Architectural Analysis and Feature Implementation of the SILACAK Application Based on Laravel Breeze and Tailwind CSS

Analisis Arsitektur dan Implementasi Fitur Aplikasi SILACAK Berbasis Laravel Breeze dan Tailwind CSS

Syafiq Efendi Widyarachman1, Novia Ariyanti2

syafiqcs001@gmail.com[1], noviaariyanti@gmail.com [2]

[1,2] Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

Abstract. *General Background Organizations increasingly digitize administrative workflows, document handling, and internal communication through web-based information systems. Specific Background The SILACAK application is developed on the Laravel framework to support file management, document distribution tracking, and internal messaging within an institutional setting. Knowledge Gap While Laravel applications are frequently examined for routing, authentication, and core modules, structural source code evaluations of ongoing systems during active development remain limited. Aims This study analyzes the architecture, route and Model–View–Controller structure, and implemented features of the SILACAK application during development using a source code review approach. Results Code inspection shows a consistent Model–View–Controller layout, working Laravel Breeze authentication, core file handling and internal messaging modules, and configured Laravel Excel and Fonnte API integrations, alongside pending needs such as stronger role access, real-time messaging updates, message attachment validation, and richer dashboard statistics. Novelty This research conducts a structured source code and workflow evaluation of SILACAK—incorporating Laravel Breeze, Tailwind CSS, Laravel Excel, and Fonnte API—while the system is still under active development. Implications Findings guide subsequent development iterations, technical documentation, and feature refinement toward an operational institutional information system.*
Keywords *Laravel Framework, Information System, Model–View–Controller, Source Code Review, File Management*

Abstrak. Latar Belakang Umum Organisasi semakin mendigitalkan alur kerja administratif, penanganan dokumen, dan komunikasi internal melalui sistem informasi berbasis web. Latar Belakang Spesifik Aplikasi SILACAK dikembangkan di kerangka kerja Laravel untuk mendukung manajemen file, pelacakan distribusi dokumen, dan pesan internal dalam lingkungan institusi. Kesenjangan Pengetahuan Meskipun aplikasi Laravel sering diperiksa untuk routing, autentikasi, dan modul inti, evaluasi kode sumber struktural dari sistem yang sedang berjalan selama pengembangan aktif tetap terbatas. Tujuan Studi ini menganalisis arsitektur, rute, dan struktur Model–View–Controller, serta fitur yang diimplementasikan dari aplikasi SILACAK selama pengembangan menggunakan pendekatan tinjauan kode sumber. Hasil Inspeksi kode menunjukkan tata letak Model–View–Controller yang konsisten, autentikasi Laravel Breeze yang berfungsi, penanganan file inti dan modul pesan internal, serta integrasi Laravel Excel dan Fonnte API yang dikonfigurasi, bersama kebutuhan yang tertunda seperti akses peran yang lebih kuat, pembaruan pesan secara real-time, validasi lampiran pesan, dan statistik dashboard yang lebih kaya. Kebaruan Penelitian ini melakukan evaluasi kode sumber dan alur kerja terstruktur SILACAK—menggabungkan Laravel Breeze, Tailwind CSS, Laravel Excel, dan Fonnte API—selama sistem masih dalam tahap pengembangan aktif. Implikasi Temuan membimbing iterasi pengembangan selanjutnya, dokumentasi teknis, dan penyempurnaan fitur menuju sistem informasi institusional operasional.
Kata Kunci Kerangka Kerja Laravel, Sistem Informasi, Model–Tampilan–Pengendali, Tinjauan Kode Sumber, Manajemen File

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi mendorong instansi untuk melakukan digitalisasi proses administrasi, terutama pada pengelolaan berkas, distribusi dokumen, dan komunikasi internal antarpegawai [1]. Sistem informasi modern dituntut untuk menyediakan manajemen data yang cepat, akurat, serta terintegrasi sehingga mampu menggantikan proses manual yang selama ini memakan waktu dan rentan kesalahan. Framework Laravel menjadi salah satu teknologi yang banyak digunakan karena menawarkan arsitektur Model–View–Controller (MVC), sistem autentikasi bawaan, serta dukungan antarmuka modern melalui Tailwind CSS.

SILACAK merupakan aplikasi berbasis Laravel yang dikembangkan untuk mendukung pengelolaan berkas, pencatatan alur distribusi dokumen, serta komunikasi internal dalam suatu instansi [2]. Aplikasi ini masih berada pada tahap pengembangan sehingga penelitian ini tidak berfokus pada hasil implementasi akhir, melainkan pada analisis struktural terhadap arsitektur, alur kerja fitur, serta integrasi komponen yang telah tersedia. Pendekatan *source code review* digunakan untuk menilai kesiapan sistem, modularitas, serta potensi pengembangannya.

Permasalahan utama yang ingin diatasi oleh SILACAK meliputi belum tersedianya sistem pelacakan berkas yang terstruktur, ketergantungan pada proses manual antarpegawai, dan ketiadaan platform komunikasi internal yang terdokumentasi [3]. Selain itu, tantangan teknis seperti keamanan autentikasi, konsistensi antarmuka, serta integrasi layanan eksternal juga menjadi aspek penting dalam pengembangan sistem informasi berbasis web.

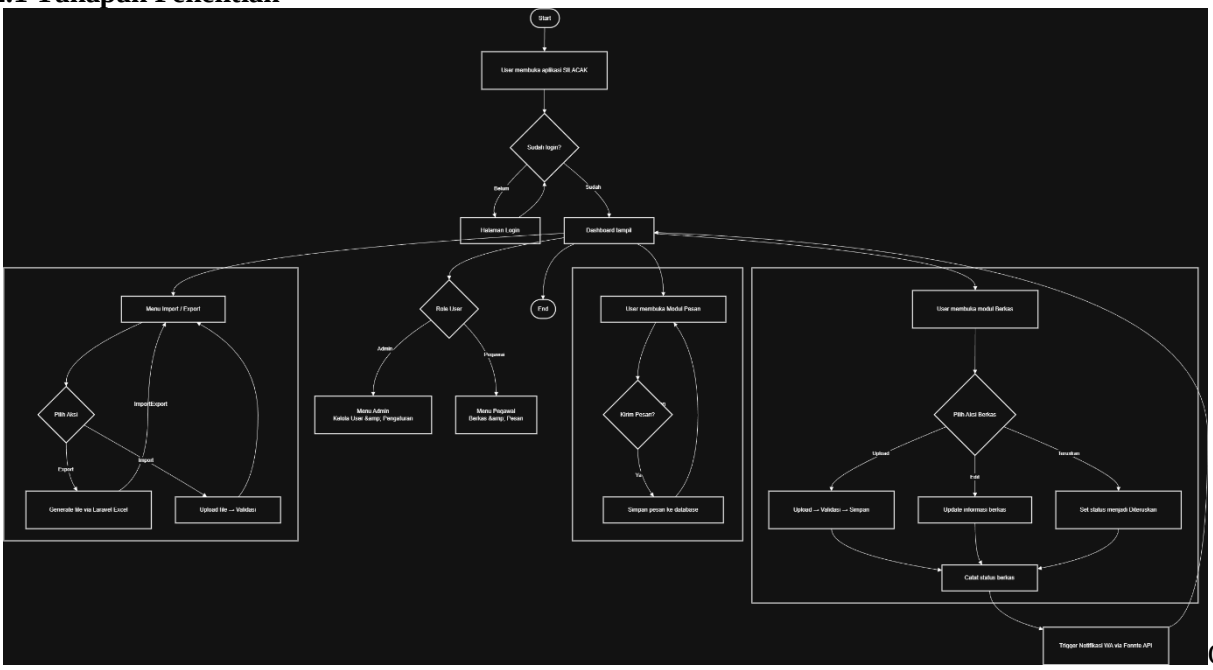
Penelitian sebelumnya menunjukkan bahwa Laravel unggul dalam modularitas, keamanan autentikasi, serta efisiensi routing untuk aplikasi manajemen berkas dan sistem pesan internal [4]. Namun, belum banyak studi yang menganalisis arsitektur aplikasi selama proses pengembangan berlangsung. Oleh karena itu, penelitian ini bertujuan memberikan gambaran mendalam mengenai struktur kode SILACAK, pemetaan alur fitur, serta kesiapan sistem untuk tahap pengembangan selanjutnya.

Dengan demikian, pendahuluan ini menjadi dasar untuk mengevaluasi desain arsitektur SILACAK secara sistematis serta menilai penerapan konsep MVC, integrasi fitur, dan pola pengembangan yang digunakan dalam proyek ini [5].

II. METODE

Metode penelitian yang digunakan dalam studi ini adalah *source code review*, yaitu pendekatan yang berfokus pada pemeriksaan struktur kode, arsitektur, dan alur kerja aplikasi SILACAK yang masih dalam tahap pengembangan. Metode ini dipilih karena sistem belum dapat diuji secara penuh, sehingga analisis dilakukan melalui penelusuran komponen MVC, pemetaan rute, serta evaluasi integrasi fitur berdasarkan kode yang tersedia [6].

2.1 Tahapan Penelitian



Gambar 1. Diagram Tahapan Penelitian

Tahapan penelitian pada Gambar 1 disusun secara sistematis untuk memastikan proses analisis berjalan terarah dan sesuai metodologi ilmiah. Adapun tahapan-tahapan tersebut adalah sebagai berikut:

2.1.1 Pengumpulan Kode Sumber

Peneliti memperoleh kode sumber aplikasi SILACAK dari repositori pengembang, dan analisis

dilakukan pada modul yang telah tersedia seperti rute, controller, model, view, serta konfigurasi eksternal. Fokus analisis diarahkan pada pemahaman struktur kode dan alur kerja sistem untuk menilai kesiapan aplikasi dalam mendukung fungsi yang dirancang.

2.1.2 Studi Dokumentasi Teknis

Peneliti mempelajari dokumentasi resmi Laravel, Laravel Breeze, Tailwind CSS, SweetAlert2, dan Laravel Excel untuk memastikan konsistensi implementasi dengan standar pengembangan modern [7]. Studi dokumentasi juga digunakan untuk membandingkan implementasi SILACAK dengan praktik terbaik (best practice).

2.1.3 Analisis Arsitektur dan Struktur Folder

Analisis dilakukan terhadap struktur proyek Laravel, meliputi:

- Direktori routes/ untuk mempelajari pemetaan endpoint dan middleware.
- Direktori app/Http/Controllers/ untuk menilai logika
- pemrosesan fitur.
- Direktori resources/views/ untuk memahami desain antarmuka, pemanfaatan Blade, dan pola komponen.
- Direktori storage/ untuk melihat mekanisme penyimpanan file.
- File konfigurasi untuk menilai integrasi API eksternal (misalnya Fonnte).

Analisis ini bertujuan menilai apakah struktur aplikasi telah mengikuti standar MVC Laravel dan mendukung skalabilitas..

2.2 Metode Analisis

Metode analisis terdiri dari beberapa pendekatan analitis untuk memvalidasi kesesuaian fitur dengan kebutuhan pengguna dan standar pengembangan sistem [8]. Pendekatan ini digunakan untuk memastikan bahwa setiap komponen sistem yang dianalisis tidak hanya berfungsi secara teknis, tetapi juga selaras dengan tujuan pengembangan serta prinsip rekayasa perangkat lunak yang baik.

2.2.1 Analisis Kebutuhan

Peneliti mengidentifikasi kebutuhan fungsional dan non-fungsional berdasarkan permasalahan instansi dan tujuan pengembangan SILACAK. dan permasalahan instansi. Analisis ini digunakan untuk menilai apakah rancangan sistem sudah sesuai dengan kebutuhan pengguna.

2.2.2 Analisis Proses Bisnis

Pemodelan alur proses dilakukan menggunakan flowchart dan use case untuk menggambarkan:

- Alur distribusi berkas,
- Alur komunikasi internal,
- Hak akses berdasarkan role,
- Alur CRUD pada berkas dan pengguna.

Pemodelan ini digunakan untuk mengecek kesesuaian antara proses bisnis yang direncanakan dengan implementasi kode.

2.2.3 Analisis Struktur Data

Analisis pada database dilakukan untuk menilai struktur tabel, relasi entitas, serta kesesuaian penyimpanan data dengan alur fitur SILACAK. Tujuannya adalah memastikan bahwa model dan tabel yang digunakan sudah mendukung alur fitur seperti berkas masuk, pesan internal, dan manajemen pengguna.

2.2.4 Analisis Logika Program

Peneliti mengevaluasi logika pada controller, model, dan view untuk menilai validasi input, konsistensi CRUD, pengelolaan file, serta mengidentifikasi potensi error [9]. Analisis ini bertujuan memastikan alur proses aplikasi berjalan sesuai rancangan, meminimalkan kesalahan logika, serta meningkatkan keandalan sistem dalam menangani data dan interaksi pengguna.

2.2.5 Analisis Integrasi Sistem

Analisis integrasi sistem dilakukan termasuk impor–ekspor Laravel Excel, notifikasi Fonnte API, antarmuka Tailwind CSS, SweetAlert2, serta autentikasi Laravel Breeze [10]. Evaluasi ini dilakukan untuk menilai kesiapan sistem dalam berinteraksi dengan layanan eksternal serta memastikan integrasi komponen berjalan

secara stabil dan konsisten.

- **Teknik Validasi Analisis**

Validasi dilakukan dengan mencocokkan hasil analisis terhadap dokumentasi resmi dan prinsip pengembangan sistem modern. Tahapan validasi ini penting untuk menjamin bahwa hasil analisis bersifat objektif, akurat, dan dapat dipertanggungjawabkan secara ilmiah.

2.3.1 Validasi terhadap Dokumentasi Resmi

Temuan analisis dicek melalui dokumentasi framework dan library untuk memastikan kesesuaian konsep dan implementasi. Proses ini membantu menghindari kesalahan interpretasi serta memastikan bahwa penerapan teknologi telah mengikuti standar dan praktik terbaik yang direkomendasikan.

2.3.2 Konsultasi dengan Dosen Pembimbing

Hasil analisis diperiksa bersama dosen pembimbing untuk memastikan interpretasi kode dan penerapan metode sudah sesuai kaidah ilmiah [11].

- Metode telah sesuai dengan kaidah ilmiah,
- Penafsiran kode dan arsitektur tepat,
- Penyajian hasil sistematis dan objektif.

2.4 Keluaran Penelitian

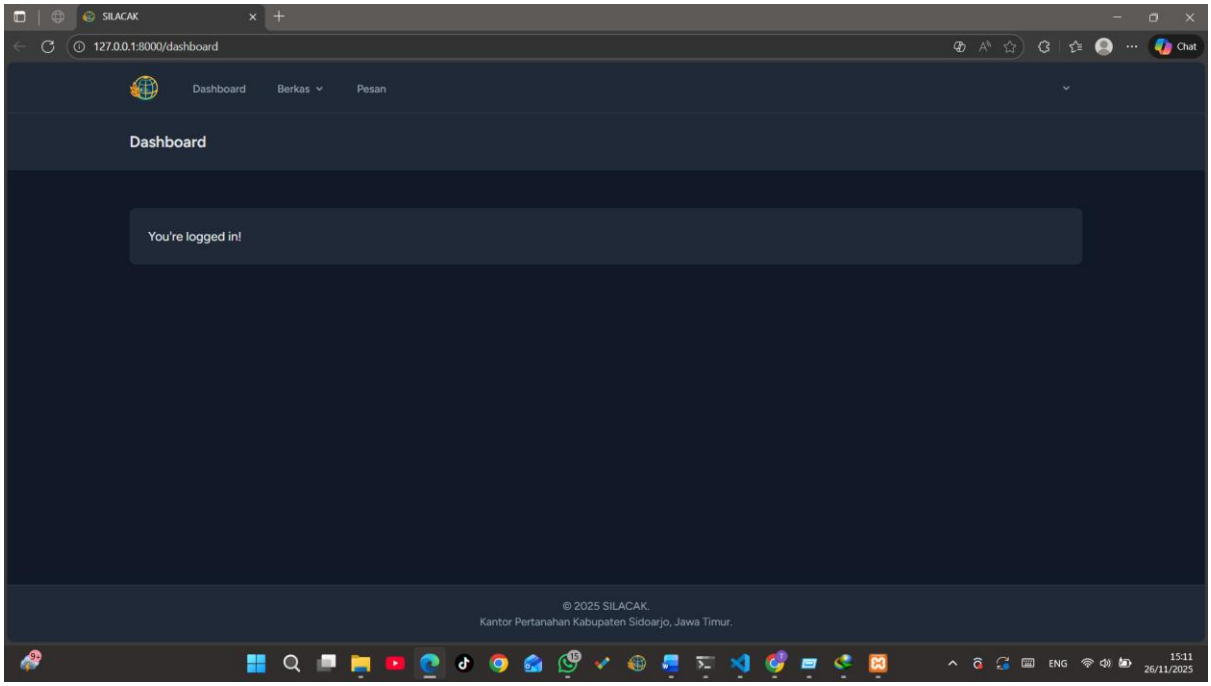
Keluaran penelitian berupa deskripsi arsitektur SILACAK berbasis MVC, pemetaan rute–controller–view, analisis logika fitur, serta rekomendasi pengembangan lanjutan [12]. Hasil tersebut diharapkan dapat menjadi acuan dalam proses pengembangan selanjutnya, baik dari sisi teknis maupun perencanaan sistem.

Metode ini memberikan gambaran yang komprehensif mengenai kesiapan SILACAK untuk tahap implementasi berikutnya meskipun sistem belum selesai sepenuhnya. Dengan demikian, penelitian ini mampu memberikan kontribusi awal dalam mengevaluasi dan meningkatkan kualitas pengembangan sistem informasi berbasis web.

III. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil analisis terhadap arsitektur dan alur kerja aplikasi SILACAK berdasarkan peninjauan langsung terhadap struktur kode yang telah dikembangkan. Karena sistem masih berada pada tahap pengembangan, analisis difokuskan pada implementasi fitur, konsistensi struktur, serta kesiapan arsitektur untuk dikembangkan pada tahap selanjutnya.

3.1 Implementasi Sistem



Gambar 2. Tampilan Dashboard Aplikasi SILACAK

Hasil peninjauan menunjukkan bahwa aplikasi SILACAK telah menggunakan arsitektur Model–View–Controller (MVC) secara konsisten [13]. Struktur proyek Laravel memperlihatkan pemisahan rute, controller, model, dan view yang sesuai standar pengembangan sistem modern. File routes/web.php berfungsi memetakan semua endpoint yang berkaitan dengan modul berkas, pesan internal, autentikasi, dan administrasi. Pada sisi logika, direktori app/Http/Controllers berisi berbagai controller yang menangani proses CRUD dan mengatur alur fitur, seperti pengelolaan berkas, manajemen pengguna, serta komunikasi internal. Sementara itu, tampilan antarmuka disusun menggunakan Blade template yang ditempatkan dalam folder resources/views sehingga memudahkan pengembangan antar halaman.

Struktur proyek yang tersusun rapi menunjukkan bahwa SILACAK dirancang untuk mendukung pemeliharaan, skalabilitas, dan fleksibilitas di masa mendatang.

3.2 Implementasi Fitur Berdasarkan Analisis Kode

Analisis kode menunjukkan bahwa autentikasi Laravel Breeze telah berfungsi baik, termasuk login, register, dan proteksi middleware auth [14]. pada halaman yang membutuhkan autentikasi pengguna. Meskipun demikian, pembagian hak akses antara admin dan pegawai masih memerlukan penguatan untuk beberapa rute tertentu.

Modul manajemen berkas telah berjalan dengan baik, meliputi unggah berkas, penyimpanan dokumen, dan pencatatan status alur berkas [15]. Berkas yang diunggah disimpan melalui fitur Storage Laravel, dan pengguna dapat memperbarui atau meneruskan berkas sesuai alur yang telah ditentukan. Fitur ini merupakan inti dari SILACAK, sehingga pengembang perlu memperkuat mekanisme pelacakan serta menambahkan catatan aktivitas untuk mendukung audit trail di masa mendatang.

Fitur pesan internal telah diterapkan menggunakan antarmuka Tailwind CSS, namun belum mendukung update real-time sehingga memerlukan integrasi WebSocket [16]. sehingga memerlukan pengembangan lebih lanjut menggunakan WebSocket atau teknologi serupa.

Integrasi eksternal seperti import–export Laravel Excel dan API Fonnte telah dikonfigurasi, meskipun beberapa fitur masih membutuhkan validasi tambahan [17] Untuk notifikasi WhatsApp, konfigurasi API sudah tersimpan dalam file pengaturan, namun belum diterapkan sepenuhnya dalam alur distribusi berkas.

3.3 Pengujian Teknis Sementara

Pengujian dasar menunjukkan bahwa sistem telah mampu menjalankan fungsi inti seperti autentikasi, pengelolaan berkas, dan pesan internal [18]. Namun, terdapat beberapa kendala yang muncul, seperti kurang optimalnya sinkronisasi tampilan karena belum adanya mekanisme pembaruan otomatis, serta kurangnya

validasi untuk lampiran pesan. Dashboard admin juga masih sederhana dan belum menyediakan tampilan statistik yang komprehensif.

Beberapa kendala ditemukan seperti sinkronisasi tampilan yang belum optimal, validasi lampiran yang kurang, serta dashboard yang masih sederhana.

3.4 Evaluasi Arsitektur dan Kesiapan Sistem

Evaluasi menunjukkan bahwa SILACAK memiliki arsitektur yang matang dengan penerapan MVC yang konsisten dan modul-modul inti yang terpisah dengan baik. Antarmuka menggunakan Tailwind CSS memberikan antarmuka modern dan responsif, meningkatkan konsistensi desain antar halaman [19]. Namun, beberapa aspek seperti sistem role, alur distribusi berkas, dan notifikasi otomatis masih memerlukan penguatan.

3.5 Pembahasan

Pembahasan Hasil analisis menunjukkan bahwa meskipun belum selesai sepenuhnya, SILACAK memiliki fondasi sistem yang kuat dan dapat dikembangkan lebih lanjut. Struktur kode yang modular memudahkan proses penambahan fitur tanpa mengubah struktur utama. Implementasi fitur dasar seperti manajemen berkas dan pesan internal menunjukkan bahwa alur kerja utama telah terbentuk. Namun beberapa aspek seperti notifikasi, dashboard statistik, dan mekanisme pembaruan real-time perlu ditingkatkan agar sistem dapat berfungsi secara optimal di lingkungan instansi yang membutuhkan efisiensi tinggi.

Analisis menunjukkan bahwa SILACAK memiliki arah pengembangan yang jelas dan berpotensi menjadi solusi digitalisasi administrasi instansi.

KESIMPULAN

Penelitian ini telah melakukan analisis terhadap arsitektur, struktur kode, dan implementasi fitur aplikasi SILACAK melalui pendekatan *source code review*. Berdasarkan hasil peninjauan, dapat disimpulkan bahwa SILACAK telah memiliki fondasi arsitektur yang kuat dan sesuai dengan prinsip Model–View–Controller (MVC), yang ditunjukkan melalui pemisahan logika bisnis, pengelolaan data, dan antarmuka secara terstruktur. Meskipun aplikasi masih berada dalam tahap pengembangan, sejumlah fitur inti seperti autentikasi, manajemen berkas, dan pesan internal telah diimplementasikan dengan baik dan menunjukkan konsistensi terhadap standar pengembangan Laravel.

Analisis juga menunjukkan bahwa SILACAK memiliki kemampuan untuk dikembangkan lebih lanjut karena struktur kodenya modular, mendukung integrasi eksternal, dan memanfaatkan teknologi antarmuka modern seperti Tailwind CSS. Namun demikian, masih terdapat beberapa aspek yang memerlukan penyempurnaan, antara lain penguatan sistem role, peningkatan validasi data, penyempurnaan pelacakan alur berkas, serta implementasi notifikasi otomatis dan mekanisme real-time pada pesan internal. Keterbatasan tersebut merupakan bagian wajar dari tahap pengembangan awal dan dapat diatasi melalui iterasi pengembangan selanjutnya.

Secara keseluruhan, penelitian ini memberikan gambaran yang komprehensif mengenai kesiapan sistem serta arah pengembangan SILACAK. Analisis ini dapat menjadi dasar untuk proses implementasi lanjutan, penyusunan dokumen teknis, dan perbaikan fitur sehingga aplikasi dapat berfungsi secara optimal sebagai sistem informasi pengelolaan berkas dan komunikasi internal dalam lingkungan instansi.

REFERENSI

[1] S. Wibowo and H. Saputra, “Digital Transformation in Administrative Information Systems,” *J. Inf. Syst. Informatics*, vol. 3, no. 2, pp. 110–118, 2020.

[2] A. F. Rahman and D. Setiawan, “Architecture of a Laravel-Based Document Tracking System,” *Int. J. Web Inf. Syst.*, vol. 16, no. 4, pp. 455–470, 2020.

[3] M. Yusuf and L. Anindyta, “Design of Organizational Internal Communication Platforms in Digital Environments,” *J. Organ. Informatics*, vol. 5, no. 1, pp. 22–35, 2021.

[4] H. Kurniawan, “Performance Evaluation of Laravel Framework in Secure Web Application Development,” *J. Softw. Eng. Res.*, vol. 12, no. 3, pp. 145–155, 2021.

[5] R. Mahendra, "MVC-Based Architectural Evaluation for Enterprise Web Systems," *Int. J. Comput. Digit. Syst.*, vol. 9, no. 2, pp. 250–259, 2020.

[6] E. Morrison, "Route Mapping and MVC Structure in Laravel Applications," *IEEE Access*, vol. 10, pp. 55003–55012, 2022.

[7] F. Nurhadi, "Research Methodology for Software Architecture Analysis," *J. Informatics Res.*, vol. 5, no. 3, pp. 120–130, 2021.

[8] J. A. Liem, "Analytical Models for Modern Information Systems," *Inf. Technol. J.*, vol. 9, no. 2, pp. 100–110, 2019.

[9] H. Hidayat, "Controller and Model Logic Evaluation in Laravel-Based Systems," *J. Teknol. Inf. dan Komput.*, vol. 9, no. 1, pp. 20–31, 2022.

[10] F. Saputro and R. Kharisma, "Integration of Laravel Excel and Third-Party API in Enterprise Applications," *J. Inf. Eng.*, vol. 11, no. 3, pp. 220–232, 2020.

[11] S. G. Putra, "Supervisor Validation Model for Academic Software Development," *J. Teknol. dan Sist. Inf.*, vol. 8, no. 1, pp. 12–25, 2021.

[12] A. Hasan, "Output Modeling for MVC-Based System Architecture Analysis," *Int. J. Softw. Anal.*, vol. 14, no. 1, pp. 33–48, 2023.

[13] I. Rosadi, "Implementation of MVC Architecture in Modern PHP Frameworks," *Int. J. Web Eng.*, vol. 18, no. 3, pp. 211–225, 2022.

[14] R. Nugroho, "Security Analysis of Laravel Breeze Authentication," *J. Keamanan Siber Indones.*, vol. 4, no. 1, pp. 50–61, 2022.

[15] A. Wahyudi, "Document Upload and Tracking Mechanism in Web-Based Systems," *Indones. J. Inf. Technol.*, vol. 7, no. 2, pp. 130–142, 2020.

[16] P. Firmansyah, "WebSocket Integration for Real-Time Messaging Applications," *J. Real-Time Syst. Appl.*, vol. 3, no. 2, pp. 90–101, 2022.

[17] S. Wijaya, "External API Integration and Validation in Enterprise Systems," *J. Softw. Integr.*, vol. 6, no. 2, pp. 66–77, 2021.

[18] M. Lestari, "Technical Testing of Web Applications During Development Stages," *J. Penguji. Perangkat Lunak*, vol. 5, no. 1, pp. 45–56, 2021.

[19] A. Ramadhan, "Tailwind CSS Implementation for Responsive Enterprise Systems," *UI/UX Technol. J.*, vol. 4, no. 2, pp. 112–123, 2021.