JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 14 No. 3, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v14i3.10303

RANCANG BANGUN CONTENT MANAGEMENT SYSTEM DENGAN EDITORIAL WORKFLOW BERBASIS WEB PADA PORTAL BERITA KLOJEN.COM

Farrel Aqeel Danendra [1*], Ismy Dahlia Handayani[2], Faydzaki Ananta Putra[3]
Yoga Ari Tofan[4]

[1,2,3,4]Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jawa Timur; Jl Raya, Rungkut Madya, Gunung Anyar, Surabaya, Jawa Timur 60294; Telp. +62 (031) 870 6369 / fax +62 (031) 870 6372

Keywords:
content management system; editorial workflow; portal berita; Laravel; Next.js; cron job; black-box testing.

Corespondent Email:
farrqeel@gmail.com

Abstrak. Portal berita digital memerlukan sistem pengelolaan konten yang terstruktur untuk mendukung alur kerja redaksi. Penelitian ini merancang dan membangun Content Management System (CMS) berbasis web pada portal berita Klojen.com dengan menerapkan arsitektur decoupled dan pola Service-Repository. Sistem dikembangkan menggunakan Laravel 12 sebagai backend dan Next.js 15 sebagai frontend, dengan fitur utama meliputi manajemen artikel dengan revision history, editorial workflow berbasis role (jurnalis, editor, admin), penjadwalan publikasi otomatis menggunakan cron job, pengelolaan pengguna dengan kontrol akses berbasis peran, serta pencarian artikel menggunakan full-text search. Pengujian fungsional menggunakan metode black-box testing menunjukkan bahwa seluruh fitur berjalan sesuai spesifikasi yang ditetapkan, termasuk mekanisme auto-publish artikel terjadwal dan penanganan dampak penghapusan pengguna terhadap data terkait.

Abstract. *Digital news portals require a structured content management system to support editorial workflows effectively. This study designs and develops a web-based Content Management System (CMS) for the Klojen.com news portal by implementing a decoupled architecture and the Service-Repository pattern. The system is developed using Laravel 12 as the backend and Next.js 15 as the frontend. Its main features include article management with revision history, role-based editorial workflow (journalists, editors, and administrators), automated scheduled publishing using cron jobs, user management with role-based access control, and article search using full-text search. Functional testing using the black-box testing method shows that all system features operate according to the defined specifications, including the scheduled article auto-publish mechanism and the handling of user deletion impacts on related data.*

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong berbagai instansi dan perusahaan media untuk membangun sistem pengelolaan konten berbasis web yang terstruktur. Roby et al. [1] menyatakan bahwa website yang dikelola melalui Content Management System (CMS) memberikan kemudahan bagi organisasi dalam mengelola informasi secara mandiri tanpa memerlukan keahlian pemrograman yang mendalam. Dalam konteks industri media digital, CMS menjadi komponen krusial yang memungkinkan tim redaksi untuk mengelola siklus hidup konten berita secara efisien, mulai dari tahap penulisan draf oleh jurnalis, proses peninjauan dan persetujuan oleh editor, hingga publikasi ke portal berita [2].

Martha et al. [2] dalam penelitiannya tentang CMS berbasis Next.js menyimpulkan bahwa pemilihan framework frontend yang tepat seperti Next.js dengan kemampuan Server-Side Rendering (SSR) dapat meningkatkan performa dan optimasi SEO pada portal berita. Sementara itu, Maranatha et al. [3] menekankan pentingnya pengujian fungsional menggunakan metode black-box testing untuk memastikan seluruh fitur CMS berjalan sesuai spesifikasi sebelum dirilis ke pengguna.

PT. Ketik Media Siber merupakan perusahaan media digital yang mengelola beberapa portal berita, salah satunya adalah Klojen.com yang berfokus pada informasi seputar Kota Malang dengan empat pilar utama: wisata, kuliner, hotel, dan pendidikan. Sebelum adanya Content Management System (CMS), proses pengelolaan konten berita dilakukan

secara manual melalui komunikasi informal antara jurnalis dan editor menggunakan aplikasi pesan instan. Kondisi ini menyebabkan beberapa permasalahan signifikan yang menghambat produktivitas dan kualitas publikasi: (1) kesulitan melacak riwayat perubahan artikel karena tidak adanya sistem pencatatan revisi yang terstruktur, (2) tidak adanya mekanisme penjadwalan publikasi otomatis sehingga editor harus melakukan publish secara manual pada waktu yang ditentukan, yang sering kali terlewatkan karena kesibukan, (3) ketiadaan kontrol akses berbasis peran yang terstruktur sehingga semua pengguna memiliki hak akses yang sama tanpa pembedaan antara jurnalis, editor, dan admin, serta (4) tidak adanya sistem pencarian artikel yang efisien sehingga editor kesulitan menemukan artikel lama untuk referensi atau verifikasi.

Beberapa penelitian sebelumnya telah membahas aspek-aspek terkait pengembangan CMS. Al Farisi et al. [4] mengimplementasikan Role-Based Access Control (RBAC) menggunakan Filament Shield pada sistem informasi pengelolaan tugas akhir, menunjukkan bahwa RBAC efektif dalam mengatur hak akses pengguna berdasarkan peran. Model RBAC yang dikemukakan oleh Sandhu et al. [5] menjadi landasan teoritis untuk implementasi kontrol akses berbasis peran pada berbagai sistem informasi. Rozi [6] dalam penelitian tugas akhirnya membahas penerapan Service-Repository Pattern pada pengembangan back end berbasis REST API, yang terbukti meningkatkan modularitas dan maintainability kode. Sementara itu, Irawan dan Sanusi

[7] mengembangkan sistem ERP berbasis web menggunakan kombinasi Next.js dan Laravel dengan pendekatan SDLCn Prototyping, menunjukkan bahwa kombinasi kedua framework ini mampu menghasilkan aplikasi web yang responsif dan scalable.

Berdasarkan penelitian-penelitian sebelumnya, dapat diketahui bahwa pengembangan Content Management System (CMS), implementasi Role-Based Access Control (RBAC), serta penggunaan arsitektur decoupled dengan kombinasi Laravel dan Next.js telah banyak dilakukan pada berbagai domain aplikasi. Namun, belum ditemukan penelitian yang secara spesifik mengintegrasikan seluruh komponen tersebut dalam satu sistem CMS portal berita yang mendukung editorial workflow terstruktur, revision history, serta mekanisme scheduled publishing otomatis. Selain itu, sebagian besar penelitian terdahulu lebih berfokus pada pengelolaan konten secara umum dan belum menekankan kebutuhan spesifik industri media digital yang memerlukan alur editorial kolaboratif, kontrol akses berbasis peran, serta ketepatan waktu publikasi berita. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun Content Management System berbasis web pada portal berita Klojen.com dengan mengintegrasikan editorial workflow, scheduled publishing otomatis, dan role-based access control menggunakan arsitektur decoupled berbasis Laravel dan Next.js.

Artikel ini diorganisasikan sebagai berikut. Bagian 2 membahas tinjauan pustaka yang menjadi landasan teoritis penelitian. Bagian 3 menjelaskan metode penelitian yang digunakan. Bagian 4 menyajikan hasil implementasi dan pembahasan. Bagian 5 menarik kesimpulan dan saran untuk pengembangan selanjutnya.

2. TINJAUAN PUSTAKA

2.1 Content Management system

Content Management System (CMS) adalah aplikasi yang memfasilitasi pembuatan, pengelolaan, modifikasi, dan publikasi konten digital secara terstruktur dan kolaboratif. Roby et al. [1] mendefinisikan CMS sebagai sistem

yang memisahkan concerns antara pembuatan konten (content creation), pengelolaan alur kerja (workflow management), dan presentasi (presentation layer). Martha et al. [2] dalam penelitiannya tentang CMS berbasis Next.js pada company profile menyimpulkan bahwa CMS memberikan kemudahan bagi organisasi dalam mengelola konten secara mandiri tanpa memerlukan keahlian pemrograman yang mendalam, serta framework Next.js dengan kemampuan Server-Side Rendering (SSR) mampu meningkatkan performa dan optimasi SEO. Maranatha et al. [3] menekankan bahwa CMS yang berkualitas harus melalui pengujian fungsional yang komprehensif menggunakan metode black-box testing sebelum dirilis ke pengguna.

2.2 Editorial Workflow dan Role-Based Access Control

Editorial workflow merupakan serangkaian proses terstruktur yang dilalui konten berita dari tahap penulisan hingga publikasi. Workflow editorial yang efektif dalam media daring mencakup empat tahapan utama: drafting (penulisan draf oleh jurnalis), reviewing (peninjauan oleh

editor), approval (persetujuan untuk publikasi), dan publishing (proses publikasi ke portal berita). Setiap tahapan melibatkan aktor dengan peran spesifik yang memiliki hak akses berbeda terhadap konten.

Role-Based Access Control (RBAC) adalah model keamanan yang membatasi akses sistem berdasarkan peran yang diberikan kepada pengguna. Sandhu et al. [5] dalam paper fundamentalnya yang diterbitkan oleh NIST menjelaskan bahwa RBAC memungkinkan pembagian akses yang terstruktur berdasarkan peran pengguna, yang mengurangi kompleksitas manajemen hak akses dibandingkan pendekatan tradisional. Al Farisi et al. [4] mengimplementasikan RBAC menggunakan Filament Shield pada sistem informasi pengelolaan tugas akhir dan menunjukkan bahwa pendekatan ini efektif dalam mengatur hak akses pengguna berdasarkan peran secara granular. Dalam konteks CMS editorial, peran-peran tersebut meliputi: jurnalis (penulis artikel dengan hak membuat dan mengedit draf), editor (peninjau artikel dengan hak approve, reject, atau publish), dan admin (pengelola sistem dengan hak penuh termasuk manajemen pengguna).

2.3 Laravel Framework

Laravel adalah framework PHP open-source yang mengikuti arsitektur Model-View-Controller (MVC) dan menyediakan berbagai fitur untuk mempercepat pengembangan aplikasi web. Irawan dan Sanusi [7] memanfaatkan Laravel sebagai backend dalam pengembangan sistem ERP berbasis web, menunjukkan bahwa Laravel menyediakan fitur lengkap untuk pengembangan backend termasuk Eloquent ORM, middleware, Artisan CLI, Task Scheduler, dan Mailable. Beberapa fitur Laravel yang dimanfaatkan dalam penelitian ini meliputi: Eloquent ORM untuk interaksi dengan database, middleware untuk autentikasi dan otorisasi, Artisan Command Line Interface untuk membuat custom commands, Task Scheduler untuk penjadwalan tugas otomatis, dan Mailable untuk pengiriman email. Laravel juga mendukung penerapan berbagai pola arsitektur seperti Service-Repository Pattern yang memisahkan logika bisnis dari operasi database, sehingga kode menjadi lebih modular, testable, dan mudah dipelihara [6].

2.4 Next.js Framework

Next.js adalah framework React yang dikembangkan oleh Vercel untuk membangun aplikasi web modern dengan performa tinggi. H. A. Jartarghar [8] dalam penelitiannya tentang React Apps dengan Server-Side Rendering menggunakan Next.js menyimpulkan bahwa SSR secara signifikan meningkatkan performa loading halaman dan optimasi SEO dibandingkan pendekatan Client-Side Rendering (CSR) murni. Martha et al. [2] juga memanfaatkan Next.js dalam pengembangan CMS company

profile dan menunjukkan bahwa fitur-fitur Next.js seperti Server-Side Rendering (SSR), Static Site Generation (SSG), Incremental Static Regeneration (ISR), dan App Router memberikan fleksibilitas tinggi dalam pengembangan frontend. Dalam konteks penelitian ini, Next.js digunakan untuk membangun antarmuka frontend CMS (dashboard, tulis berita, tinjauan artikel, preview berita) serta portal berita publik

(beranda, detail artikel, kategori) dengan kemampuan SEO yang optimal melalui dynamic metadata generation.

2.5 Service-Repository Pattern

Service-Repository Pattern adalah pola arsitektur perangkat lunak yang memisahkan logika bisnis (service layer) dari operasi akses data (repository layer). Rozi [6] dalam penelitian tugas akhirnya di UGM mengimplementasikan Service-Repository Pattern pada pengembangan back end berbasis REST API untuk sistem informasi properti dan menyimpulkan bahwa pola ini meningkatkan modularitas, testability, dan maintainability kode secara signifikan. Dalam implementasinya pada Laravel: Controller hanya menangani HTTP request/response dan validasi input, Service layer berisi seluruh logika bisnis dan orkestrasi antar komponen, sedangkan Repository layer menangani query database dan pemetaan objek. Pemisahan ini memudahkan pengujian unit karena setiap layer dapat diuji secara independen dengan mocking dependency-nya.

2.6 Decoupled Architecture

Decoupled architecture (atau headless architecture) memisahkan backend dan frontend menjadi dua aplikasi independen yang berkomunikasi melalui Application Programming Interface (API). Penelitian tentang performa arsitektur decoupled [9] menunjukkan bahwa pendekatan ini memberikan fleksibilitas dalam pemilihan teknologi untuk masing-masing sisi serta memungkinkan skalabilitas yang lebih baik dibandingkan arsitektur monolitik. Irawan dan Sanusi [7] juga mengadopsi pendekatan decoupled dengan memisahkan frontend Next.js dan backend Laravel dalam sistem ERP-nya, yang memungkinkan tim frontend dan backend bekerja secara paralel. Dalam penelitian ini, Laravel sebagai backend mengekspos RESTful API yang dikonsumsi oleh Next.js sebagai frontend, dengan JSON Web Token (JWT) sebagai mekanisme autentikasi.

2.8 Black-Box Testing

Black-box testing adalah metode pengujian perangkat lunak yang mengevaluasi fungsionalitas sistem tanpa melihat struktur kode internal. Maranatha et al. [3] dalam penelitiannya tentang pengujian CMS Quest Master menggunakan black-box testing dengan teknik equivalence partitioning dan menyimpulkan bahwa metode ini efektif dalam menemukan cacat fungsional pada sistem. Saian et al. [10] juga menerapkan black-box testing dengan teknik boundary value analysis pada CMS Sekolahku dan menunjukkan bahwa kombinasi equivalence partitioning dan boundary value analysis mampu mencakup skenario pengujian secara komprehensif. Dalam konteks penelitian ini, black-box testing dipilih karena fokus pengujian adalah pada perilaku sistem dari perspektif pengguna akhir, bukan pada implementasi internal. Setiap skenario pengujian didokumentasikan dengan langkah-langkah yang jelas, hasil yang diharapkan, dan hasil aktual untuk memudahkan pelacakan dan verifikasi.

3. METODE PENELITIAN

Penelitian ini menggunakan metode pengembangan perangkat lunak Waterfall yang terdiri dari lima tahapan utama, yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Metode ini dipilih karena sesuai untuk pengembangan sistem yang memiliki kebutuhan terdefinisi dengan jelas sejak awal serta minim perubahan selama proses pengembangan. Penelitian dilaksanakan di PT. Ketik Media Siber selama periode Februari hingga Juni 2026.

3.1. Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan melalui observasi langsung alur kerja redaksi dan diskusi dengan stakeholder. Hasil analisis kebutuhan sistem diidentifikasi menjadi dua kategori: kebutuhan fungsional dan kebutuhan non-fungsional, sebagaimana dirangkum pada Tabel 1.

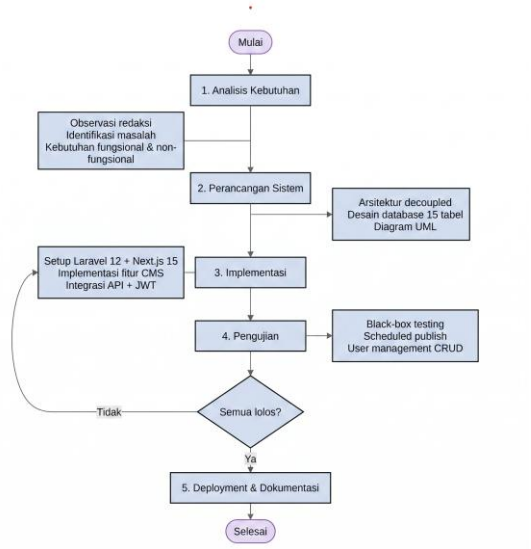
Tabel.1 *Analisis Kebutuhan Sistem*

Kategori	**Kebutuhan**	**Deskripsi**
Fungsional	Manajemen Artikel	CRUD artikel dengan revision history dan auto-generate slug

	Editorial Workflow	Alur draft → review → publish berbasis role
	Scheduled Publish	Penjadwalan publikasi otomatis menggunakan cron job
	User Management	CRUD pengguna dengan kontrol akses berbasis peran
	Pencarian Artikel	Full-text search dengan indexing otomatis
	Preview Artikel	Preview tampilan artikel sebelum dipublikasikan
Non-Fungsional	Keamanan	JWT authentication dan RBAC
	Performa	SSR/ISR untuk optimasi SEO dan kecepatan
	Maintainability	Service-Repository Pattern untuk modularitas
	Responsiveness	Tampilan optimal di desktop dan mobile

3.2. Alur penelitian

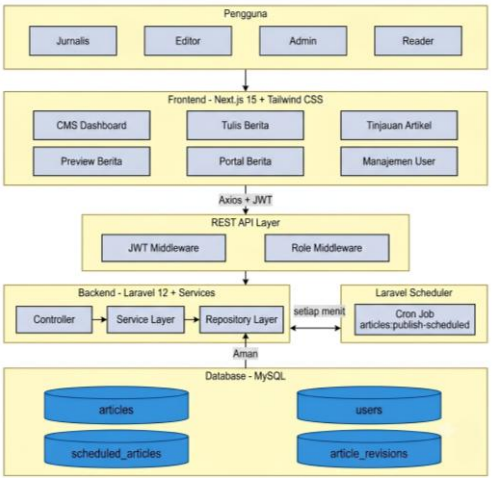
Alur tahapan penelitian digambarkan pada Gambar 1. Setiap tahapan menghasilkan luaran spesifik yang menjadi masukan untuk tahapan berikutnya. Setiap tahapan menghasilkan luaran spesifik yang menjadi dasar untuk tahapan berikutnya hingga sistem siap diimplementasikan untuk perbaikan sebelum dilanjutkan ke deployment.



Gambar 1. Alur Metode Penelitian

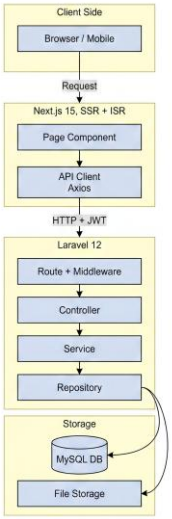
3.3. Perancangan Sistem

Sistem dirancang menggunakan arsitektur decoupled yang memisahkan backend (Laravel 12, PHP 8.4, dan database MySQL) serta frontend (Next.js 15, TypeScript, Tailwind CSS). Komunikasi antara frontend dan backend dilakukan melalui RESTful API dengan autentikasi JWT. Backend menerapkan Service-Repository Pattern untuk memisahkan logika bisnis dari operasi database. . Gambar 2 menunjukkan arsitektur sistem secara keseluruhan, sedangkan Gambar 3 menggambarkan interaksi antar komponen dalam arsitektur decoupled.



Gambar 2. Arsitektur Sistem CMS

Klojen.com



Gambar 3. Diagram Interaksi Komponen Decoupled Architecture

3.4. Implementasi

Implementasi dilakukan secara bertahap sesuai tahapan yang telah dirancang dengan prioritas fitur sebagai berikut: (1) setup proyek dan konfigurasi lingkungan pengembangan termasuk database migration dan seeder, (2) implementasi fitur inti CMS meliputi manajemen artikel dan editorial workflow dengan revision history, (3) implementasi scheduled publish dengan Artisan Command dan Laravel Task Scheduler, (4) implementasi user management dengan RBAC dan email notification, serta (5) integrasi frontend-backend menggunakan Axios dan pengujian fungsional.

3.5. Pengujian

Pengujian dilakukan menggunakan metode black-box testing sebagaimana dijelaskan pada bagian 2.7. Skenario pengujian dirancang untuk mencakup seluruh fitur utama sistem dengan fokus pada: (1) pengujian scheduled publish yang mencakup penjadwalan artikel, verifikasi auto-publish oleh cron job, validasi waktu minimal penjadwalan, dan pembatalan jadwal, (2) pengujian user management yang mencakup operasi CRUD lengkap, kontrol akses berbasis peran, dan penanganan dampak penghapusan pengguna. Setiap skenario pengujian didokumentasikan dengan langkah-langkah detail, hasil yang diharapkan, dan screenshot hasil eksekusi untuk memudahkan verifikasi dan reproduksi.

4. HASIL DAN PEMBAHASAN

4.1. Arsitektur Sistem

Sistem CMS Klojen.com berhasil dibangun dengan arsitektur decoupled yang memisahkan backend (Laravel 12) dan frontend (Next.js 15) menjadi dua aplikasi independen seperti ditunjukkan pada Gambar 1 dan Gambar 2. Backend mengekspos RESTful API yang dikonsumsi oleh frontend melalui HTTP request dengan header Authorization berisi Bearer token JWT. Autentikasi menggunakan JWT dengan refresh token yang disimpan di database untuk mendukung mekanisme token revocation ketika terjadi perubahan role atau penghapusan pengguna.

Arsitektur ini memberikan beberapa keuntungan: (1) tim backend dan frontend dapat bekerja secara paralel tanpa saling mengganggu, (2) penggantian teknologi di salah satu sisi (misalnya migrasi dari Next.js ke framework lain) tidak mempengaruhi sisi backend selama API contract tetap sama,

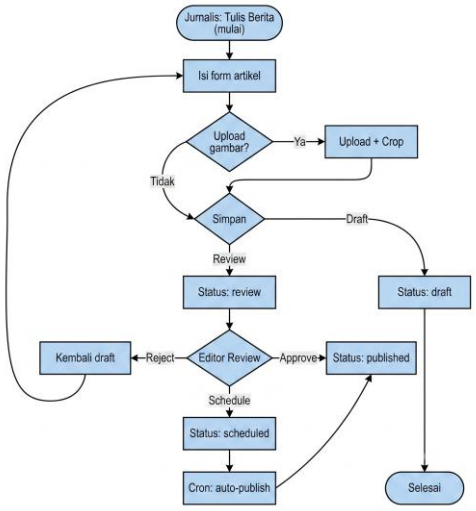
(3) API yang sama dapat dikonsumsi oleh berbagai client termasuk aplikasi mobile di masa depan, dan (4) skalabilitas yang lebih baik karena backend dan frontend dapat di-deploy dan di-scale secara independen.

4.2. Editorial Workflow Berbasis Role

Editorial workflow diimplementasikan menggunakan RBAC dengan tiga peran utama: jurnalis, editor, dan admin. Gambar 3 menggambarkan alur editorial workflow secara lengkap dari tahap penulisan hingga publikasi. Jurnalis dapat membuat artikel baru dengan status draft dan mengirimkannya ke review. Editor dapat meninjau artikel yang berstatus review dengan tiga opsi: approve (mengubah status menjadi published), reject (mengembalikan ke draft dengan catatan), atau schedule (menjadwalkan publikasi pada waktu tertentu). Editor juga memiliki kemampuan untuk mengunci artikel (lock) yang sedang dikerjakan untuk mencegah konflik editing antar editor yang mengerjakan artikel yang sama secara bersamaan.

Implementasi editorial workflow berbasis role pada sistem ini berhasil mengatasi

permasalahan alur kerja manual yang sebelumnya terjadi di Klojen.com. Dengan pemisahan hak akses antara jurnalis, editor, dan admin, setiap proses pengelolaan artikel menjadi lebih terstruktur dan terkontrol. Jurnalis hanya berfokus pada pembuatan konten, sedangkan editor bertanggung jawab terhadap validasi kualitas artikel sebelum publikasi. Pendekatan ini mampu mengurangi risiko publikasi artikel yang belum melalui proses review serta meningkatkan akuntabilitas setiap tahapan editorial.



Gambar 4. Activity Diagram Editorial Workflow

Setiap perubahan pada artikel dicatat dalam tabel article_revision sebagai audit trail. Tabel ini menyimpan snapshot judul dan konten sebelum perubahan dilakukan, beserta identitas pengguna yang melakukan perubahan, timestamp, dan catatan perubahan. Mekanisme ini memungkinkan pelacakan riwayat perubahan artikel secara lengkap dari awal pembuatan hingga versi terkini.

Fitur revision history juga memberikan peningkatan signifikan dalam aspek auditability karena seluruh perubahan artikel tercatat secara sistematis. Hal ini memudahkan editor dalam melacak perubahan konten, mengidentifikasi sumber perubahan, dan memulihkan versi sebelumnya apabila diperlukan. Dibandingkan proses manual sebelumnya yang tidak memiliki pencatatan revisi, pendekatan ini memberikan kontrol editorial yang jauh lebih baik.

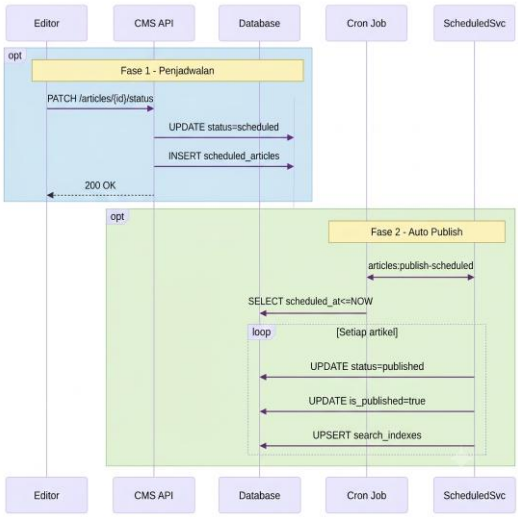
Proses pembuatan artikel baru melalui beberapa tahapan dalam service layer: (1) penentuan slug — jika pengguna mengirim slug manual, sistem memvalidasi keunikan; jika tidak, sistem melakukan auto-generate dari judul dengan loop pengecekan ke database dan penambahan suffix (-2, -3, dst.) jika terjadi duplikasi, (2) insert data artikel ke database dengan status awal draft, (3) pemrosesan relasi tag menggunakan pendekatan firstOrCreate yang membuat tag baru jika belum ada berdasarkan slug-nya, (4) penyimpanan revision pertama sebagai audit trail yang mencatat snapshot judul dan konten, serta (5) reindex artikel ke search index untuk memastikan artikel langsung muncul di hasil pencarian.

4.3. Penjadwalan Publish Otomatis

Fitur scheduled publish diimplementasikan menggunakan Artisan Command (articles:publish-scheduled) yang dipicu oleh Laravel Task Scheduler setiap menit melalui cron daemon di server. ScheduledPublishService memproses artikel yang sudah waktunya tayang dengan langkah-langkah berikut: (1) mengambil daftar artikel dari tabel scheduled_articles dengan kondisi scheduled_at <= NOW() dan is_published = false, (2) untuk setiap artikel, melakukan UPDATE status menjadi published dan published_at dengan waktu saat ini dalam satu database transaction, (3) memperbarui flag is_published menjadi true pada tabel scheduled_articles untuk menandai bahwa artikel sudah diproses, serta (4) melakukan reindex pada tabel search_indexes agar artikel langsung muncul di hasil pencarian. Gambar 5 menunjukkan sequence diagram interaksi antar komponen.

implementasi scheduled publish memberikan efisiensi yang signifikan dalam proses publikasi berita. Sebelum sistem ini diterapkan, editor harus melakukan publikasi secara manual pada waktu tertentu, yang berpotensi menimbulkan keterlambatan akibat human error atau keterbatasan operasional. Dengan mekanisme otomatis berbasis cron job, artikel dapat dipublikasikan secara tepat waktu sesuai jadwal yang ditentukan. Pendekatan ini sangat relevan bagi portal berita yang membutuhkan konsistensi waktu publikasi

untuk menjaga engagement pembaca dan relevansi informasi.



Gambar 5. Sequence Diagram Scheduled Publish

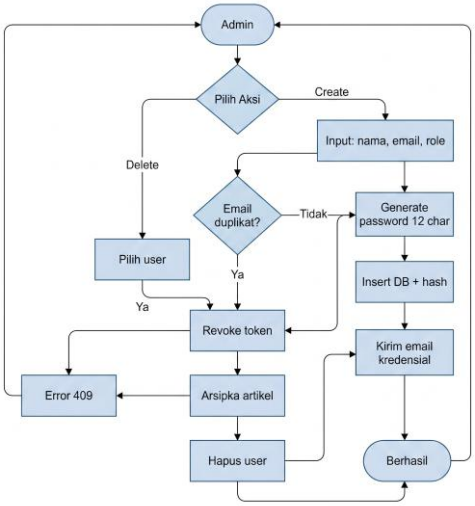
Seluruh operasi publish dibungkus dalam database transaction untuk menjamin konsistensi data (ACID properties). Jika salah satu operasi gagal (misalnya update status berhasil tetapi reindex gagal), seluruh perubahan akan di-rollback sehingga data tetap konsisten. Validasi waktu minimal 5 menit diterapkan untuk memberikan buffer sebelum cron job berikutnya berjalan. Command ini mengembalikan jumlah artikel yang berhasil di-publish sebagai output untuk keperluan monitoring.

Penggunaan database transaction juga meningkatkan reliability sistem dengan memastikan seluruh proses publikasi berjalan secara atomik. Jika salah satu proses gagal, seluruh perubahan dibatalkan sehingga inkonsistensi data dapat dihindari. Hal ini menjadi faktor penting dalam menjaga integritas data pada sistem CMS berskala produksi.

4.4. Pengelolaan Pengguna dengan RBAC

Fitur user management hanya dapat diakses oleh pengguna dengan role admin melalui middleware admin yang memvalidasi role sebelum request diproses. Fitur ini mencakup operasi CRUD lengkap dengan beberapa mekanisme keamanan: (1) validasi duplikasi email menggunakan unique constraint di database dan pengecekan di service layer, (2) auto-generate password 12 karakter dengan komposisi minimal satu huruf besar, huruf kecil, angka, dan simbol untuk memastikan kekuatan password, (3) pengiriman email berisi kredensial login menggunakan NewUserCredentials Mailable dengan template HTML yang informatif, serta (4) pencegahan admin menonaktifkan atau mengedit akunnya sendiri untuk menghindari penguncian akun secara tidak sengaja.

Pada operasi nonaktifkan user, sistem melakukan tiga langkah berurutan dalam satu operasi: (1) revoke seluruh refresh token pengguna dari tabel refresh_tokens untuk logout paksa dari semua perangkat yang aktif, (2) arsipkan seluruh artikel milik pengguna dengan mengubah statusnya menjadi archived menggunakan query UPDATE batch, serta (3) hapus data pengguna dari tabel users. Mekanisme arsipkan artikel (bukan menghapus) diterapkan untuk menjaga integritas sejarah konten — artikel tetap tersimpan di database namun tidak tampil di portal berita publik. Gambar 6 menunjukkan alur operasi user management.



Gambar 6. Activity Diagram Manajemen User (CRUD)

Proses penghapusan user melalui tiga langkah berurutan yang diorkestrasi oleh service layer: pertama, revoke seluruh refresh token dari tabel refresh_tokens untuk memastikan pengguna langsung logout dari semua perangkat; kedua, arsipkan seluruh

artikel milik pengguna dengan mengubah status menjadi archived melalui query UPDATE batch; ketiga, hapus data pengguna dari tabel users. Validasi CANNOT_MODIFY_SELF memastikan admin tidak dapat menghapus akunnya sendiri.

Implementasi user management berbasis RBAC meningkatkan keamanan sistem melalui pembatasan akses berdasarkan peran pengguna. Pendekatan ini memastikan bahwa fitur-fitur sensitif seperti pengelolaan pengguna hanya dapat diakses oleh admin. Selain meningkatkan keamanan, mekanisme ini juga mempermudah administrasi hak akses karena kontrol dilakukan pada level role, bukan individual user.

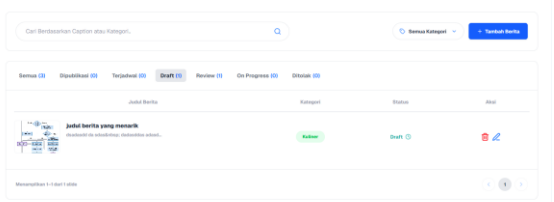
4.5. Hasil Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode black-box testing terhadap seluruh fitur utama sistem. Setiap skenario pengujian didokumentasikan dengan langkah-langkah detail, hasil yang diharapkan, hasil aktual, dan screenshot hasil eksekusi untuk memudahkan verifikasi. Tabel 2 merangkum hasil pengujian fitur editorial workflow

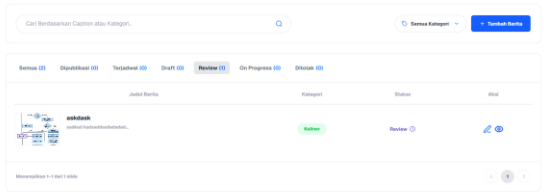
Tabel 2. Hasil Pengujian Fitur Editorial Workflow

No	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Jurnalis buat draft	Artikel status draft	Berhasil
2	Submit ke review	Status review	Berhasil
3	Editor publish	Status berubah ke published	Berhasil
4	Editor on progress	Status on progress	Berhasil

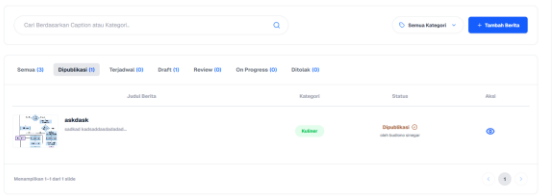
Berikut adalah dokumentasi screenshot hasil pengujian Editorial Workflow:



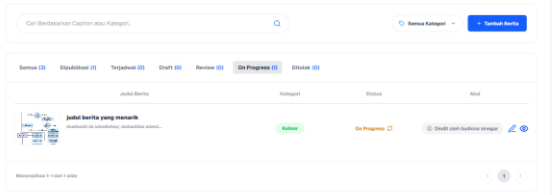
Gambar 7. Screenshot Hasil Pengujian: Artikel Status Draft



Gambar 8. Screenshot Hasil Pengujian: Artikel Status Review



Gambar 9. Screenshot Hasil Pengujian: Artikel Status Published



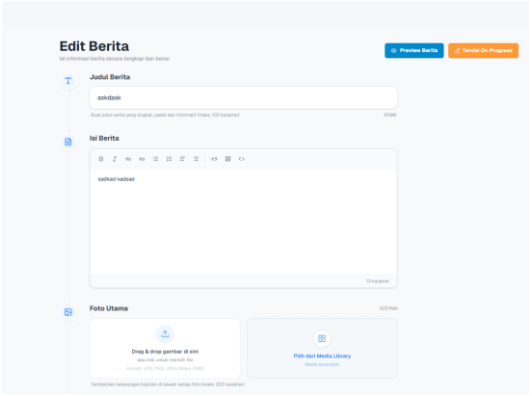
Gambar 10. Screenshot Hasil Pengujian: Artikel Status On Progress

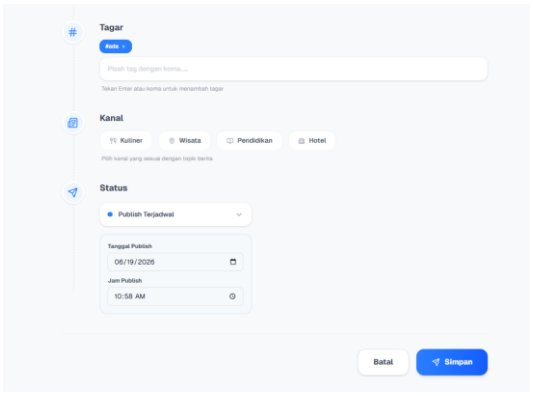
Tabel 3 merangkum hasil pengujian fitur scheduled publish.

Tabel 3. Hasil Pengujian Fitur Scheduled Publish

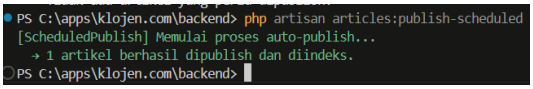
No	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Publish artikel > 5 menit dari sekarang	Status berubah menjadi scheduled	Berhasil
2	Auto-publish oleh cron job	Status published saat jadwal tiba	Berhasil
3	Waktu < 5 menit	Error 400:	Berhasil
4	Batalkan jadwal	Status menjadi draft	Berhasil
5	Artikel belum waktunya publish	Status tetap scheduled	Berhasil
6	Verifikasi search index reindex	Artikel muncul di hasil pencarian	Berhasil

Berikut adalah dokumentasi screenshot hasil pengujian scheduled publish:

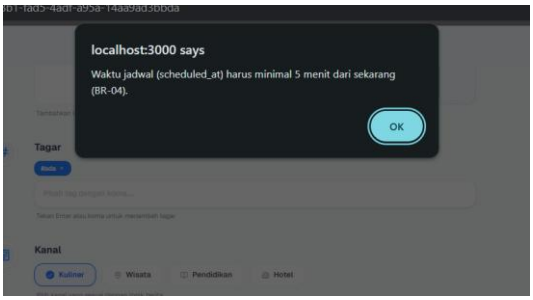




Gambar 11. Screenshot Hasil Pengujian: Jadwal Artikel



Gambar 12. Screenshot Hasil Pengujian: Auto-Publish oleh Cron Job



Gambar 13. Screenshot Hasil Pengujian: Validasi Waktu Minimal

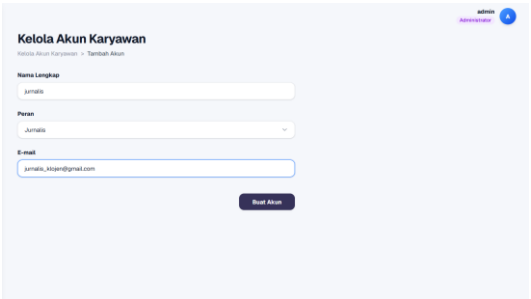
Tabel 4 merangkum hasil pengujian fitur user management.

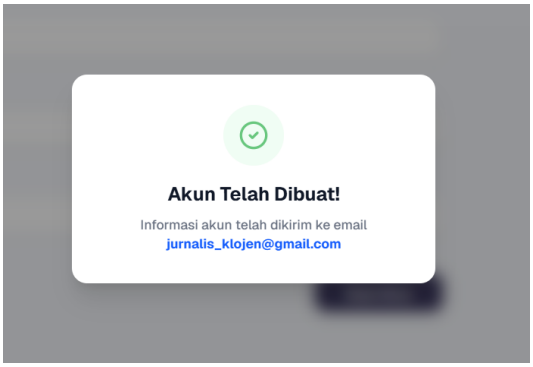
Tabel 4. Hasil Pengujian Fitur User Management

No	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Create user data lengkap	Response 201, email kredensial terkirim	Berhasil
2	Create user email duplikat	Error 409 `EMAIL_ALREADY_EXISTS`	Berhasil

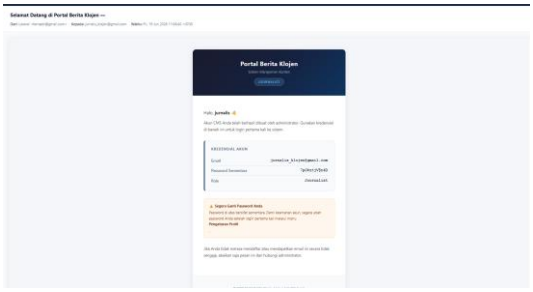
3	Verifikasi email kredensial	Email berisi nama, password sementara, dan role	Berhasil
4	Update nama user	Nama berhasil diperbarui	Berhasil
5	Update role (editor ke kontributor)	Role berubah, semua token dicabut	Berhasil
6	Nonaktifkan user (is_active = false)	User berhasil dinonaktifkan	Berhasil
7	Edit akun sendiri	Error 403 `CANNOT_MODIFY_SELF`	Berhasil
8	Nonaktifkan user	User dinonaktifkan, artikel berubah ke archived	Berhasil
9	Akses tanpa role admin	Response 403 Forbidden	Berhasil

Berikut adalah dokumentasi screenshot hasil pengujian user management

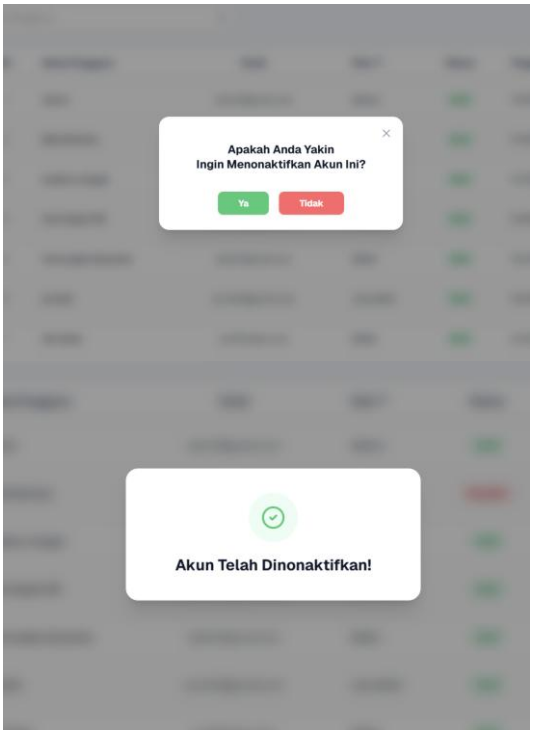


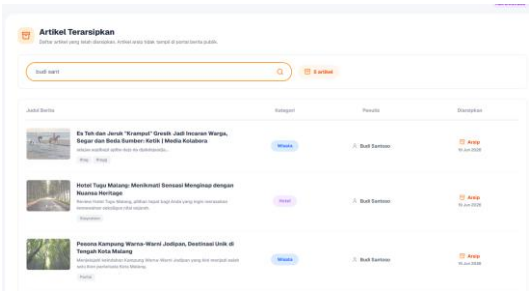


Gambar 14. Screenshot Hasil Pengujian: Create User Berhasil



Gambar 15. Screenshot Hasil Pengujian: Email Kredensial Terkirim





Gambar 16. Screenshot Hasil Pengujian: Nonaktifkan User dengan Arsip Artikel

Hasil pengujian menunjukkan bahwa seluruh fitur berjalan sesuai spesifikasi yang ditetapkan. Mekanisme database transaction pada fitur scheduled publish menjamin konsistensi data saat proses auto-publish berlangsung jika salah satu operasi gagal, seluruh perubahan akan di-rollback. Pada fitur user management, mekanisme arsipkan artikel saat penghapusan pengguna berhasil menjaga integritas data konten meskipun data pengguna telah dihapus dari sistem. Validasi kontrol akses berbasis peran juga berjalan dengan
baik, memastikan bahwa hanya admin yang dapat mengakses endpoint user management.

4.6. Pembahasan

Berdasarkan hasil implementasi dan pengujian, sistem CMS Klojen.com telah berhasil memenuhi kebutuhan yang diidentifikasi pada tahap analisis. Beberapa temuan penting dari penelitian ini meliputi:

Pertama, penerapan Service-Repository Pattern terbukti efektif dalam menghasilkan kode yang modular dan mudah dipelihara. Pemisahan logika bisnis dari operasi database memudahkan pengujian unit dan memungkinkan penggantian implementasi repository tanpa mempengaruhi service layer. Namun, pola ini juga menambah kompleksitas karena memerlukan lebih banyak file dan class dibandingkan pendekatan monolitik sederhana.

Kedua, mekanisme revision history menggunakan tabel article_revisions memberikan kemampuan pelacakan perubahan yang komprehensif. Setiap versi artikel dapat

direkonstruksi dengan mengambil snapshot dari tabel revisions. Keterbatasan pendekatan ini adalah pertumbuhan tabel yang cepat jika artikel sering diubah, sehingga diperlukan strategi archival atau pagination untuk artikel dengan banyak revisi.

Ketiga, implementasi scheduled publish menggunakan cron job dan database transaction memberikan jaminan konsistensi data yang kuat. Namun, pendekatan ini memiliki keterbatasan skalabilitas karena cron job berjalan pada satu

server. Untuk skala yang lebih besar, diperlukan message queue (seperti Redis Queue) untuk mendistribusikan tugas publish ke multiple workers.

Keempat, mekanisme arsip artikel saat menonaktifkan pengguna berhasil menjaga integritas konten tanpa kehilangan data historis. Pendekatan ini lebih aman dibandingkan hard delete karena memungkinkan pemulihan jika penghapusan pengguna dilakukan secara tidak sengaja.

5. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan beberapa hal sebagai berikut:

1. Penelitian berhasil membangun Content Management System berbasis web pada portal berita Klojen.com menggunakan arsitektur decoupled dengan Laravel dan Next.js.
2. Implementasi editorial workflow berbasis role dan revision history mampu mendukung proses pengelolaan artikel secara terstruktur.
3. Mekanisme scheduled publishing menggunakan Laravel Task Scheduler berhasil melakukan publikasi artikel secara otomatis sesuai waktu yang ditentukan.
4. Pengujian black-box terhadap 19 skenario menunjukkan seluruh fungsi sistem berjalan dengan tingkat keberhasilan 100%.

Beberapa saran untuk pengembangan selanjutnya meliputi:

1. implementasi soft delete pada tabel users untuk mempertahankan riwayat pengguna meskipun telah dihapus.
2. penambahan audit log untuk melacak seluruh perubahan yang dilakukan oleh admin

3. implementasi message queue untuk meningkatkan skalabilitas fitur scheduled publish.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada PT. Ketik Media Siber yang telah memberikan kesempatan untuk melaksanakan program magang mandiri dan terlibat secara langsung dalam pengembangan proyek Sistem Portal Berita dan CMS Redaksi Klojen.com. Terima kasih juga disampaikan kepada Bapak Yoga Ari Tofan, S.Kom., M.Kom. selaku dosen pembimbing yang telah memberikan arahan dan masukan selama pelaksanaan penelitian, serta kepada Bapak Galih Rivanto selaku Direktur IT PT. Ketik Media Siber atas bimbingan teknis selama magang.

DAFTAR PUSTAKA

[1]I. Roby, R. Cahyana, R. Kurniawati, dan L. Fitriani, "Rancang Bangun Content Management System Untuk Situs Web Instansi Pemerintahan," Jurnal Algoritma, vol. 18, no. 2, pp. 367–375, Jan 2022, doi: 10.33364/algoritma.v18i2.367.

[2]G. Martha, I. N. Y. A. Wijaya, dan N. W. Utami, "Rancang Bangun Company Profile Berbasis Content Management System Menggunakan Framework Next.Js Pada Satelit NET Komputer," Jurnal Saintikom (Jurnal Sains Manajemen Informatika dan Komputer), vol. 24, no. 2, Agu 2025, doi: 10.53513/jis.v24i2.11912.

[3]I. F. Maranatha, A. J. Santoso, dan A. W. R. Emanuel, "Pengujian Content Management System – Quest Master Menggunakan Black Box Testing (Studi Kasus: Astra Credit Companies)," Jurnal Informatika Atma Jogja, vol. 3, no. 1, Mei 2022, doi: 10.24002/jiaj.v3i1.5908.

[4]R. Al Farisi, A. R. Zayn, B. A. Nugroho, A. Heriadi, dan N. D. Susanti, "Integrasi Role-Based Access Control (RBAC) Menggunakan Filament Shield Pada Sistem Informasi Pengelolaan Tugas Akhir," Jurnal Sistem Informasi Triguna Dharma (JURSI TGD), vol. 5, no. 2, Mar 2026, doi: 10.53513/jursi.v5i2.12379.

[5]R. S. Sandhu, D. F. Ferraiolo, dan R. Kuhn, "The NIST Model for Role-Based Access Control: Towards a Unified Standard," in Proceedings of the 5th ACM Workshop on Role-Based Access Control, Berlin, Germany, 2000, pp. 47–63.

[6]M. F. Rozi, "Pengembangan Back End Berbasis REST API pada Sistem Informasi Properti Menggunakan Service Repository Pattern," Tugas Akhir, D4 Teknologi Perangkat Lunak, Universitas Gadjah Mada, Yogyakarta, 2024.

[7]A. N. Irawan dan A. P. Sanusi, "Development of a Web-Based ERP System Using Next.js and Laravel: A Prototyping SDLC Approach," SMATIKA Jurnal, vol. 16, no. 01, Mar 2026, doi: 10.32664/smatika.v16i01.2174.

[8]H. A. Jartarghar, G. R. Salanke, A. A. Kumar, dan S. Dalali, "React Apps with Server-Side Rendering: Next.js," Journal of Telecommunication, Electronic and Computer Engineering, vol. 14, 2022.

[9]D. Apolinário, D. R. F., & de França, B. B. (2021). A method for monitoring the coupling evolution of microservice-based architectures Journal of the Brazilian Computer Society, 27(17).

[10]S. D. S. Saian, N. L. Kakihary, dan T. Wahyono, "Pengujian Content Management System (CMS) Sekolahku Menggunakan Metode Black Box Testing dengan Teknik Boundary Value Analysis," IT-Explore: Jurnal Penerapan Teknologi Informasi dan Komunikasi, vol. 1, no. 2, pp. 100–113, Jun 2022, doi: 10.24246/itexplore.v1i2.2022.pp100-113.

[11]R. A. Sukamto dan M. Shalahuddin, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika Bandung, 2018.

[12]Jaelani, M., Shandyasa, I. W., & Khrisne, D. C. (2023). Integrasi Framework Laravel dalam Pengembangan Perangkat Lunak Sistem Smart PJU Berbasis Internet of Things. Jurnal SPEKTRUM, 10(4), 225–234.

[13]Hidayatullah, A. R., & Suartana, I. M. (2025). Penerapan Server Side Rendering untuk Meningkatkan Performa Website Kejaksaan Negeri X. Journal of Informatics and Computer Science (JINACS), 7(1), 12–20.

[14]S. Newman, Building Microservices: Designing Fine-Grained Systems, 2nd ed. Sebastopol: O'Reilly Media, 2021.

[15]M. Fowler, Patterns of Enterprise Application Architecture. Boston: Addison Wesley, 2003.

[16]Sofian, R., & Ghozali, A. F. (2025). *Perancangan Perangkat Lunak Berbasis Website Menggunakan Metode Waterfall*. Jurnal Teknologi Informasi dan Komunikasi.

[17]Rahayu, Y. S., Saputra, Y., & Irawan, D. (2024). *Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Mobile E-DISARPUS*. ZONAsi: Jurnal Sistem Informasi.

[18]Nurseptaji, A., et al. (2021). *Implementasi Metode Waterfall pada Perancangan Sistem Informasi Perpustakaan*. Jurnal Dialektika Informatika.