

Perancangan dan Implementasi Aplikasi Web Berita Online Menggunakan Framework Next Js

Zoulvia Hanest Khinanti^{*#1}, Bintang Lailatul Mukaromah^{#2}, Indyah Hartami Santi^{#3}
^{#123}Teknik Informatika, Universitas Islam Balitar, Blitar, Indonesia
Korespondensi author ^{*}zlvhnstkhnt@gmail.com

Info Artikel

Diajukan: 28 Mei 2025
Diterima: 1 Juli 2025
Diterbitkan: 9 Juli 2025

Keywords:
News Blog; Waterfall Method;
Next.Js; JWT; MongoDB; Website

Kata Kunci:
Blog Berita; Metode Waterfall;
Next.Js; JWT; MongoDB; Website



Lisensi: cc-by-sa

Copyright © 2025 Zoulvia Hanest Khinanti,
Bintang Lailatul Mukaromah, Indyah Hartami
Santi

Abstract

Online news websites are one form of utilization of digital technology that continues to develop, along with the increasing need of the community for fast, accurate, and easily accessible information. To support these needs, a web-based system was developed by implementing modern technologies such as API, Google authentication, JSON Web Token (JWT), Next.js, Tailwind CSS, and MongoDB. API is used as a link between the user interface and the backend system in managing news data and user accounts. Google authentication simplifies the login process without manual registration, while JWT is used to maintain security during data exchange. Next.js is applied to improve performance through server-side rendering, while Tailwind CSS helps create a responsive and consistent interface. MongoDB functions as a flexible database to store information efficiently. This system is designed with a component-based approach to be more structured and easy to develop. The final results show that the news website that was built is able to provide a fast, safe, and visually appealing user experience, and is worthy of being used as a modern and efficient digital information platform.

Abstrak

Web berita online merupakan salah satu bentuk pemanfaatan teknologi digital yang terus berkembang, seiring dengan meningkatnya kebutuhan masyarakat terhadap informasi yang cepat, akurat, dan mudah diakses. Untuk mendukung kebutuhan tersebut, dikembangkan sebuah sistem berbasis web dengan menerapkan teknologi modern seperti API, autentikasi Google, JSON Web Token (JWT), Next.js, Tailwind CSS, dan MongoDB. API digunakan sebagai penghubung antara antarmuka pengguna dan sistem backend dalam pengelolaan data berita serta akun pengguna. Autentikasi Google mempermudah proses login tanpa registrasi manual, sedangkan JWT digunakan untuk menjaga keamanan saat pertukaran data. Next.js diterapkan untuk meningkatkan performa melalui server-side rendering, sementara Tailwind CSS membantu menciptakan antarmuka yang responsif dan konsisten. MongoDB berfungsi sebagai basis data fleksibel untuk menyimpan informasi secara efisien. Sistem ini dirancang dengan pendekatan berbasis komponen agar lebih terstruktur dan mudah dikembangkan. Hasil akhir menunjukkan bahwa web berita yang dibangun mampu memberikan pengalaman pengguna yang cepat, aman, dan menarik secara visual, serta layak digunakan sebagai platform informasi digital yang modern dan efisien.

Cara mensitasi artikel:

Z. H. Khinanti, B. L. Mukaromah, I. H. Santi. "Sistem Informasi Wedding Organizer VK Project Berbasis Web Menggunakan Metode Prototype." *Jurnal Teknologi Informasi: Teori, Konsep, dan Implementasi (JTI-TKI)*, vol. 16, no. 2, pp. 65-71, Oktober 2025, <https://doi.org/10.36382/jti-tki.v16i2.578>

PENDAHULUAN

Kemajuan teknologi informasi telah memberikan dampak signifikan terhadap berbagai aspek kehidupan, termasuk dalam bidang publikasi dan penyebaran informasi digital. Masyarakat saat ini membutuhkan media berbasis web yang tidak hanya cepat dan mudah diakses, tetapi juga aman serta mampu memberikan kemudahan dalam pengelolaan konten. Namun, tidak semua komunitas atau individu memiliki kemampuan teknis maupun akses terhadap platform *blog* yang memadai. Permasalahan seperti kompleksitas pengelolaan sistem, keterbatasan dalam desain antarmuka, serta kerentanan terhadap isu keamanan menjadi hambatan umum dalam proses digitalisasi publikasi informasi.

Subyek pengabdian dalam kegiatan ini adalah kelompok masyarakat kreatif dan individu penggiat literasi digital yang belum memiliki akses terhadap platform *blog* yang terintegrasi dengan sistem keamanan modern dan

tampilan yang responsif di mana tampilan dan tata letak halaman web menyesuaikan secara otomatis sesuai dengan ukuran layar atau perangkat yang digunakan [1] Berdasarkan hasil observasi, ditemukan bahwa banyak pelaku literasi atau usaha mikro yang membutuhkan media digital untuk publikasi tetapi masih menggunakan platform yang terbatas, atau bahkan belum menggunakan media daring sama sekali.

Isu utama yang diangkat dalam kegiatan ini adalah kebutuhan akan media publikasi *blog* yang sederhana namun lengkap secara fungsional. Fokus pengabdian diarahkan pada pengembangan dan penerapan aplikasi *blog* berbasis web yang memanfaatkan integrasi teknologi modern seperti Next.js, autentikasi Google, JSON Web Token (JWT), dan Tailwind CSS. Penerapan teknologi ini diharapkan mampu memberikan kemudahan penggunaan, keamanan data, serta efisiensi dalam pengelolaan konten.

Tujuan dari kegiatan ini adalah untuk mengimplementasikan aplikasi *blog* yang dapat digunakan oleh masyarakat umum secara praktis dan aman, sekaligus

dapat dimanfaatkan sebagai media alternatif untuk mendorong peningkatan kemampuan berpikir, memperluas wawasan, serta melatih literasi melalui tulisan. Selain itu, *blog* turut mendukung pengembangan kreativitas individu dalam menuangkan ide-ide secara tertulis. [2]. Kegiatan pengabdian ini merupakan bagian dari hilirisasi hasil penelitian dan pengembangan sistem berbasis API dan *framework* modern yang sebelumnya telah diuji secara fungsional.

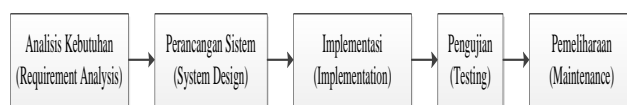
Dukungan literatur menunjukkan bahwa penggunaan JWT dapat meningkatkan keamanan data dalam sistem berbasis web karena proses autentikasi dan otorisasi dilakukan secara terenkripsi [3]. *Framework* Next.js terbukti mampu meningkatkan performa aplikasi melalui *server-side rendering* dan pengoptimalan SEO [4], sementara Tailwind CSS banyak digunakan untuk mempercepat proses pengembangan antarmuka tanpa mengorbankan fleksibilitas desain. Selain itu, autentikasi Google OAuth 2.0 telah diadopsi secara luas sebagai metode login yang lebih cepat dan aman [5].

Melalui kegiatan ini, diharapkan terjadi perubahan sosial berupa peningkatan kemampuan masyarakat dalam mengelola media digital secara mandiri dan aman. Potensi komunitas yang aktif dalam literasi digital menjadi landasan kuat untuk keberlanjutan pemanfaatan aplikasi ini. Selain itu, penggunaan perangkat lunak open source mendukung terciptanya solusi yang berkelanjutan dan dapat disesuaikan sesuai kebutuhan komunitas.

METODE

Penelitian ini menggunakan metode pengembangan perangkat lunak *Waterfall*. Model *Waterfall* adalah salah satu metode dalam *Software Development Life Cycle* (SDLC) yang menggunakan pendekatan linear dan berurutan. Proses pengembangan dilakukan tahap demi tahap, mulai dari perencanaan, analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan. Metode ini juga dikenal sebagai *Linear Sequential Model* dan pertama kali diperkenalkan oleh Winston Royce pada tahun 1970. Meski terbilang klasik, model ini masih sering digunakan karena alurnya yang jelas dan dokumentasinya yang lengkap [6].

Metode ini dipilih karena sesuai untuk pengembangan sistem dengan kebutuhan yang sudah jelas dan minim perubahan. Alur kerja yang terstruktur memudahkan pengendalian dan pemantauan proses, serta memungkinkan evaluasi setiap tahap sebelum lanjut ke tahap berikutnya, sehingga hasil akhir lebih terjamin. Tahapan dari metode *waterfall* seperti Gambar 1. berikut.



Gambar 1 Metode *Waterfall*

Tahapan metode *Waterfall* yang diterapkan

A. Analisis Kebutuhan (*Requirement Analysis*)

Tahap ini bertujuan untuk mengidentifikasi serta merumuskan kebutuhan sistem berdasarkan hasil observasi dan diskusi dengan pihak-pihak terkait. Informasi yang dikumpulkan mencakup fitur utama yang dibutuhkan dalam aplikasi, seperti pengelolaan konten berita oleh admin, tampilan berita untuk pengguna, serta aspek tambahan seperti keamanan, kemudahan akses, dan kecepatan sistem. Hasil analisis ini kemudian disusun dalam bentuk dokumen spesifikasi kebutuhan yang akan menjadi acuan dalam tahap perancangan sistem.

B. Perancangan Sistem (*System Design*)

Setelah kebutuhan sistem terdefinisi, tahap selanjutnya adalah merancang struktur sistem secara menyeluruh. Perancangan ini mencakup desain antarmuka pengguna (*user interface*), struktur basis data, alur navigasi, serta arsitektur sistem yang akan digunakan. Desain ini bertujuan untuk menggambarkan bagaimana sistem akan berjalan dan berinteraksi dengan pengguna. Hasil dari tahap ini akan menjadi pedoman utama dalam proses implementasi.

C. Implementasi (*Implementation*)

Pada tahap ini, desain sistem yang telah dibuat diterjemahkan ke dalam bentuk kode program. Aplikasi dikembangkan menggunakan *framework* Next.js dengan bahasa pemrograman JavaScript. Next.js dipilih karena kemampuannya dalam mendukung *server-side rendering*, yang memberikan keunggulan dalam performa dan optimasi mesin pencari (SEO). Selain itu, *framework* Tailwind CSS digunakan untuk membangun antarmuka yang responsif dan konsisten di berbagai ukuran layar.

D. Pengujian (*Testing*)

Setelah proses implementasi selesai, dilakukan pengujian terhadap seluruh fitur sistem untuk memastikan bahwa aplikasi berjalan sesuai dengan spesifikasi yang telah ditentukan. Pengujian dilakukan pada berbagai aspek, seperti fungsionalitas fitur, tampilan antarmuka, kompatibilitas di berbagai perangkat, dan stabilitas sistem secara keseluruhan. Tujuan utama dari tahap ini adalah untuk menemukan dan memperbaiki kesalahan (*bug*) sebelum sistem digunakan secara luas oleh pengguna.

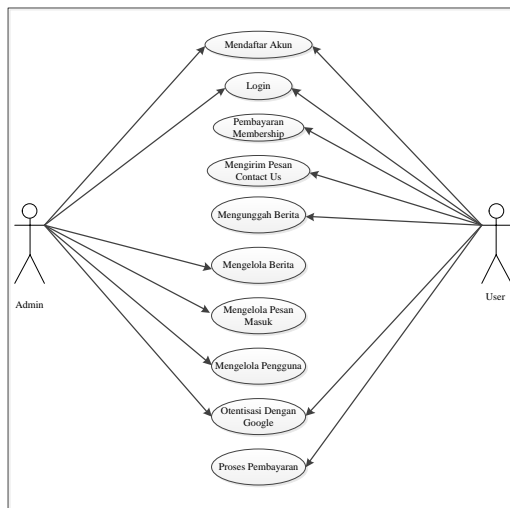
E. Pemeliharaan (*Maintenance*)

Tahap pemeliharaan dilakukan setelah sistem mulai digunakan secara aktif oleh pengguna. Kegiatan pada tahap ini mencakup perbaikan kesalahan yang muncul saat penggunaan, pembaruan sistem sesuai kebutuhan baru, serta peningkatan performa agar sistem tetap optimal. Pemeliharaan merupakan proses berkelanjutan yang bertujuan untuk memastikan sistem tetap berjalan dengan baik dan relevan dengan perkembangan kebutuhan pengguna maupun teknologi.

F. Unified Modeling Language (UML)

UML merupakan suatu teknik untuk memodelkan sistem. Pengertian lainnya, UML adalah seperangkat aturan dan notasi untuk spesifikasi sistem software. Notasi ini menyediakan satu set elemen grafis untuk pemodelan sistem. Perancangan dan pembangunan aplikasi atau software berbasis objek atau *Object Oriented Analysis and Design* (OOAD) menganggap segala sesuatunya adalah objek serta sistem dipandang sebagai interaksi dari banyak objek yang dimodelkan menggunakan UML. UML versi terbaru terdiri dari lima belas diagram yang dibagi menjadi dua kelompok, yaitu *structure* dan *behaviour diagram*. *Structure diagram* menggambarkan data dan hubungan statis dalam suatu sistem informasi, sedangkan *behaviour diagram* menggambarkan hubungan dinamis antara objek yang mewakili sistem informasi [7].

1) **Use case diagram:** Diagram *use case* adalah diagram UML yang menggambarkan apa yang dapat dilakukan aktor dalam suatu sistem. *Use case* mendeskripsikan tugas tertentu, sedangkan aktor dalam *use case diagram* adalah unit atau kelompok unit yang berinteraksi dengan sistem. untuk melakukan pekerjaan tertentu [8]. Gambar 2 berikut diagram *use case* dari sistem.



Gambar 2. Use Case Diagram

a) **Admin:** Admin merupakan aktor utama yang memiliki kendali penuh terhadap sistem aplikasi web berita online. Admin tidak perlu melakukan proses pendaftaran karena akun telah disediakan langsung oleh pengelola sistem. Setelah login, admin dapat mengelola seluruh konten berita, termasuk menambahkan, mengedit, dan menghapus berita yang ditampilkan kepada pengguna.

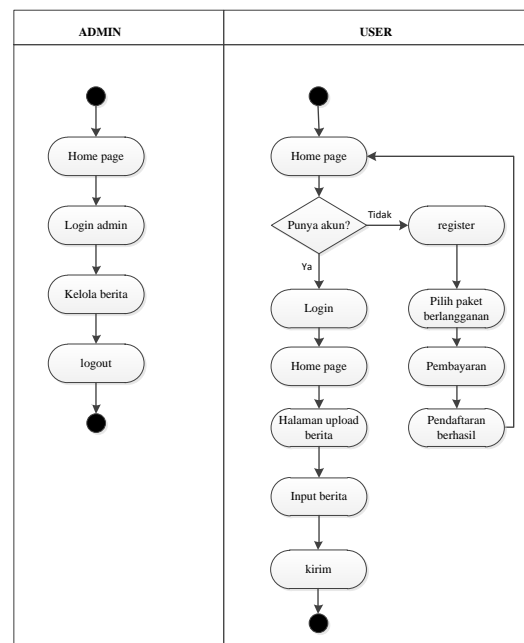
Selain itu, admin juga bertanggung jawab dalam menangani pesan dari pengguna yang masuk melalui fitur *Contact Us*. Admin dapat membaca, memfilter, dan memberikan tanggapan terhadap pesan tersebut sesuai kebutuhan. Admin juga memiliki wewenang untuk mengelola data pengguna, seperti melakukan verifikasi akun atau menghapus akun yang tidak valid.

Untuk kemudahan akses, sistem menyediakan opsi login menggunakan akun Google. Admin juga memantau dan memverifikasi proses pembayaran yang dilakukan oleh pengguna, khususnya untuk layanan *membership*. Secara keseluruhan, admin berperan penting dalam menjaga kelancaran operasional, kualitas konten, dan keamanan sistem.

b) **User:** *User* adalah aktor yang berperan sebagai pengguna umum dari aplikasi berita online. Pengguna dapat melakukan pendaftaran dan login untuk mengakses fitur-fitur yang tersedia. Setelah login, user dapat membaca berita, mengirim pesan melalui fitur *Contact Us*, serta mengunggah berita jika sistem mengizinkan kontribusi dari pengguna.

User juga memiliki akses untuk melakukan pembayaran *membership* guna menikmati fitur tambahan atau konten premium. Untuk mempermudah proses login, pengguna dapat menggunakan akun Google melalui fitur autentikasi pihak ketiga. Dengan demikian, user tidak hanya berperan sebagai pembaca, tetapi juga dapat berkontribusi dan berinteraksi secara aktif dalam sistem.

2) **Diagram activity:** Diagram *activity* adalah gambaran aliran kerja atau aktivitas dari sebuah sistem atau menu yang ada pada perangkat lunak. Diagram ini digunakan untuk memvisualisasikan proses bisnis atau alur sistem secara rinci, mulai dari awal hingga akhir proses. Setiap langkah dalam sistem digambarkan sebagai aktivitas, dengan hubungan antar aktivitas ditunjukkan melalui panah. *Activity Diagram* memudahkan pengembang dan pemangku kepentingan dalam memahami alur proses, mengidentifikasi kebutuhan sistem, serta merancang alur kerja yang efisien dan sistematis. [4]



Gambar 3. Diagram Activity

a. Admin

Pada sistem ini, admin memulai proses dengan mengakses halaman utama. Selanjutnya, admin melakukan proses autentikasi melalui menu *login* untuk dapat masuk ke dalam sistem. Setelah berhasil *login*, admin diarahkan ke fitur kelola berita, yang berfungsi untuk melakukan manajemen konten seperti menambah, mengubah, atau menghapus berita yang tersedia. Setelah seluruh aktivitas pengelolaan selesai, admin mengakhiri sesi dengan melakukan *logout* sebagai bentuk keluar dari sistem secara aman dan terkontrol.

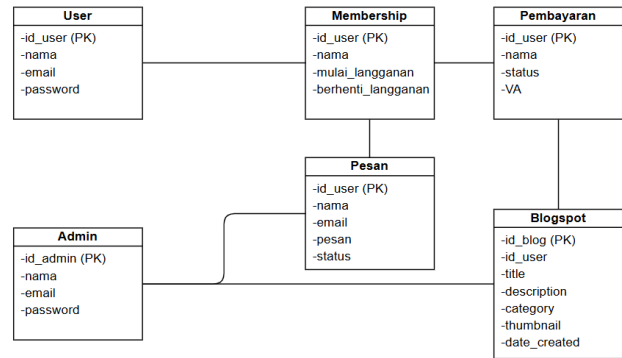
b. User

Pengguna (*user*) memulai aktivitasnya dengan mengakses halaman utama sistem. Sistem akan memverifikasi apakah pengguna telah memiliki akun. Apabila belum, maka pengguna akan diarahkan ke proses registrasi, kemudian memilih paket berlangganan yang tersedia, dan melanjutkan ke proses pembayaran. Setelah pembayaran berhasil, sistem akan menampilkan konfirmasi bahwa pendaftaran telah berhasil dan pengguna kembali ke halaman utama untuk melakukan *login*.

Jika pengguna telah memiliki akun sebelumnya, maka dapat langsung melakukan login ke dalam sistem. Setelah proses *login* berhasil, pengguna diarahkan ke halaman *upload* berita, di mana pengguna dapat melakukan pengisian data berita pada *form* yang disediakan. Setelah data berita selesai diisi, pengguna dapat mengirimkan konten tersebut ke dalam sistem. Proses ini menandai berakhirnya aktivitas pengguna dalam sistem.

3) **Class diagram : Class Diagram** merupakan suatu asosiasi dari arsitektur class pada suatu sistem. Diagram ini digunakan untuk menggambarkan struktur dan hubungan antar *class* dalam sistem, termasuk atribut, metode (fungsi), serta jenis relasi seperti inheritance, asosiasi, agregasi, dan komposisi.[9]. Melalui *class diagram*, pengembang dapat memahami bagaimana setiap komponen dalam sistem saling berinteraksi secara logis. Hal ini sangat membantu dalam proses perancangan, pengembangan, hingga pemeliharaan perangkat lunak. Selain itu, *class diagram* juga menjadi elemen penting dalam pemodelan sistem berbasis objek karena merepresentasikan rancangan dasar dari komponen-komponen yang akan dibangun. *Class Diagram* ditunjukkan oleh Gambar 4.

Class diagram pada system blog berita *online* ini menggambarkan hubungan antar entitas utama, yaitu: *user*, *admin*, *membership*, *pembayaran*, *pesan*, dan *blogspot*. Entitas *user* secara langsung terhubung dengan entitas *membership*, *pembayaran*, dan *pesan*. Hubungan ini menunjukkan bahwa seorang pengguna dapat memiliki data keanggotaan, melakukan transaksi pembayaran, serta mengirim pesan atau masukan. Entitas *membership* mencatat informasi paket berlangganan yang dipilih pengguna, sedangkan entitas *pembayaran* menyimpan detail transaksi pembayaran tersebut, termasuk status dan metode pembayaran.

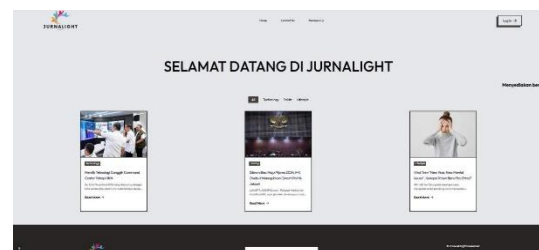


Gambar 4. Class Diagram

Sementara itu, admin terhubung ke pesan sebagai pihak yang bertugas menerima dan merespons masukan dari pengguna. Selain itu, admin juga berperan dalam pengelolaan konten artikel yang disimpan pada entitas *blogspot*. *Blogspot* berisi data artikel yang dibuat oleh pengguna, termasuk judul artikel, deskripsi, kategori, dan waktu publikasi. Seluruh relasi antar entitas menunjukkan keterkaitan fungsional antarkomponen system dalam mendukung operasi platform *blog* secara menyeluruh, baik dari sisi pengguna maupun pengelola.

HASIL DAN PEMBAHASAN

A. Tampilan Home



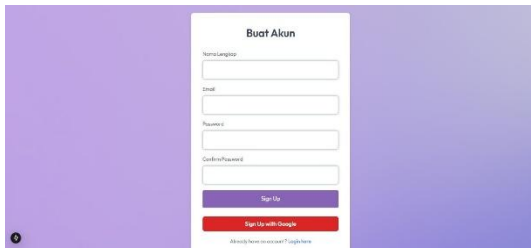
Gambar 5 Tampilan Home

Gambar 5 adalah beranda Jurnalight memiliki tampilan yang sederhana dan mudah dipahami. Di bagian atas, terdapat menu navigasi seperti *Home*, *Contact Us*, dan tombol *Login*. Tulisan besar "Selamat Datang Di Jurnalight" di tengah halaman memberikan kesan hangat bagi pengunjung. Artikel-artikel ditampilkan dengan gambar pendukung, deskripsi singkat, dan tombol "*Read More*" untuk membaca lebih lanjut. Kategori seperti *Technology*, *Politic*, dan *Lifestyle* mempermudah pencarian artikel, sementara bagian bawah halaman dilengkapi kolom berlangganan email serta informasi hak cipta.

B. Tampilan Sign Up

Gambar 6 adalah halaman pendaftaran akun dirancang sederhana dengan judul "Buat Akun" di bagian atas. Pengguna diinstruksikan untuk mengisi kolom Nama Lengkap, Email, Password, dan Konfirmasi Password. Setelah mengisi data, pengguna dapat memilih tombol *Sign Up* berwarna ungu untuk mendaftar langsung, atau tombol *Sign Up with Google* berwarna merah untuk mendaftar

melalui akun Google. Di bagian bawah halaman, terdapat tautan "*Login here*" bagi pengguna yang sudah memiliki akun.



Gambar 6 Halaman *Sign Up*

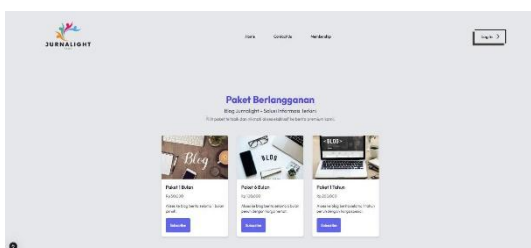
C. Tampilan Login



Gambar 7 Halaman *Login*

Halaman login ini memungkinkan pengguna untuk masuk dengan memasukkan Email dan Password atau menggunakan akun Google melalui tombol *Login* berwarna ungu dan *Login with Google* berwarna merah. Tautan "*Don't have an account? Sign up*" memberi opsi untuk mendaftar jika pengguna belum memiliki akun.

D. Tampilan Daftar Membership



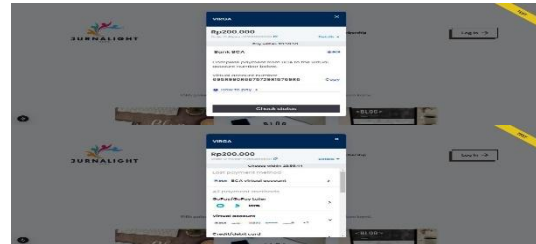
Gambar 8 Halaman Daftar *Membership*

Halaman ini menawarkan tiga pilihan paket langganan: Paket 1 Bulan, Paket 6 Bulan, dan Paket 1 Tahun. Setiap paket dilengkapi dengan informasi harga, deskripsi singkat mengenai durasi akses ke *blog* premium, serta tombol "*Subscribe*" berwarna biru yang memudahkan pengguna untuk mendaftar.

E. Tampilan Pembayaran

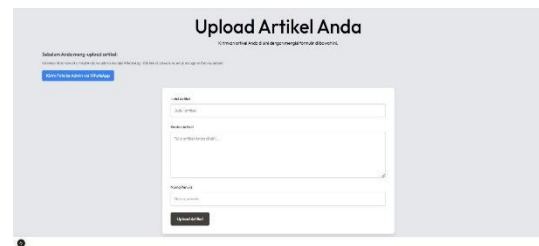
Halaman pembayaran ini dibuat untuk mempermudah pengguna yang sudah menjadi membership untuk melakukan transaksi ketika sudah memilih paket berlangganan yang diinginkan. Terdapat nama pengguna yang akan melakukan transaksi dibagian atas, dibawahnya

terdapat nominal yang akan dibayarkan, halaman ini akan menyediakan kode *Virtual Account* dan menyediakan beberapa alternatif pembayaran lainnya seperti Qris dan lainnya, dan pengguna juga bisa memilih dari bank komersial mana untuk melanjutkan transaksi.



Gambar 9 Halaman Pembayaran

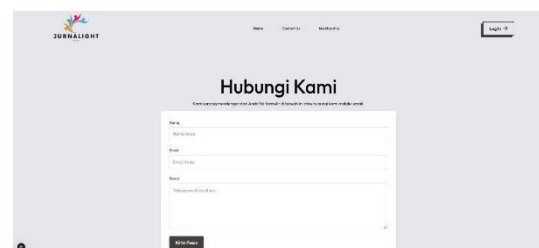
F. Tampilan Upload Berita



Gambar 10 Halaman *Upload* Berita

Di halaman upload berita untuk **user**, kami menyediakan tempat untuk pengguna mengirimkan artikel mereka ke admin dengan mengisi judul artikel, konten artikel, nama penulis, serta mengirim foto ke admin untuk ditampilkan bersama artikel yang mereka tulis. Lalu tersedia tombol *Upload Artikel* untuk mengirim artikel ke admin yang mana nantinya akan di konfirmasi oleh admin dan di *upload* admin ke laman web.

G. Tampilan Contact Us

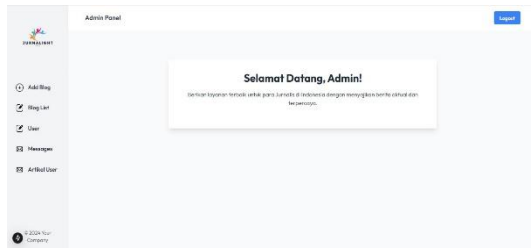


Gambar 11 Halaman *Contact Us*

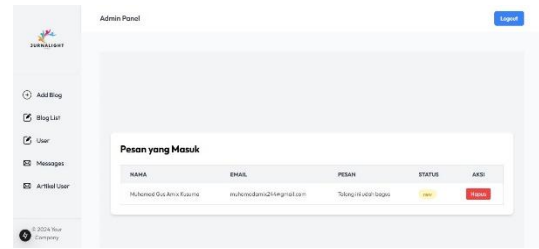
Halaman *Contact Us* menyediakan formulir yang harus diisi pengguna dengan memasukkan nama, email, dan pesan yang akan disampaikan. Lalu tersedia tombol Kirim Pesan untuk mengirim pesan dari pengguna kepada admin.

H. Tampilan Admin Panel

Halaman admin, terdapat tulisan *Selamat Datang, Admin!* untuk menyapa admin ketika *login*. Tersedia juga menu *add blog*, *blog list*, *user*, *messages*, dan artikel user.

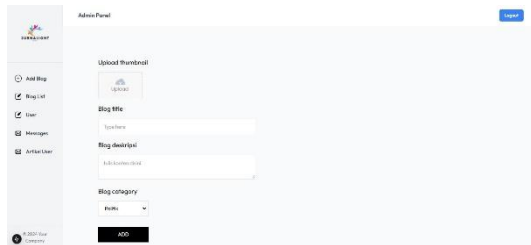


Gambar 12 Halaman Admin Panel



Gambar 15 Halaman Pesan

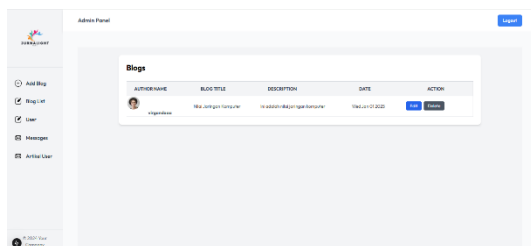
I. Tampilan Tambah Blog



Gambar 13 Halaman Tambah Blog

Untuk halaman *Add Blog* tersedia *upload thumbnail* yang mana merupakan tempat mengupload foto yang sudah dikirim user, berikutnya ada *blog title* untuk menuliskan judul artikel dari user, lalu ada *blog deskripsi* untuk menuliskan isi artikel yang sudah dikirim oleh user, tersedia juga *blog category* untuk memilih kategori artikel yang akan di upload ke website, yang terakhir adalah tombol *ADD* di bagian bawah halaman untuk mengupload artikel ke halaman homepage.

J. Tampilan Blog List



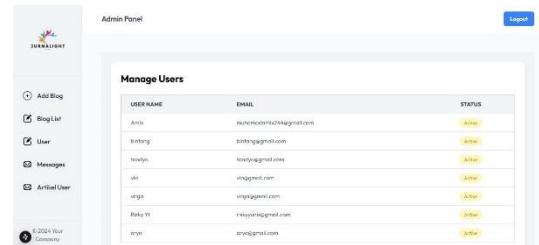
Gambar 14 Halaman Blog List

Halaman *blog list* berisi informasi nama pengirim artikel, judul artikel yang mereka kirim, deskripsi, dan tanggal pengiriman artikel yang kemudian akan di-review admin. Untuk bagian *action* ada dua pilihan, *edit* atau *delete* yang mana tombol *edit* digunakan ketika akan mengupload artikel setelah di-review, dan tombol *delete* digunakan untuk menghapus artikel ketika admin telah me-review artikel tapi tidak sesuai kriteria dalam pengiriman artikel.

K. Tampilan Pesan

Pada halaman *messages* tersedia informasi dari *user* yang mengirim pesan kepada admin. Tertera nama pengirim, email, pesan yang diketik user, status pesan, dan aksi yang terdapat tombol hapus.

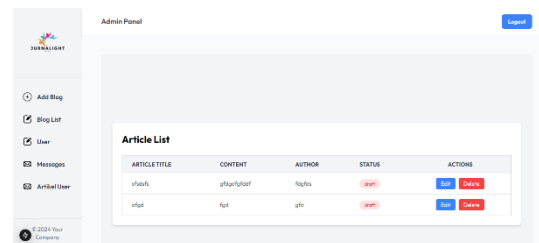
L. Tampilan Manage User



Gambar 16 Halaman Manage User

Halaman *user*, tertera nama nama dari *user* beserta email, dan ada status untuk melihat user tersebut berstatus *active* atau tidak.

M. Tampilan Artikel User



Gambar 17 Halaman Artikel User

Halaman *Artikel user* merupakan halaman admin mengelola artikel dari *user*, dimana *user* mengirimkan artikel pada halaman upload artikel, setelah itu masuk ke dalam admin artikel *user* disini admin mempunyai wewenang untuk mengedit dan mendelete artikel dan memberikan status di setuju atau ditolak artikelnnya

KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pembahasan, dapat disimpulkan bahwa pengembangan aplikasi web berita *online* menggunakan Next.js, API, Google *Login*, JWT, Tailwind CSS, dan MongoDB telah berhasil menghasilkan sebuah platform yang efisien, aman, dan responsif dalam menyajikan konten berita digital. Fitur Google *Login* memudahkan proses autentikasi pengguna dan meningkatkan keamanan. API digunakan untuk menghubungkan *frontend* dan *backend* agar pengelolaan data seperti artikel dan komentar berjalan dengan baik. MongoDB, sebagai basis data non-relasional, memberikan fleksibilitas dalam pengelolaan data yang bersifat dinamis.

Selain itu, Next.js mendukung performa tinggi melalui *server-side rendering* dan pengoptimalan SEO, sementara Tailwind CSS mempermudah pembuatan antarmuka pengguna yang modern dan responsif. Penggunaan JWT juga menjaga keamanan dalam proses pertukaran data antar sistem.

Untuk pengembangan ke depannya, sistem masih dapat ditingkatkan, terutama dengan mengoptimalkan kinerja API *backend* dan menerapkan *caching* guna mempercepat akses data, khususnya saat jumlah pengguna meningkat. Pengujian langsung bersama pengguna akhir juga disarankan untuk mengevaluasi pengalaman penggunaan aplikasi serta memperoleh masukan untuk perbaikan lebih lanjut. Selain itu, dokumentasi teknis yang lengkap mengenai struktur sistem dan proses pengembangan sangat penting disiapkan sebagai dasar dalam proses pemeliharaan dan pengembangan aplikasi secara berkelanjutan.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada semua pihak yang telah memberikan dukungan selama proses penyusunan jurnal ini. Terima kasih khusus kepada dosen pembimbing atas bimbingan dan arahannya, serta kepada keluarga, rekan-rekan, dan pihak-pihak lain yang turut membantu, baik secara langsung maupun tidak langsung. Semoga jurnal ini bermanfaat dan menjadi referensi untuk pengembangan system serupa di masa mendatang.

REFERENSI

- [1] C. R. Dhona, P. K. Handayani, Dan E. Darmanto, "Rancang Bangun Web Responsif Untuk Sistem Informasi Pelayanan Perbaikan Smartphone," *Indonesian Journal Of Technology, Informatics And Science (Ijtis)*, Vol. 4, No. 1, Hlm. 5–8, Des 2022.
- [2] F. S. Hamdi Dan I. Maíta, "Pelatihan Pembuatan Website Memanfaatkan Wix Untuk Blog Pribadi Pada Siswa Sman 2 Gunung Talang: Website Development Training Using Wix For Personal Blogs For Students Of Sman 2 Gunung Talang," *Consen*, Vol. 2, No. 2, Hlm. 64–69, Nov 2022.
- [3] Unggul Budi Astowo Dan Ari Sujarwo, "Penerapan Json Web Token Sebagai Strategi Pengamanan Data Pada Aplikasi Multimasjid," *Journal Of Social Science Research*, Vol. 3, No. 6, Hlm. 5279–5292, 2023.
- [4] R. A. Ananda Dan G. F. Nama, "Analisis Dan Perancangan Layanan Streaming Film Berbasis Web Langganan Menggunakan Framework Nextjs," *Jitet*, Vol. 12, No. 1, Jan 2024.
- [5] F. F. Azzahra, B. Rosa Sekamayang, Dan R. Habibi, "Integrasi Google Oauth Dan Payment Gateway Midtrans Dalam Kosconnect," *Processor*, Vol. 20, No. 1, Mei 2025, Doi: 10.33998/Processor.2025.20.1.2185.
- [6] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *Jurnal Ilmu-Ilmu Informatika Dan Manajemen Stmik*, 2020.
- [7] P. Maharani, "Pengembangan Website Pt. Rantangin Digital Indonesia Menggunakan Framework Next Js Dan Tailwind Css," *Repeater*, Vol. 3, No. 1, Hlm. 129–137, Jan 2025.
- [8] M. Alda Dan D. R. Sahendra, "Perancangan Sistem Informasi Inventaris Barang Berbasis Web Pada Dispora Sumut," *Simtek : Jurnal Sistem Informasi Dan Teknik Komputer*, Vol. 8, No. 1, Art. No. 1, Apr 2023, Doi: 10.51876/Simtek.V8i1.163.
- [9] G. K. Wardana, B. Rahayudi, Dan W. H. N. Putra, "Pengembangan E-Commerce Dengan Integrasi Api Payment Gateway Midtrans," *Urnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol. 5, No. 11, Hlm. 4770–4774, 2021.