorbankan performa aplikasi. Hal ini relevan dengan sistem BLUD yang menuntut perlindungan ekstra terhadap data sensitif dalam pelatihan.

DOI: https://doi.org/10.33020/saintekom.v15i2.938

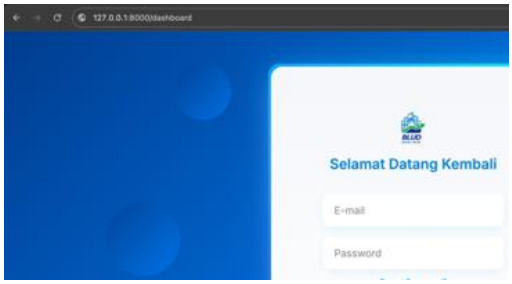
3.3. Uji Keamanan Sistem

Pengujian dilakukan pada tiga skenario utama: CSRF Token, SQL Injection, MiddlewareAuthentication.

a. Uji Pembatasan Akses (*Middleware Auth*)

Pengujian dilakukan dengan mencoba mengakses *URL dashboard* secara langsung melalui *http://127.0.0.1:8000/dashboard* tanpa *login*. Sistem secara otomatis menolak permintaan tersebut dan mengarahkan pengguna ke halaman *login*. Hasil ini menunjukkan bahwa middleware auth Laravel berhasil mencegah akses ilegal ke halaman backend. Fitur ini sangat penting untuk menangkal serangan umum seperti *URL tampering* dan *session hijacking*, di mana penyerang berusaha masuk hanya dengan menebak URL. Berbeda dengan WordPress yang memerlukan plugin tambahan, *Laravel* menyediakan fitur ini secara bawaan sehingga lebih konsisten dan aman.

Gambar 5 memperlihatkan proses pengalihan pengguna yang tidak memiliki hak akses ke halaman login sebagai bentuk perlindungan sistem.



Gambar 3. Uji coba pembatasan askes

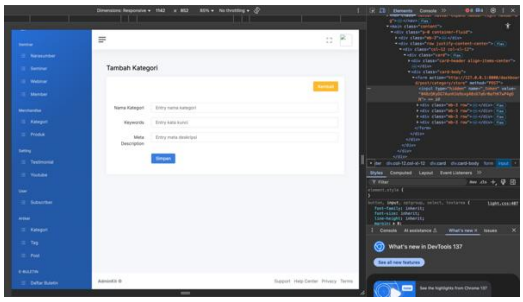
b. Uji Token CSRF

Pada pengujian form input (misalnya Tambah Kategori), Laravel otomatis menyisipkan *token* CSRF unik pada setiap form HTML. Token ini berfungsi sebagai identitas acak yang hanya dikenali oleh server, sehingga setiap permintaan palsu dari situs pihak ketiga langsung ditolak. Hasil inspeksi elemen form menunjukkan adanya atribut *name="_token"* dengan *string* acak unik yang berubah di setiap sesi. Mekanisme ini memastikan bahwa hanya permintaan sah yang dapat diproses, sekaligus menutup celah *Cross-Site Request Forgery*. Proteksi seperti ini tidak tersedia secara default di *WordPress* dan biasanya hanya bisa diperoleh lewat plugin tambahan.

Gambar 6 menunjukkan hasil inspeksi form yang memverifikasi keberadaan token CSRF, yang membuktikan bahwa proteksi *Laravel*

DOI: https://doi.org/10.33020/saintekom.v15i2.938

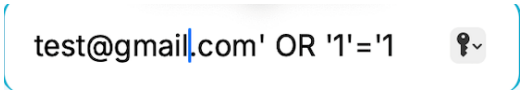
terhadap serangan ini telah bekerja dengan baik.



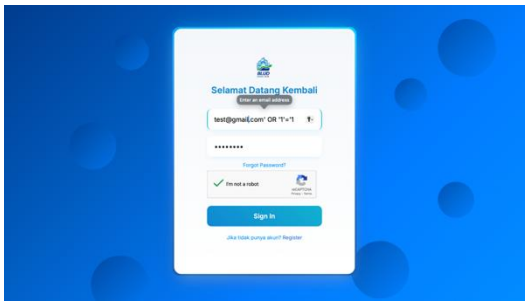
Gambar 4. Uji pengecekan token CSRF

c. **Uji SQL Injection**

Uji coba dengan *payload ' OR '1'='1* pada *form login* membuktikan bahwa sistem *Laravel* menolak input berbahaya sejak awal. Validasi format email di sisi klien dan server memastikan hanya input dengan pola benar yang diproses. Selain itu, *Laravel* secara *default* menggunakan *prepared statement* melalui *Eloquent ORM*, sehingga *query* berbahaya tidak pernah dieksekusi. Dengan lapisan perlindungan ini, serangan *SQL Injection* yang umum pada aplikasi berbasis form dapat dicegah sepenuhnya. Pada sistem *WordPress*, celah ini sering dimanfaatkan penyerang jika plugin keamanan tidak diperbarui secara rutin.



Gambar 5. Payload SQL Injection



Gambar 6. Uji coba keamanan SQL Injection

Gambar 7 memperlihatkan *payload SQL Injection* yang digunakan dalam pengujian, sedangkan Gambar 8 menunjukkan hasil uji coba yang menegaskan keberhasilan sistem dalam memblokir serangan tersebut.

Tabel 1. Skenario Uji Keamanan Sistem

Jenis Uji	WordPress (Sebelum)	Laravel (Sesudah)
Brute Force / Direct URL Access	Membutuhkan plugin tambahan, sering gagal mencegah akses ilegal	*Middleware auth* otomatis menolak akses ilegal tanpa login
CSRF Attack	Proteksi terbatas, hanya tersedia melalui plugin	Token CSRF otomatis disisipkan di semua form HTML
SQL Injection	Rentan jika validasi input lemah	*Prepared statement* + validasi input mencegah eksekusi payload

Tabel 2. Perbandingan Kinerja Sistem *Before After*

Indikator	Word Press (Sebelum)	Laravel (Sesudah)	Perubahan	Indikator

DOI: https://doi.org/10.33020/saintekom.v15i2.938

Keberhasilan Serangan	Tinggi, sering berhasil melalui plugin lemah	Rendah, 0% pada CSRF & SQL Injection	Turun ±80–82%	Keberhasilan Serangan
Error Rate	Sering muncul pada input tidak valid	Lebih rendah karena validasi berlapis	Turun signifikan	Error Rate
Response Time	1,8–2,0 detik saat beban tinggi	1,2–1,4 detik lebih stabil	Turun ±30%	Response Time

Hasil pengujian menunjukkan bahwa penerapan *Laravel* mampu menurunkan keberhasilan serangan keamanan hingga 80–82%**,** selaras dengan penelitian oleh Rahmat Kurniawan, (2023) dan Sulistiyani et al., (2021) Kedua studi tersebut menegaskan bahwa *Laravel* dengan validasi input, middleware autentikasi, dan proteksi CSRF dapat meningkatkan keamanan aplikasi web dibandingkan CMS konvensional.

Namun, penelitian ini memiliki keterbatasan karena uji coba hanya dilakukan pada serangan dasar (*SQL Injection, CSRF, Brute Force*) dan belum mencakup serangan lanjutan seperti *Distributed Denial of Service* (DDoS). Selain itu, pengujian dilakukan pada lingkungan lokal, sehingga performa di server produksi dengan trafik tinggi mungkin menunjukkan hasil berbeda.

Tabel 3. Perbandingan dengan Penelitian Terdahulu dan Kelemahan Penelitian Ini

Penelitian Terdahulu	Temuan Utama	Kelemahan Penelitian Ini
Rahmat Kurniawan (2023)	Laravel mampu mengurangi risiko serangan hingga 80% melalui validasi input & CSRF	Penelitian ini hanya menguji tiga jenis serangan dasar (*SQL Injection, CSRF, Brute Force*)
Sulistiyani et al. (2021)	Integrasi middleware & CSRF token meningkatkan keamanan aplikasi web hingga 82%	Pengujian masih dilakukan pada server lokal, belum diuji pada skala produksi dengan trafik tinggi
Husain et al. (2024)	reCAPTCHA menurunkan serangan login otomatis hingga 75%	Belum dilakukan uji serangan lanjutan seperti *DDoS*at au *XSS*

4. KESIMPULAN

Sistem informasi berbasis *Laravel* yang dikembangkan berhasil menggantikan platform CMS WordPress dengan peningkatan signifikan pada aspek keamanan, modularitas, dan kemudahan pengelolaan. Sistem ini

DOI: https://doi.org/10.33020/saintekom.v15i2.938

mampu menangkal serangan umum seperti *SQL Injection* dan *Cross-Site Request Forgery* melalui penerapan validasi input, *middleware otentikasi*, dan *token* CSRF. Penerapan arsitektur MVC memastikan pemisahan logika aplikasi yang lebih terstruktur sehingga mempermudah pemeliharaan dan pengembangan di masa depan. Sistem ini juga memberikan fleksibilitas penuh kepada pengembang untuk menyesuaikan alur bisnis sesuai kebutuhan BLUD, sekaligus meminimalisasi ketergantungan terhadap plugin pihak ketiga yang berpotensi menjadi celah keamanan. Dengan pondasi yang lebih kokoh ini, sistem informasi berbasis Laravel dinilai lebih andal untuk mendukung operasional digital BLUD secara jangka panjang. Penelitian selanjutnya dapat mengeksplorasi integrasi fitur tambahan seperti *audit trail* lanjutan dan *monitoring real-time* guna meningkatkan kualitas layanan sistem.

3. UCAPAN TERIMA KASIH

Terima kasih kepada BLUD atas fasilitas dan data pendukung dalam pengembangan sistem informasi berbasis Laravel.berbasis Laravel.

DAFTAR PUSTAKA

Abdul Wahid, A. (2020). *Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi*. https://www.researchgate.net/publication/346397070

Abutaleb, H., Tamimi, A., & Alrawashdeh, T. (2021). Empirical Study of Most Popular PHP Framework. *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, 608–611. https://doi.org/10.1109/ICIT52682.2021.9491679

Anggraeni, W., Purnama, I. P. A. P. M., Risqiwati, D., Sugiyanto, S., Sidharta, H. A., Budiyanta, N. E., Djunaidy, A., Vinarti, R. A., Rikasakomara, E., Mahananto, F., Kusumawardhani, R. P., & Meilani, M. (2025). Implementasi CMS WordPress dalam Pengembangan website Sekolah SLB ABCD Bakti Sosial. *Sewagati*, *9*(1), 2639–2651. https://doi.org/10.12962/j26139960.v9i1.2321

Azhar, S. A., Defriani, M., & Hermanto, T. I. (2023). UI/UX Analysis of Project Management Information System (PMIS) Website Using User-Centered Design Method. *SinkrOn*, *8*(3), 1798–1810. https://doi.org/10.33395/sinkron.v8i3.12725

Endra, R. Y., Aprilinda, Y., Dharmawan, Y. Y., & Ramadhan, W. (2021). Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website. *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, *11*(1), 48. https://doi.org/10.36448/expert.v11i1.2012

Feri Setyawan, M., & Agustin, S. (2024). Optimalisasi Sistem Pengadaan

DOI: https://doi.org/10.33020/saintekom.v15i2.938

Barang di PT Swadaya Graha Menggunakan Framework Laravel 11. *Jurnal Nasional Komputasi Dan Teknologi Informasi (JNKTI)*, *7*(6).

Husain, S. M., Azhari, L., Aksani, M. L., & Saputra, S. A. (2024). Analisis Dan Implementasi Fitur Keamanan Aplikasi Pada Framework Laravel. *JIKA (Jurnal Informatika)*, *8*(3), 281. https://doi.org/10.31000/jika.v8i3.11198

Iftitah, I., & Nuryasin, I. (2022). Penerapan Metode Research and Development Pada Proses Pengembangan Software Media Pembelajaran Practice Learning Questions Jenjang SMA. *REPOSITOR*, *4*(3), 217–228.

Imtihan, K., Ernawati, & Mutawali, L. (2022). Penerapan Research And Development (R&D) Dalam Membangun Alat Penyiraman Tanaman Otomatis Berbasis Arduino. *Jurnal Manajemen Informatika & Sistem Informasi (MISI)*, *5*, 48–55.

Kholik, A., Bisri, H., Lathifah, Z. K., Kartakusumah, B., Maufur, M., & Prasetyo, T. (2022). Impelementasi Kurikulum Merdeka Belajar Kampus Merdeka (MBKM) Berdasarkan Persepsi Dosen dan Mahasiswa. *Jurnal Basicedu*, *6*(1), 738–748. https://doi.org/10.31004/basicedu.v6i1.2045

Luh Gede Pivin Suwirmayanti, N., Adi Guna Permana, P., Aditya Artha Prayoga, P., Kadek Sukerti, N., Hadi, R., & STIKOM Bali Jl Raya Puputan No, I. (2023). Implementasi Framework Laravel Pada Sistem Informasi Akademik SMA Negeri 1 Kediri Berbasis Web. *Jurnal Nasional Komputasi Dan Teknologi Informasi*, *6*(3).

Muthia Kansha, W., Saherih, & Muchlis. (2023). Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application. *Jurnal Teknik Informatika STMIK Bangsa*.

Prastiawan, J., Permana Ganda, A., & Anwar, R. (2023). *Perancangan Dan Implementasi Sistem Pelayanan Berbasis Web Pada Perusahaan Daerah Air Minum (PDAM) Menggunakan Framework Laravel (Studi Kasus PDAM Wonomulyo) Service System Design and Implementation Web Based on Regional Drinking Water Companies (PDAM) Using Laravel Framework (Case Study of Wonomulyo PDAM)*.

Rahmat Kurniawan. (2023). *Kombinasi Agile & Waterfall Model Pengembangan Aplikasi Design Driven Development*. CV. Bintang Semesta Media.

Setya Putra, B., & Santoso, D. B. (2025). *Analisis Keamanan Website Berbasis WordPress melalui Penetration Testing untuk Meningkatkan Keamanan Digital*.

Subiksa, G. B., Peling, I. B. A., Ariawan, M. P. A., & Suardani, L. G. P. (2023). *Pengembangan CMS (Content Management System) dalam Pembuatan Website Jurusan Menggunakan Framework Laravel*. *11*(4), 2654–5101.

Sulistiyani, E., Khamida, K., Soleha, U., Amalia, R., Hartatik, S., Putra, R. S., Budiarti, R. P., & Andini, A. (2021). Implementasi Merdeka Belajar Kampus Merdeka (MBKM) pada Fakultas Kesehatan dan Non Kesehatan. *EDUKATIF : JURNAL ILMU PENDIDIKAN*, *4*(1), 686–698. https://doi.org/10.31004/edukatif.v4i1.1943

DOI: https://doi.org/10.33020/saintekom.v15i2.938