



SURAT KETERANGAN PUBLIKASI JURNAL

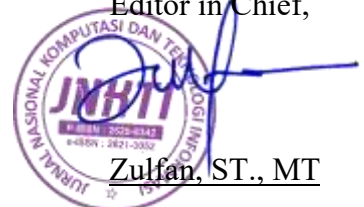
No: 10877/LoA-JNKTI.9.4/VIII/2026

Nama : Aisyah Ditta Fitrinandasari Taufik¹, Thonthowie Jauharie², Asri Wahyuni³
Afiliasi : ¹⁻³Program Studi Teknologi Informasi, Universitas Bina Sarana Informatika
Judul Artikel Jurnal : Rancang Bangun Sistem Manajemen Kontributor Berbasis Mobile pada Platform Berita MileniaNews
Email : 17220039@bsi.ac.id¹, 17220396@bsi.ac.id², asri.awx@bsi.ac.id³

Dengan ini menerangkan bahwa artikel dengan judul tersebut di atas telah direview dan dinyatakan diterima serta memenuhi syarat untuk dipublikasikan pada Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI) Vol 9. No 4. Tahun 2026 yang terbit pada bulan Agustus 2026.

Banda Aceh, Agustus 2026

Editor in Chief,



Zulfan, ST., MT

NIDN. 0115088402

****Corresponding Author***

Rancang Bangun Sistem Manajemen Kontributor Berbasis Mobile pada Platform Berita MileniaNews

Aisyah Ditta Fitrinandasari Taufik¹, Thonthowie Jauharie², Asri Wahyuni³
Program Studi Teknologi Informasi, Universitas Bina Sarana Informatika
Jl. Letnan Sutopo, BSD Sektor XIV Blok C1/1, Lengkong Gudang Timur, Kec. Serpong
E-mail: 17220039@bsi.ac.id¹, 17220396@bsi.ac.id², asri.awx@bsi.ac.id³

Submitted Date : 27 Juli 2026

Accepted Date : dd/mm/yyyy

Abstrak - MileniaNews merupakan perusahaan media digital yang berdiri sejak tahun 2018 dan mengandalkan jurnalis lepas serta kontributor dalam memproduksi konten berita. Namun, pengelolaan anggota penerima naskah, komunikasi revisi, hingga perhitungan royalti masih dilakukan secara manual melalui email dan WhatsApp, sehingga menyebabkan sistem kerja redaksi menjadi kurang efisien, minim transparansi, serta menghambat keberagaman dan kuantitas konten yang dihasilkan. Penelitian ini bertujuan untuk menganalisis sistem manajemen kontributor yang sedang berjalan pada platform MileniaNews, merancang sistem berbasis mobile yang dapat memperbaiki workflow redaktur dan kontributor tanpa mengubah alur editorial yang sudah berjalan di WordPress, serta mengintegrasikan proses moderasi, pelacakan naskah, dan perhitungan royalti secara otomatis. Metode penelitiannya menggunakan pendekatan kualitatif melalui observasi, wawancara, dan studi pustaka, sedangkan pengembangan aplikasi menggunakan metode Hybrid yang menggabungkan metode Waterfall pada tahap analisis dan perancangan, dengan Agile pada tahap implementasi yang dibagi menjadi empat sprint, serta pengujian menggunakan Black Box Testing. Hasil penelitian berupa aplikasi mobile berbasis React Native dengan backend Laravel REST API yang terintegrasi dengan WordPress melalui mekanisme webhook, sehingga memudahkan kontributor dalam mengajukan naskah, memantau status secara real-time, dan mengelola royalti, sementara redaktur tetap bekerja melalui WordPress tanpa mengubah kebiasaan kerja yang sudah berjalan.

Kata kunci: Manajemen Kontributor, Aplikasi Mobile, React Native, Laravel, WordPress, Integrasi REST API.

Abstract - MileniaNews is a digital media company established in 2018 that relies on freelance journalists and contributors to produce news content. However, the management of members, manuscript submissions, revision communication, and royalty calculations is still carried out manually through email and WhatsApp, resulting in an inefficient editorial workflow, minimal transparency, and constraints on the diversity and volume of content produced. This study aims to analyze the contributor management system currently running on the MileniaNews platform, to design a mobile-based system capable of improving the workflow between editors and contributors without altering the existing editorial process on WordPress, and to integrate the manuscript moderation, tracking, and royalty calculation processes automatically. The research method employs a qualitative approach through observation, interviews, and literature study, while the application development uses a Hybrid method combining Waterfall for the analysis and design stages with Agile for the implementation stage, which is divided into four sprints, and testing is conducted using Black Box Testing. The result of this study is a mobile application built with React Native, backed by a Laravel REST API integrated with WordPress through a webhook mechanism, making it easier for contributors to submit manuscripts, monitor their status in real time, and manage royalties, while editors continue to work through WordPress without changing their established work habits.

Keywords: Contributor Management, Mobile Application, React Native, Laravel, WordPress, API Integration

1. Pendahuluan

Perkembangan media sosial dan platform digital telah merevolusi cara masyarakat menerima dan menyampaikan informasi, sehingga publik dapat mengakses berita kapan saja tanpa bergantung pada terbitan cetak. Perubahan ini menuntut jurnalis untuk memproduksi berita secara nonstop agar sesuai dengan kebutuhan pembaca digital yang menginginkan informasi real-time, sekaligus mengubah pola kerja jurnalis lepas yang harus menyesuaikan diri dengan sistem royalti digital yang berbeda dari media cetak.

Riset terdahulu menunjukkan bahwa masyarakat semakin memilih media digital karena kepraktisan dan personalisasi informasinya [1], sementara otomatisasi dan kecerdasan buatan mempercepat produksi konten meski tetap membutuhkan pengawasan editorial yang ketat [2]. Di tengah dinamika tersebut, MileniaNews, perusahaan media digital yang telah berdiri sejak 2018, mengandalkan jurnalis lepas dan

kontributor untuk mendistribusikan naskah berita. Namun demikian, pengelolaan anggota dan penerimaan naskah masih dilakukan secara manual tanpa sistem yang terintegrasi, sehingga kerja redaksi internal cenderung tidak tertata.

Kondisi tersebut memunculkan sejumlah persoalan: jumlah anggota penulis yang minim akibat ketergantungan pada proses rekrutmen dan pelatihan manual, pengiriman naskah yang masih melalui e-mail dan WhatsApp, antarmuka situs web internal yang tidak ramah pengguna, koordinasi revisi yang berulang kali berpindah platform, serta pendataan royalti akhir bulan yang masih dilakukan secara mandiri oleh masing-masing penulis. Penelitian mengenai citizen journalism menunjukkan bahwa kolaborasi antara jurnalis warga dan editor profesional dapat memperkaya variasi topik sekaligus menjaga kredibilitas melalui kurasi yang ketat [3], sementara pengelolaan media komunitas yang menerapkan fungsi manajemen secara efektif terbukti mampu menjaga produktivitas konten meski dengan keterbatasan sumber daya [4], [5]. Temuan-temuan ini memperkuat argumen bahwa kombinasi antara keterlibatan kontributor yang lebih luas dan pengelolaan konten yang terstruktur dapat menghasilkan alur kerja redaksi yang lebih baik.

Berdasarkan uraian tersebut, penelitian ini diarahkan untuk menganalisis dan merancang sistem manajemen kontributor berbasis mobile bagi Milenianews, dengan tujuan meningkatkan efisiensi kerja redaktur, transparansi status naskah dan royalti, serta kualitas dan keberagaman konten, tanpa mengubah alur editorial berbasis WordPress yang sudah berjalan.

2. Tinjauan Pustaka

2.1. Konsep Dasar Aplikasi

Perancangan aplikasi kontributor Milenianews mengacu pada konsep arsitektur full stack yang terdiri atas frontend, backend, serta komunikasi sistem dengan basis data [6]. Frontend berperan menampilkan antarmuka agar pengguna dapat berinteraksi tanpa menyentuh baris kode, dan pada penelitian ini dibangun menggunakan React Native yang memungkinkan pengembangan aplikasi lintas platform (Android dan iOS) dengan satu baris kode JavaScript, dilengkapi dukungan integrasi pustaka pihak ketiga dan akses API perangkat.

Backend berperan mengelola logika, pengolahan, dan keamanan data di balik layar, dan pada aplikasi ini dibangun menggunakan Laravel, sebuah framework PHP yang menyediakan sistem autentikasi siap pakai, manajemen routing, ORM Eloquent, serta sistem migration basis data. Laravel berfungsi sebagai penyedia REST API yang menjembatani aplikasi mobile dengan basis data, sekaligus terhubung dua arah dengan WordPress melalui REST API bawaannya (endpoint.wp-json/v2/) menggunakan HTTP client Guzzle, sehingga Laravel dapat berperan sebagai sistem utama pengelola data kontributor sementara WordPress tetap menjadi platform publikasi bagi pembaca.

Komunikasi antara sistem dan basis data dikelola sepenuhnya melalui Eloquent ORM, dimana setiap tabel direpresentasikan sebagai model, sehingga operasi data dapat dilakukan tanpa menulis query SQL secara manual. Alur data berjalan mulai dari request aplikasi mobile, validasi melalui Form Request, pemrosesan di Controller, penyimpanan melalui Model, hingga pengembalian respon dalam format JSON.

2.2. Metode Algoritma

Algoritma didefinisikan sebagai rangkaian langkah sistematis, logis, dan berurutan untuk menyelesaikan masalah, mencakup tahap input, proses, dan output [7]. Aplikasi manajemen kontributor MileniaNews menerapkan dua algoritma utama. Pertama, algoritma Quick Sort, sebuah metode pengurutan berbasis divide and conquer dengan nilai pivot sebagai acuan pembagian data, digunakan untuk mengurutkan riwayat pengajuan poin dan daftar artikel berdasarkan waktu. Kedua, algoritma Linear Search, yang memeriksa setiap elemen data secara berurutan hingga ditemukan kecocokan, digunakan pada fitur pencarian artikel berdasarkan judul atau topik.

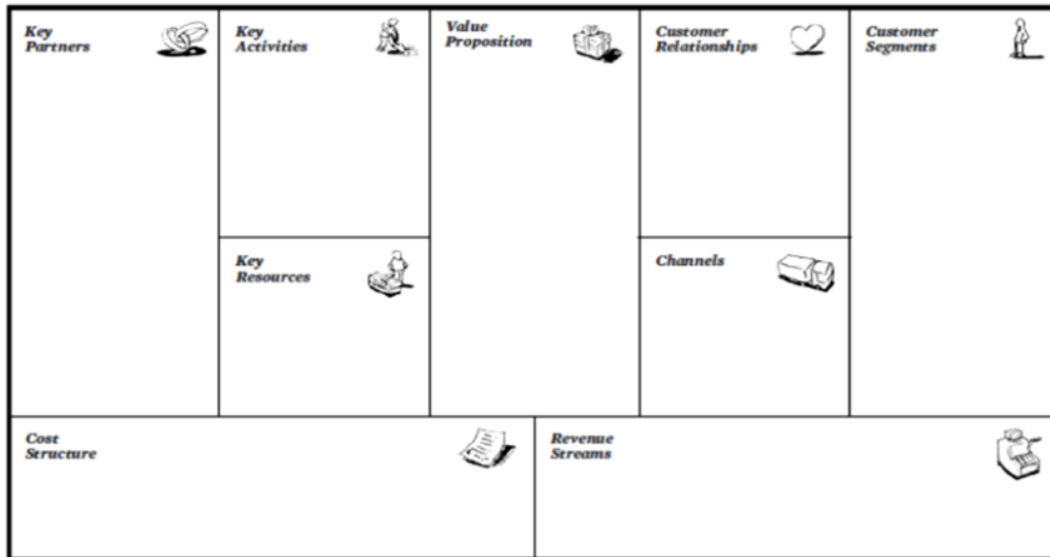


Gambar 1. Metode Hybrid

2.3. Business Model Canvas (BMC)

Business Model Canvas merupakan alat visual dengan sembilan elemen pokok yang saling berhubungan untuk menggambarkan bagaimana nilai bisnis dibuat, disampaikan, dan diperoleh [8], [9]. Kesembilan elemen tersebut, yaitu Customer Segments, Value Propositions, Channels, Customer Relationships, Revenue Streams, Key Resources, Key Activities, Key Partnerships, dan Cost Structure, digunakan dalam penelitian ini untuk memetakan model bisnis sistem manajemen kontributor secara menyeluruh sebelum masuk ke tahap perancangan.

The Business Model Canvas



Gambar 2. Business Model Canvas

2.4. Penelitian Terkait

Penelitian mengenai sistem informasi manajemen tugas multi-peran berbasis web membuktikan bahwa sistem multi-peran mampu meningkatkan efisiensi koordinasi antara pihak dibandingkan cara manual [10], yang relevan dengan struktur peran kontributor, redaktur, dan administrator pada penelitian ini. Studi perbandingan metode Agile dan Waterfall menunjukkan bahwa Waterfall unggul pada proyek dengan kebutuhan yang sudah pasti, sedangkan Agile unggul pada proyek yang membutuhkan fleksibilitas [11], yang mendasari pemilihan pendekatan Hybrid pada penelitian ini. Kajian lain mengenai perbandingan metode pengembangan perangkat lunak menegaskan bahwa agile merupakan pilihan tepat untuk aplikasi mobile yang dinamis karena siklus sprint-nya memungkinkan penyesuaian fitur berkelanjutan [12]. Sementara itu, penelitian mengenai desain dan implementasi RESTful web service membuktikan bahwa pendekatan REST API mampu menghubungkan aplikasi web dan mobile lintas platform serta bahasa pemrograman secara lancar dan mendukung sinkronisasi data real-time [13], yang menjadi rujukan utama dalam merancang integrasi antara aplikasi mobile, backend Laravel, dan WordPress pada penelitian ini.

3. Metode Penelitian

3.1. Framework React Native

Sisi antarmuka pengguna aplikasi dikembangkan menggunakan *framework* React Native. *Framework* React Native memiliki kemampuan membangun aplikasi *mobile* lintas *platform*, yaitu Android dan iOS, dari satu baris kode JavaScript yang sama, React Native menggunakan konsep komponen berbasis React, dimana setiap elemen antarmuka seperti tombol, formulir, daftar artikel, dan panel status dibangun sebagai komponen yang dapat digunakan ulang [6]. Pendekatan ini mempercepat proses pengembangan sekaligus menjaga konsistensi tampilan di berbagai ukuran layar perangkat Android.

Dalam implementasinya, React Native dipadukan dengan Expo CLI dan Expo Application Service (EAS) untuk mempermudah proses build dan deployment tanpa memerlukan konfigurasi native yang kompleks. EAS juga digunakan untuk menghasilkan file APK yang kemudian diunggah ke Google Play Store. Fitur *hot reloading* pada React Native memungkinkan tim pengembang melihat perubahan antarmuka secara langsung tanpa perlu melakukan kompilasi ulang penuh, sehingga iterasi pengembangan dalam setiap sprint dapat dilakukan lebih cepat.

Komunikasi antara *frontend* React Native dan server dilakukan melalui pemanggilan HTTP menggunakan metode fetch atau library Axios, mengirimkan dan menerima data dalam format JSON. Token JWT yang diperoleh saat login disimpan secara aman di perangkat dan disertakan pada setiap

request ke endpoint yang memerlukan autentikasi, sehingga sesi pengguna tetap terjaga tanpa perlu login ulang setiap membuka aplikasi.

3.2. *Framework Laravel*

Sisi server aplikasi dikembangkan menggunakan *framework* Laravel berbasis PHP. *Framework* Laravel menyediakan struktur pengembangan yang terorganisir melalui pola arsitektur MVC (*Model-View-Controller*), dimana *Model* Menangani logika data dan interaksi dengan basis data, *Controller* mengelola logika bisnis dan memproses request dari client, sedangkan *View* pada koneksi API digantikan oleh respons JSON yang dikirimkan ke *frontend* React Native.

Laravel memiliki fitur Eloquent ORM yang memudahkan interaksi antara kode program dan basis data MySQL tanpa perlu menulis query SQL secara langsung [6]. Setiap tabel dalam basis data direpresentasikan sebagai Model Eloquent sehingga operasi CRUD dapat dilakukan menggunakan sintaks PHP yang ringkas dan ekspresif. Selain itu, Laravel menyediakan sistem *routing* yang memungkinkan pendefinisian endpoint REST API secara terstruktur, middleware untuk validasi token JWT dan otorisasi akses berdasarkan peran pengguna (kontributor, redaktur, administrator), serta mekanisme *schedule job* untuk menjalankan tugas terjadwal seperti *polling* status ke WordPress REST API sebagai *fallback*.

Kemampuan sistem dijamin melalui implementasi autentikasi berbasis JSON Web Token (JWT). Saat kontributor berhasil *login*, Laravel menghasilkan token JWT yang ditandatangani secara kriptografis. Token ini kemudian disertakan pada setiap *request* selanjutnya dari *frontend*, dan Laravel memverifikasi keabsahannya sebelum memproses *request* tersebut. Dengan mekanisme ini, tidak ada sesi yang perlu disimpan di sisi server, sehingga sistem bersifat *stateless* dan lebih mudah untuk diskalakan.

3.3. *WordPress*

Keputusan desain paling mendasar dalam arsitektur sistem ini adalah mempertahankan WordPress sebagai *platform* eksekusi putusan editorial. Redaktur tetap melakukan seluruh pekerjaan editorial, meliputi penyuntingan naskah, pemberian catatan revisi, persetujuan, penolakan, hingga publikasi melalui antarmuka WordPress yang sudah mereka kuasai. Hal ini penting untuk memastikan adopsi sistem baru tidak terhambat oleh resistensi pengguna terhadap perubahan kebiasaan kerja.

Integrasi antara *backend* Laravel dan WordPress dibangun di atas dua mekanisme yang saling melengkapi. Integrasi dilakukan dengan WordPress, salah satu Content Management System (CMS) yang memungkinkan pengelolaan konten website tanpa keahlian teknis pemrograman [16], mekanisme utama adalah *webhook*, dimana WordPress dikonfigurasi untuk mengirimkan HTTP POST *request* ke *endpoint webhook* yang disediakan oleh Laravel setiap kali terjadi perubahan status pada sebuah naskah. *Payload webhook* berisi informasi identifikasi naskah dan status barunya, sebagai contoh *published*, *rejecter*, atau *pending revision*. Setelah menerima *payload* ini, Laravel memperbarui status naskah di basis datanya sendiri dan memicu pengiriman notifikasi push kepada kontributor yang bersangkutan.

Untuk menangani kemungkinan kegagalan pengiriman *webhook* akibat gangguan jaringan atau downtime sementara, sistem dilengkapi dengan mekanisme *fallback* berupa *polling* terjadwal. Melalui fitur *schedule job* Laravel, sistem secara berkala melakukan *query* ke WordPress REST API untuk melakukan pemeriksaan apakah terdapat perubahan status naskah yang belum tersinkronkan. Kombinasi *webhook* sebagai mekanisme utama dan *polling* sebagai *fallback* ini memastikan konsistensi data antara WordPress dan basis data Laravel tetap terjaga dalam kondisi apapun. Seluruh komunikasi antara Laravel dan WordPress REST API menggunakan format JSON dengan autentikasi *Application Password* yang disediakan oleh WordPress.

3.4. *Metode Hybrid*

Pengembangan aplikasi menggunakan pendekatan Hybrid, yaitu kombinasi antara metode Waterfall dan Agile. Pendekatan ini memiliki tahap analisis kebutuhan dan perancangan sistem yang memerlukan perencanaan yang matang dan terstruktur, sementara tahap implementasi memerlukan fleksibilitas untuk menyesuaikan fitur berdasarkan masukan dari *stakeholder* secara berkelanjutan.

Waterfall unggul pada proyek dengan kebutuhan yang sudah pasti dan terstruktur, sedangkan Agile lebih efektif untuk proyek yang memerlukan adaptasi dinamis [11]. Kombinasi keduanya dalam pendekatan Hybrid memberikan keseimbangan antara keterstrukturan dan fleksibilitas yang sesuai dengan kebutuhan pengembangan sistem ini. Adapun tahapan pengembangan dalam metode Hybrid, sebagai berikut:

1. *Requirements*, pada tahap ini dilakukan identifikasi dan pendefinisian seluruh kebutuhan sistem secara menyeluruh. Hasil analisis dikelompokkan menjadi dua bagian, yaitu kebutuhan sistem dan kebutuhan pengguna.
2. *System Design*, pada tahap ini dilakukan perancangan sistem meliputi perancangan arsitektur sistem dan struktur basis data yang digunakan untuk penyimpanan data. Pada tahap ini juga

dilakukan perancangan antarmuka (UI/UX) aplikasi guna memastikan tampilan yang dihasilkan memenuhi kebutuhan dan kemudahan pengguna bagi redaktur maupun kontributor. Perancangan perilaku sistem dilakukan menggunakan diagram UML, meliputi *Use Case Diagram* dan *Activity Diagram*. Selain itu, desain basis data dirancang menggunakan *Entity Relationship Diagram* (ERD) dan *Logical Record Structure* (LRS) untuk memastikan konsistensi dan integritas struktur data.

3. *Development in Sprint*, pada tahap ini dilakukan pengembangan secara iteratif menggunakan pendekatan Agile yang dibagi menjadi empat *sprint*, sebagai berikut:
 - a. *Sprint 1*: Berfokus pada pengembangan fitur autentikasi pengguna dan manajemen keanggotaan kontributor.
 - b. *Sprint 2*: Mencakup pengembangan fitur penulisan dan pengajuan artikel oleh kontributor.
 - c. *Sprint 3*: Meliputi pengembangan fitur peninjauan tulisan, pemberian catatan revisi, dan sistem notifikasi oleh redaktur.
 - d. *Sprint 4*: Berfokus pada pengembangan fitur pemantauan status tulisan dan pendataan tulisan sebagai dasar perhitungan royalti.

Setiap *sprint* dilaksanakan secara berurutan dan menghasilkan fitur yang dapat langsung diuji sebelum melanjutkan ke *sprint* berikutnya.

4. *Testing*, tahap ini dilakukan pada setiap akhir *sprint* untuk memastikan setiap fitur yang dikembangkan berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan. Metode pengujian yang digunakan adalah Black Box Testing, yaitu pengujian yang berfokus pada fungsionalitas sistem tanpa memperhatikan struktur internal kode. Apabila ditemukan kesalahan atau ketidaksesuaian fungsi, maka dilakukan perbaikan terlebih dahulu sebelum pengembangan dilanjutkan ke *sprint* berikutnya.
5. *Deployment*, setelah seluruh fitur berhasil melewati tahap pengujian dan dinyatakan siap pakai, dilakukan implementasi sistem secara menyeluruh. Tahap ini merupakan tahap akhir dari proses pengembangan, dimana aplikasi Sobat Milenia di *deploy* dan siap digunakan oleh pengguna, yaitu redaktur dan kontributor MileniaNews.

3.5. Metode Algoritma

Dalam pengembangannya, terdapat dua algoritma utama yang mendukung pengelolaan data, yaitu algoritma pengurutan dan pencarian. Untuk kebutuhan pengurutan data, sistem menggunakan *Quick Sort*. *Quick Sort* merupakan algoritma pengurutan yang bekerja dengan prinsip *divide and conquer*. Cara kerjanya adalah dengan menentukan nilai pivot sebagai acuan, lalu membagi data menjadi dua kelompok berdasarkan perbandingan dengan pivot tersebut. Proses ini dilakukan secara rekursif hingga seluruh data terurutkan [7]. Dalam sistem ini, *Quick Sort* digunakan untuk mengurutkan riwayat pengajuan poin atau daftar artikel berdasarkan waktu, dari yang terbaru hingga terlama, sehingga informasi menjadi lebih teratur dan mudah diakses.

Linear Search merupakan algoritma pencarian sederhana yang bekerja dengan memeriksa setiap elemen data satu persatu dari awal hingga akhir. Pencarian akan berhenti begitu data ditemukan atau seluruh data telah diperiksa. Metode ini sangat cepat jika data berada di posisi awal, namun akan lebih lambat jika data berada di akhir atau tidak ditemukan [7]. Sistem manajemen kontributor memanfaatkan *Linear Search* pada fitur pencarian artikel berdasarkan judul atau topik tertentu, sehingga kedua jenis pengguna dapat menemukan artikel yang diperlukan dengan lebih cepat dan mudah.

3.6. Black Box Testing

Pengujian Black Box adalah teknik pengujian perangkat lunak yang mengevaluasi fungsionalitas sistem berdasarkan input yang diberikan dan output yang dihasilkan, tanpa memperdulikan struktur yang ada didalamnya, arsitektur kodenya, atau logika dibalikanya [15]. Pada Black Box Testing, penguji tidak perlu mengetahui bagaimana cara kerja sistem didalamnya, yang terpenting adalah apakah sistemnya bekerja sesuai dengan spesifikasi dan kebutuhan pengguna.

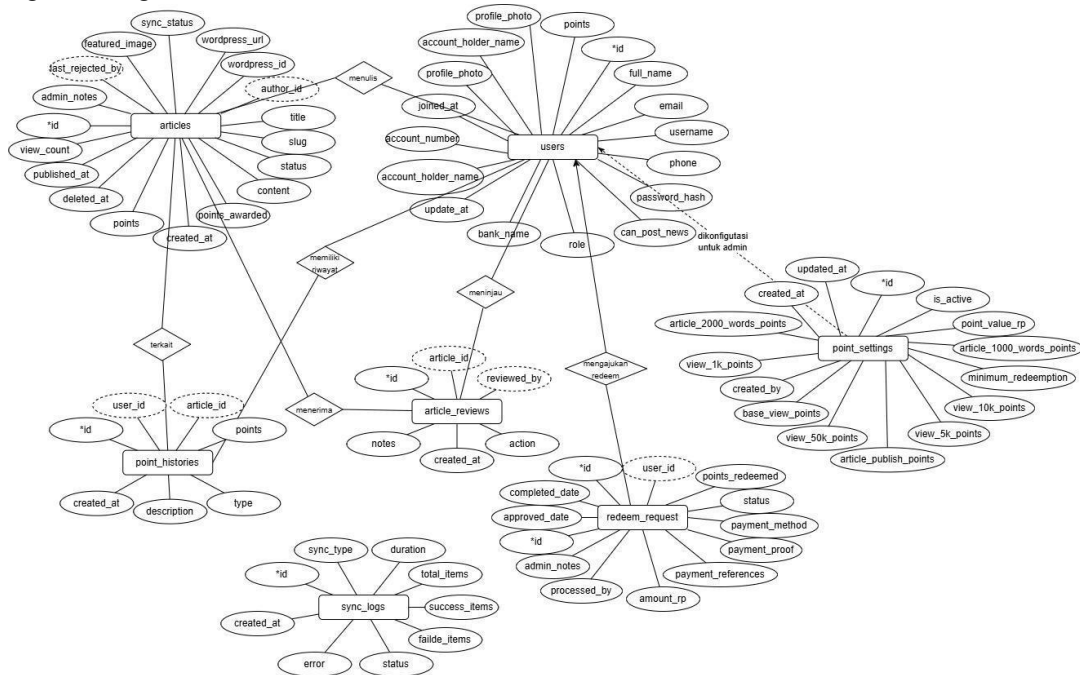
Teknik utama yang digunakan oleh Black Box Testing meliputi *equivalence partitioning*, *boundary value analysis*, *decision table testing*, *state transition testing*, dan *use case testing*. Kelebihan utama dari testing ini adalah, penguji dapat berfokus pada aspek fungsional sistem tanpa harus memahami internal atau struktur di baliknya, namun kekurangan metode ini adalah, kurang mampu untuk menemukan bug internalnya serta isu performa dan keamanan [14].

Pada aplikasi yang sedang dibangun ini, Black Box Testing digunakan secara menyeluruh mulai dari alur registrasi dengan OTP, autentikasi token, CRUD artikel, hingga mekanisme *redeem poin*. Hal ini untuk memastikan setiap endpoint berfungsi sesuai dengan kebutuhan pengguna tanpa perlu mengakses logika internal. Pendekatan ini menjamin bahwa sistem tidak hanya berjalan, namun juga berfungsi sebagaimana yang diharapkan oleh user.

3.7. Perancangan

1. Entity Relationship Diagram (ERD)

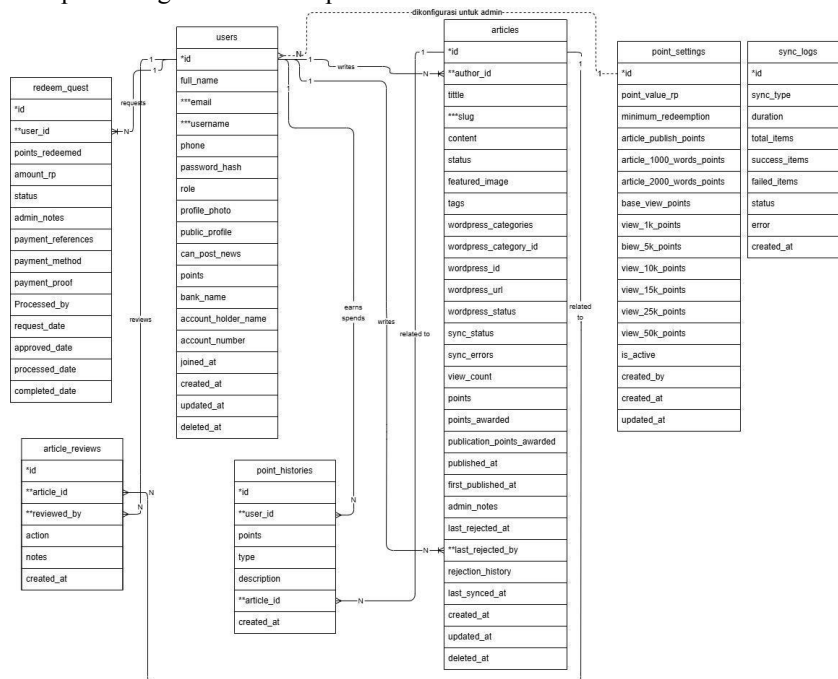
Entity Relationship Diagram (ERD) merupakan representasi visual dari struktur basis data yang menggambarkan entitas, atribut, serta hubungan antar entitas dalam sistem. Berikut adalah ERD yang dirancang.



Gambar. 3. Entity Relationship Diagram

2. Logical Record Structure (LRS)

Logical Record Structure merupakan hasil transformasi dari ERD yang menggambarkan struktur tabel beserta relasi antar tabel secara logis dalam basis data. Berikut adalah LRS yang digunakan dalam perancangan basis data aplikasi:

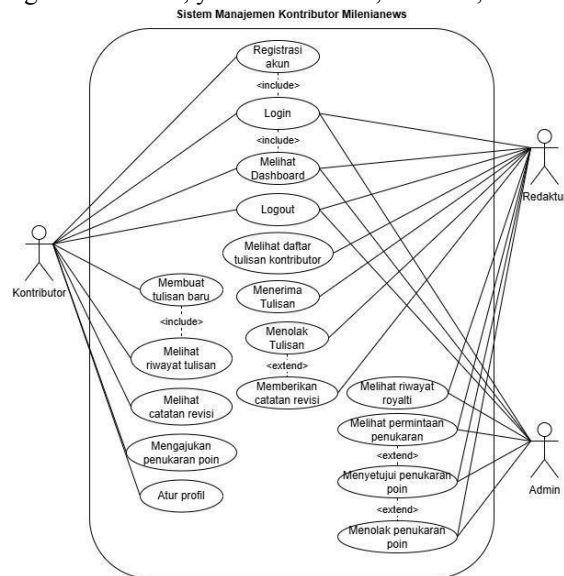


Gambar. 4. Logical Record Structure

3. Use Case Diagram

Use Case Diagram merupakan diagram UML yang menggambarkan interaksi antara pengguna (actor) dengan sistem, serta fungsionalitas yang tersedia bagi setiap pengguna. Berikut adalah Use Case

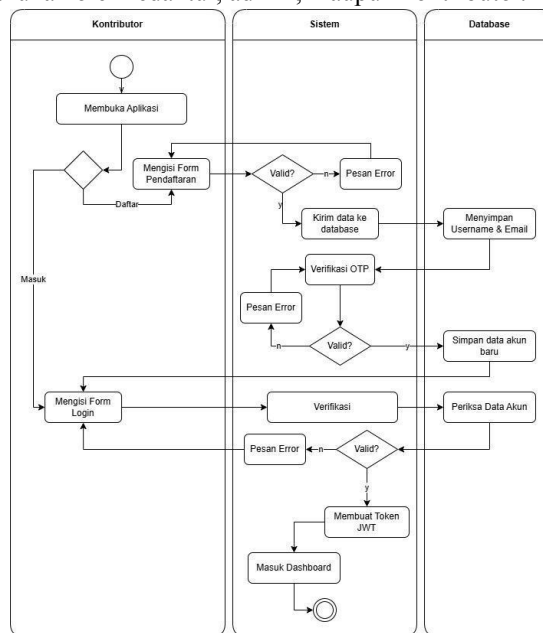
Diagram yang mencakup tiga aktor utama, yaitu kontributor, redaktur, dan admin:



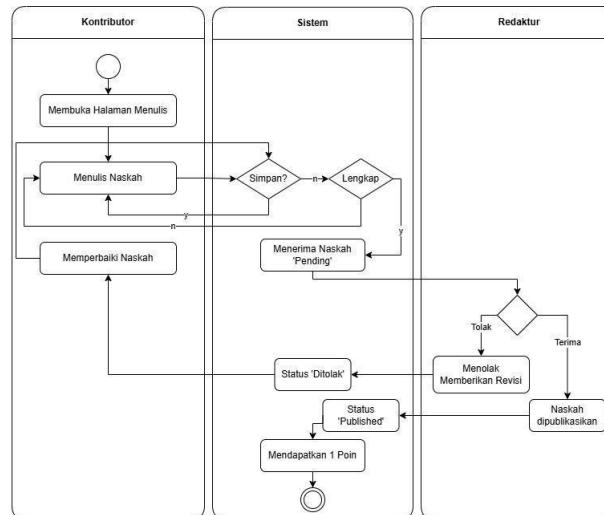
Gambar. 5. Use Case Diagram

4. Activity Diagram

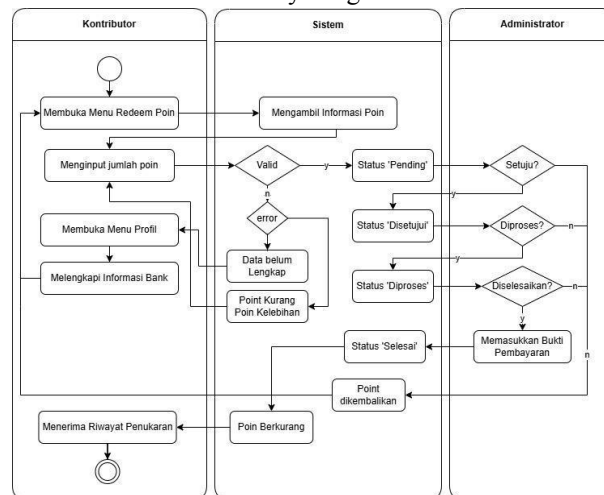
Activity Diagram merupakan diagram UML yang menggambarkan alur aktivitas atau proses yang terjadi di dalam sistem secara berurutan. Berikut adalah *Activity Diagram* yang menjelaskan alur setiap proses yang dapat dilakukan oleh redaktur, admin, maupun kontributor.



Gambar 6.. Activity Diagram Authentication



Gambar 7. Activity Diagram Submission



Gambar 8. Activity Diagram Redeem

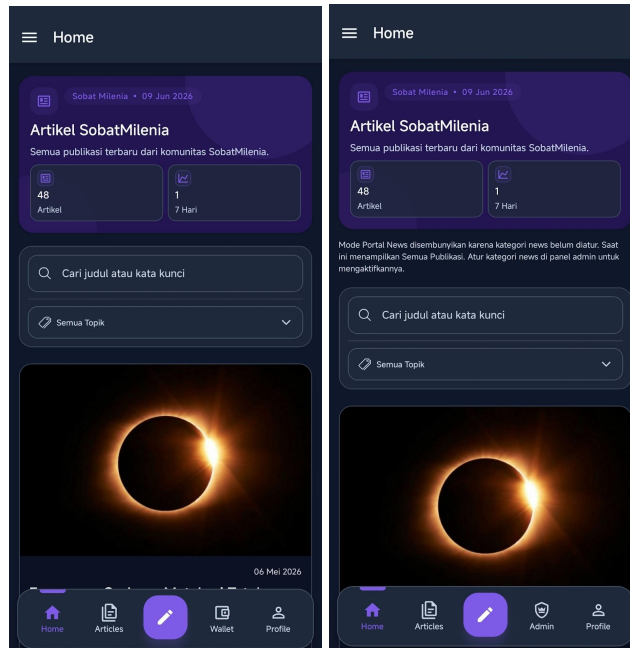
4. Hasil dan Pembahasan

4.1. Data yang Dikumpulkan :

Aplikasi manajemen kontributor MileniaNews dibuat dalam bentuk aplikasi *mobile* berbasis Android dengan menggunakan framework React native pada sisi *frontend* dan *framework* Laravel pada sisi *backend*. Metode pengembangan yang digunakan adalah metode Hybrid yang menggabungkan Waterfall dan Agile, serta diuji menggunakan metode Black Box Testing.

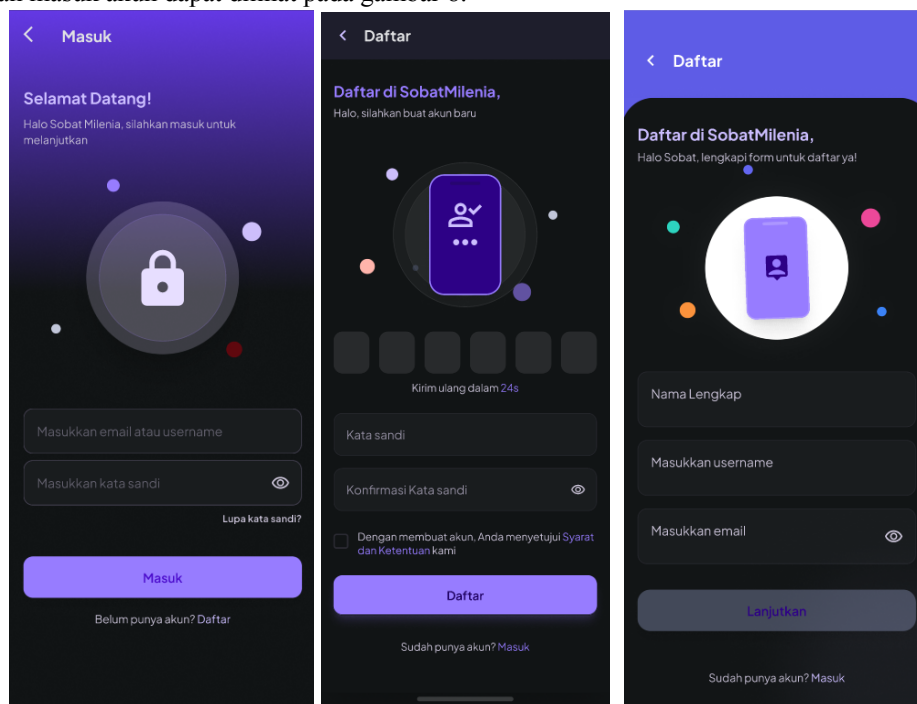
1. Hasil

Hasil yang diperoleh dari penelitian ini adalah sebuah aplikasi *mobile* manajemen kontributor bernama Sobat Milenia. Berikut adalah tampilan dari setiap halaman utama aplikasi yang telah berhasil dikembangkan. Saat pertama kali membuka aplikasi akan diarahkan ke halaman *dashboard* aplikasi. Halaman *dashboard* aplikasi dapat dilihat pada Gambar 7.



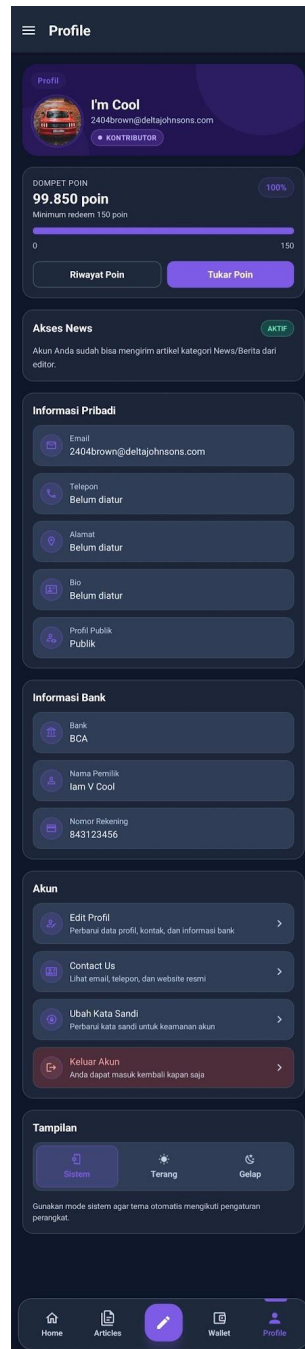
Gambar 9. Halaman Dashboard

Pengguna akan diarahkan untuk registrasi akun atau memasukkan akun saat mencoba untuk mengakses halaman daftar artikel, halaman menulis, halaman *wallet*, dan halaman profile. Halaman registrasi dan masuk akun dapat dilihat pada gambar 8.



Gambar 10. Halaman Autentikasi

Kontributor yang telah melakukan proses *login* dapat mengakses setiap halaman yang disediakan. Salah satunya halaman profil, pada halaman ini pengguna dapat melengkapi dan mengatur data diri. Halaman profil pengguna dapat dilihat pada gambar 9.



Gambar 11. Halaman Profile

Halaman *wallet* adalah halaman yang menampilkan jumlah poin, riwayat penukaran poin, dan akses untuk mengajukan penukaran poin. Halaman *wallet* dapat dilihat pada gambar 10.



Gambar 12. Halaman Wallet

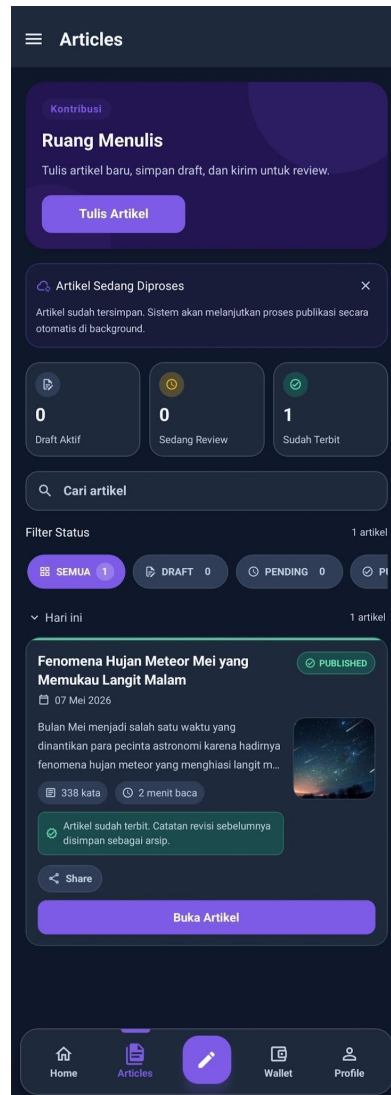
Halaman menulis dengan ikon pensil adalah halaman tempat kontributor menulis naskah, mengupload gambar, memilih kategori, dan memasukkan *tag*. Kontributor dapat menyimpannya sebagai *draft* atau langsung mengirim untuk direview. Halaman menulis dapat dilihat pada gambar 11.

The screenshot shows a mobile application interface for writing a new article. The form is titled 'Tulis Artikel Baru' (Write New Article) and includes the following sections:

- Editor:** A section at the top with the title 'Tulis Artikel Baru' and a subtitle 'Tulis dengan nyaman, rapikan isi artikelnya, lalu kirim saat sudah siap.' (Write comfortably, tidy up the article content, then send when ready). Below this are three input fields: 'Judul' (Title) with '0' characters, 'Kata' (Words) with '0' characters, and 'Kategori' (Category) with '0' characters.
- Detail Artikel:** A section with a 'Judul' label and a text input field containing 'Judul artikel'. Below the input field is a note: 'Tambahkan kategori, tag, dan mode News dari tombol Pengaturan Artikel di bawah.' (Add category, tag, and mode News from the Article Settings button below).
- Featured Image:** A section with a 'Featured Image' label and a large image placeholder. Below the placeholder is a note: 'Pilih gambar utama' (Select main image).
- Isi Artikel:** A section with a 'Isi Artikel' label and a rich text editor. The editor has a toolbar with icons for undo, redo, bold, italic, underline, heading 1, heading 2, list, and link. Below the toolbar is a large text area with the placeholder text 'Mulai menulis isi artikel...' (Start writing the article content...). Below the text area is a note: 'Pakai toolbar buat rapikan tulisan. Baca ulang dulu biar alurnya enak dibaca.' (Use the toolbar to tidy up the writing. Read it back first so the flow is easy to read). Below this is another note: 'Untuk kirim review, tulis minimal 300 kata.' (To send for review, write at least 300 words).
- Pengaturan Artikel:** A section with a 'Pengaturan Artikel' label and a note: 'kategori belum dipilih • tag opsional' (category not selected • tag optional). Below this is a note: 'Akses News belum aktif' (News access not active). Below this is a note: 'Pilih minimal 1 kategori sebelum kirim review.' (Select at least 1 category before sending for review). At the bottom of this section are two buttons: 'Simpan dulu' (Save first) and 'Kirim ke Review' (Send to Review).

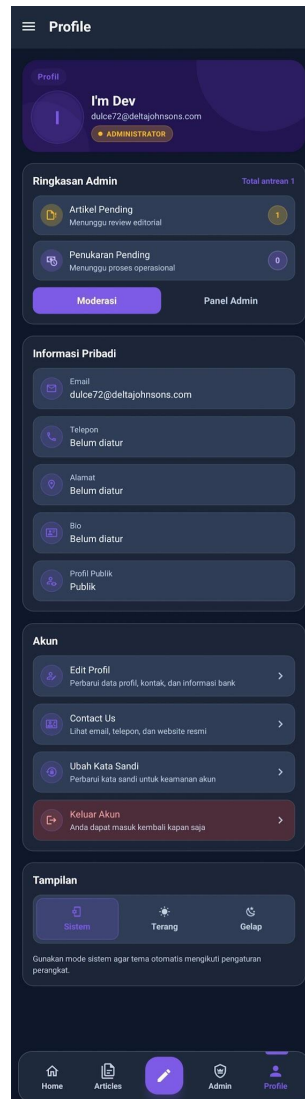
Gambar 13. Halaman Menulis Naskah

Halaman artikel adalah halaman yang menampilkan seluruh naskah yang sudah dibuat, seluruh tulisan dipisahkan berdasarkan statusnya. Halaman artikel dapat dilihat pada gambar 12.



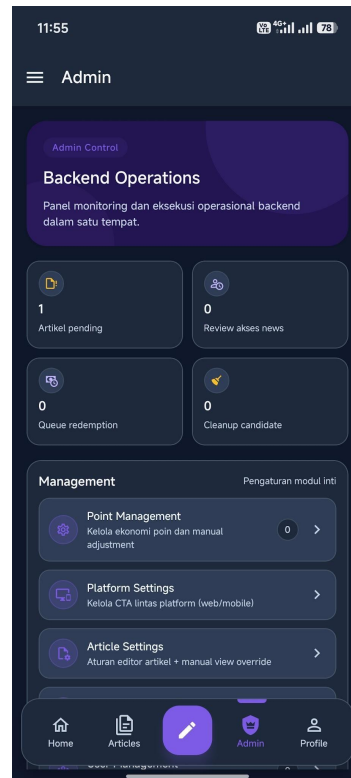
Gambar 14. Halaman Artikel

Pada sisi redaktur dan administrator memiliki akses halaman yang sama, setelah melakukan proses login juga dapat mengakses setiap halaman yang disediakan khusus untuk *role* admin redaktur. Pada halaman profil pengguna dapat melengkapi dan mengatur data diri. Halaman profil admin redaktur dapat dilihat pada gambar 13.



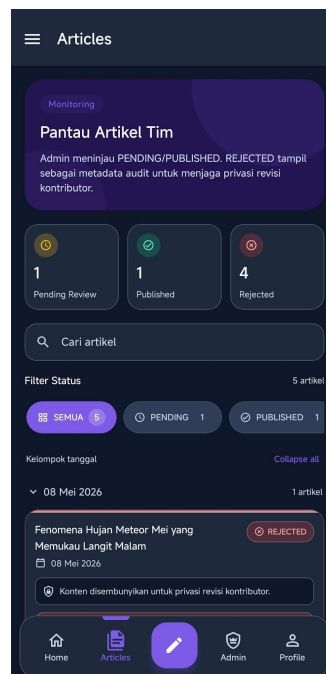
Gambar 15. Halaman Profile Admin Redaktur

Halaman admin adalah halaman yang مخصوص for admin untuk mengubah atau menetapkan pengaturan yang nantinya akan secara langsung diterapkan pada setiap akun kontributor. Halaman admin dapat dilihat pada gambar 14.



Gambar 16. Halaman Admin

Halaman artikel untuk admin redaktur adalah halaman yang menampilkan naskah yang telah dikirim oleh para kontributor, naskah ditampilkan berdasarkan statusnya. Halaman artikel admin redaktur dapat dilihat pada gambar 15.



Gambar 17. Halaman Artikel Admin Redaktur

2. Pengujian

Pengujian aplikasi dilakukan dengan menggunakan metode Black Box Testing yang berfungsi untuk memastikan bahwa seluruh fitur pada aplikasi Sobat Milenia berjalan sesuai spesifikasi yang telah ditetapkan dan bebas dari *error*. Pengujian dilakukan terhadap empat kelompok fitur utama, berikut adalah ringkasan hasil pengujiannya dapat dilihat pada tabel 1.

5. Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan aplikasi *mobile* manajemen kontributor MileniaNews berbasis Android menggunakan React Native sebagai *frontend* dan Laravel sebagai *backend* REST API. Aplikasi mengintegrasikan seluruh alur kerja editorial dari pengajuan naskah hingga pengelolaan royalti dalam satu sistem yang terpadu, tanpa mengganggu kebiasaan kerja redaktur di platform WordPress melalui mekanisme *webhook* dan *polling fallback*.

Metode pengembangan Hybrid yang menggabungkan Waterfall dan Agile terbukti efektif untuk proyek ini, memungkinkan perencanaan kebutuhan yang terstruktur sekaligus fleksibilitas implementasi yang responsif terhadap masukan stakeholder. Empat *sprint* pengembangan menghasilkan fitur autentikasi JWT, penulisan dan pengajuan naskah, sinkronisasi status *real-time* dari WordPress, serta sistem royalti berbasis poin yang terotomatisasi.

Pengujian Black Box Testing terhadap 32 skenario uji menunjukkan seluruh fitur berjalan valid sesuai spesifikasi. Sistem ini secara nyata meningkatkan efisiensi alur kerja redaksi, menyediakan transparansi status naskah bagi kontributor secara *real-time*, dan memastikan akurasi pencatatan royalti tanpa bergantung pada laporan mandiri. Untuk pengembangan lanjutan, disarankan mengintegrasikan Firebase Cloud Messaging untuk notifikasi push instan serta memperluas platform ke iOS memanfaatkan kemampuan lintas platform React Native.

Daftar Pustaka

- [1] T. R. Anugrafianto, "Kemajuan Teknologi Digital dan Preferensi Masyarakat terhadap Media Digital sebagai Sumber Berita," 2023.
- [2] T. Sjafrizal, P. Muhammad, A. Sulistyanto, and M. Madonna, "Transformasi Digital dalam Dunia Jurnalistik: Efisiensi dan Tantangan Editorial," 2025.
- [3] Mutakin, Kurniawan, and Purnama, "Peran Citizen Journalism dalam Memperluas Variasi Konten Berita: Studi Kasus Kanal Citizen6 Liputan6," 2025.
- [4] Fanani, "Pengelolaan Media Komunitas dalam Mendukung Jurnanisme Warga: Studi Kasus nusidoarjo.or.id," 2024.
- [5] Andera et al., "Peran Content Creator dan Strategi Media Sosial dalam Keberlanjutan Konten Digital," 2025.
- [6] P. Ackermann, Full Stack Web Development: The Comprehensive Guide, Bonn, Germany: Rheinwerk Publishing, 2023.
- [7] W. E. Susanto and A. Syukron, Logika & Algoritma untuk Pemula. 2020.
- [8] A. Osterwalder and Y. Pigneur, Business Model Generation, Hoboken, NJ, USA: John Wiley & Sons, 2010.
- [9] A. Wardhana, Kewirausahaan dan UMKM di Era Digital 4.0. 2023.
- [10] M. I. Alifudin, "Analisis dan Perancangan Sistem Informasi Manajemen Tugas Multi-Peran Berbasis Web," 2025.
- [11] Z. Ghinafikar, M. M. Mu'thy, and M. A. Yaqin, "Perbandingan Metode Agile dan Waterfall Berdasarkan Analisis Waktu Pengembangan Sistem," 2025.
- [12] M. Z. Ramadhan and F. Angelia, "Mengoptimalkan Pengembangan Aplikasi Mobile Melalui Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, Mobile-D, Agile, RAD)," 2023.
- [13] Herfansy, M. Julkarnain, and M. Hanif, "Desain dan Implementasi RESTful Web Service untuk Integrasi Data dan Aplikasi," 2022.
- [14] P. Leloudas, Introduction to Software Testing, Birmingham, UK: Packt Publishing, 2023.
- [15] R. S. Pressman and B. R. Maxime, Software Engineering: A Practitioner's Approach, 9th ed., New York, NY, USA: McGraw-Hill Education, 2019.
- [16] Permana, "WordPress sebagai Content Management System (CMS) untuk Pengelolaan Konten Website," 2024.