

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama masa magang, diberikan tanggung jawab untuk mengembangkan sebuah situs berita menggunakan *framework* Laravel. Tugas ini mencakup seluruh proses pengembangan mulai dari perancangan struktur aplikasi, implementasi fitur utama seperti pembuatan, pengeditan, dan penghapusan berita, hingga integrasi dengan API untuk mengambil data berita secara dinamis.

Dalam pengerjaan proyek ini, tanggung jawab meliputi memastikan keamanan aplikasi, melakukan validasi *input* pengguna, serta mengoptimalkan performa situs agar dapat berjalan dengan responsif dan stabil. Pengujian dan *debugging* dilakukan secara berkala untuk memastikan setiap fitur berjalan sesuai kebutuhan dan bebas dari kesalahan.

Selama proses pengembangan, koordinasi dilakukan dengan *team lead* untuk memperoleh arahan teknis serta masukan yang berguna dalam meningkatkan kualitas aplikasi. Dokumentasi terkait arsitektur aplikasi dan kode program disusun secara terperinci sebagai bahan referensi bagi tim pengembang.

3.2 Tugas yang Dilakukan

Selama masa magang di PT Winnicode Garuda Teknologi, kepercayaan yang diberikan adalah mengembangkan sebuah situs berita menggunakan *framework* Laravel. Proyek ini bertujuan menyediakan platform yang dinamis dan mudah digunakan untuk penyajian berbagai konten berita secara online. Adapun lingkup pekerjaan yang dilakukan meliputi:

- Merancang antarmuka pengguna menggunakan Blade, HTML, CSS, dan JavaScript untuk menciptakan tampilan yang responsif dan mudah digunakan.
- Mengembangkan logika aplikasi *back-end* untuk fitur manajemen artikel, kategori berita, komentar pengguna, dan otorisasi akses.
- Membangun sistem autentikasi dan otorisasi menggunakan Laravel Sanctum untuk mengatur akses pengguna secara aman.

- Merancang dan mengelola struktur *database* menggunakan MySQL dan Laravel Eloquent ORM *Object-Relational Mapping* agar data tersimpan secara efisien dan terstruktur.
- Mengimplementasikan sistem keamanan aplikasi dengan validasi input, sanitasi data, dan proteksi CSRF *Cross-Site Request Forgery* untuk mencegah serangan seperti XSS *Cross-Site* dan *SQL Injection*.
- Melakukan pengujian dan debugging guna memastikan setiap fitur berjalan dengan baik dan aplikasi bekerja sesuai dengan kebutuhan pengguna.
- Menyusun dokumentasi teknis proyek untuk memudahkan proses pemeliharaan dan pengembangan lebih lanjut oleh tim lain.
- Berkoordinasi secara rutin dengan *team lead* untuk mendapatkan arahan, melakukan *review*, serta menyempurnakan fitur berdasarkan masukan.

Pelaksanaan kerja magang diuraikan secara rinci seperti pada Tabel 3.1.



Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Perancangan proposal proyek situs berita untuk PT.Winnicode Garuda Teknologi
2	Mempelajari <i>Laravel framework</i> dan <i>persiapantools development</i>
3	Membuat desain <i>UI/UX</i> dengan Figma
4	Setup <i>Laravel</i> proyek dan konfigurasi <i>environment</i>
5	Membuat <i>database</i> dan tabel untuk artikel, user, dan kategori
6	Pengembangan tampilan dasar (<i>header, footer, tata letak</i>)
7	Membuat halaman <i>home</i> dan halaman detail artikel
8	Pengembangan halaman <i>user</i> untuk input artikel
9	Implementasi <i>CRUD</i> artikel (<i>Create, Read, Update, Delete</i>) dan juga Integrasi <i>NewsAPI</i>
10	Membuat sistem <i>login</i> dan registrasi user
11	Implementasi kategori berita dan pencarian artikel
12	Pengembangan fitur komentar pada artikel
13	Testing fitur-fitur yang sudah dibuat
14	Perbaikan <i>bug</i> dan penambahan validasi form
15	Dokumentasi <i>code</i> dan cara penggunaan aplikasi
16	Upload ke <i>hosting</i> dan presentasi hasil akhir

3.3 Uraian Pelaksanaan Magang

Tugas yang dilakukan sebagai *Laravel Developer Intern* di PT. Winnicode Garuda Teknologi mencakup sebagai berikut:

3.3.1 Perancangan proposal proyek situs berita untuk PT.Winnicode Garuda Teknologi

Di minggu pertama magang, dilakukan penyusunan proposal proyek sesuai dengan ketentuan yang sudah diberikan dalam *jobdesk* perusahaan. Kegiatan ini merupakan langkah awal yang cukup penting dalam proses pengembangan situs berita menggunakan *Laravel*, karena di tahap ini harus bisa merumuskan konsep, tujuan, dan rencana kerja dengan jelas[4]. Tanpa perencanaan yang matang di awal,

tentunya pengembangan aplikasi akan sulit berjalan lancar dan sesuai target yang diinginkan perusahaan.

Proposal proyek yang telah disusun mencakup seluruh aspek pengembangan situs berita dan dapat dilihat secara lengkap melalui tautan berikut:

<https://docs.google.com/document/d/1RDyf8lNjzWxiBmdGCCzFTwHukOWq9YlibMRPpT8-FCE/edit?usp=sharing>

Berdasarkan proposal yang telah ditampilkan di atas, dalam menyusun proposal ini, dilakukan perumusan konsep aplikasi situs berita yang akan dibangun dengan framework Laravel. Tujuannya adalah agar fitur-fitur yang akan dikembangkan nantinya benar-benar sesuai dengan kebutuhan PT. Winnicode Garuda Teknologi[4]. Selain itu, perlu juga ditentukan batasan dan ruang lingkup proyek, seperti fitur apa saja yang realistis bisa diselesaikan dalam waktu 16 minggu magang. Hal ini penting supaya tidak terlalu ambisius dan bisa fokus pada pengembangan fitur-fitur utama yang memang dibutuhkan.

Proposal yang dibuat mencakup beberapa bagian utama. Bagian pendahuluan berisi latar belakang kenapa perlu ada situs berita yang baik, masalah apa yang sering terjadi dalam pengelolaan konten berita, dan alasan memilih Laravel sebagai framework untuk mengatasi masalah tersebut[4]. Untuk tujuan proyek, dibagi menjadi tujuan umum yaitu membangun situs berita yang terstruktur dan aman, serta tujuan khusus seperti membuat sistem *CRUD* *Create, Read, Update, Delete* untuk artikel, sistem login dan registrasi user, fitur kategori dan pencarian, plus sistem komentar untuk interaksi pembaca.

Dari segi metodologi, proposal menjelaskan pendekatan yang akan digunakan dalam mengembangkan aplikasi, mulai dari tools dan framework yang dipilih, sampai tahapan-tahapan pengembangan dari desain hingga upload ke server. Rencana kerja berisi timeline yang dibagi menjadi 4 bulan dengan aktivitas yang jelas: bulan pertama untuk riset dan analisis kebutuhan pengguna serta menyusun desain wireframe, bulan kedua untuk mengembangkan backend dan frontend, bulan ketiga untuk pengujian dan debugging, dan bulan keempat untuk finalisasi[4]. Sedangkan spesifikasi teknis mencakup kebutuhan sistem, teknologi yang akan dipakai seperti Laravel untuk backend, Bootstrap atau JavaScript untuk frontend, database MySQL, serta hosting di DomaiNesia atau DigitalOcean.

Setelah menyelesaikan proposal di akhir minggu pertama, telah diperoleh dokumen lengkap yang bisa dijadikan panduan selama proses magang. Dokumen

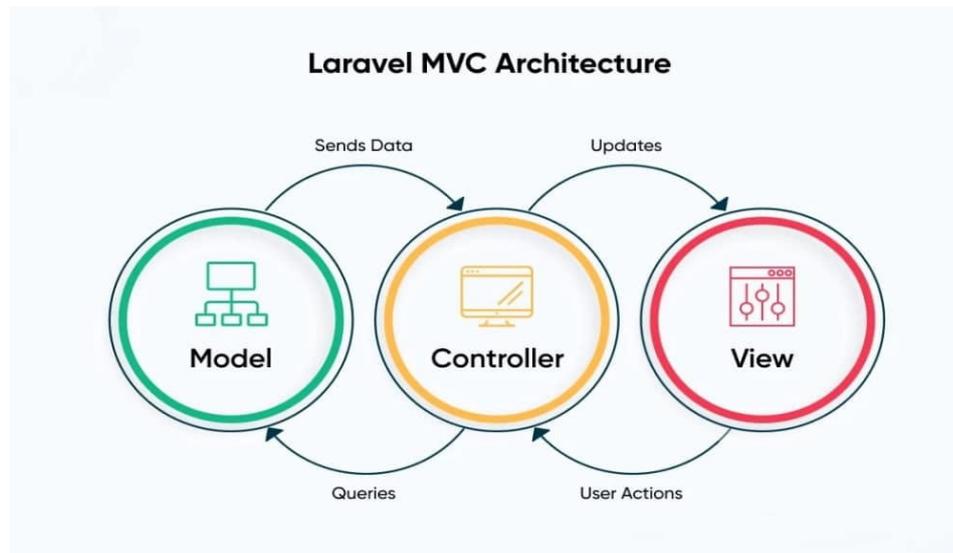
ini juga sudah mendapat persetujuan dari supervisor, sehingga dapat dimulai pelaksanaan rencana kerja yang sudah disusun. Kegiatan menyusun proposal ini ternyata cukup bermanfaat karena membantu dalam belajar menganalisis masalah, membuat perencanaan proyek yang realistis, dan berkomunikasi secara teknis dengan tim perusahaan.

Bagi perusahaan sendiri, adanya proposal yang jelas membantu memastikan bahwa proyek berjalan sesuai harapan dan tidak melenceng dari timeline yang sudah ditetapkan. Seperti yang dijelaskan dalam proposal, proyek ini diharapkan dapat memberikan dampak positif dalam menyediakan informasi yang cepat, akurat, dan mudah diakses oleh publik, sekaligus meningkatkan literasi masyarakat melalui konten dalam berbagai format seperti teks, video, dan infografis[4]. Dengan proposal yang sudah matang, diharapkan proses pengembangan situs berita bisa berjalan lebih efisien dan menghasilkan situs yang benar-benar berguna untuk kebutuhan PT. Winnicode Garuda Teknologi.

3.3.2 Mempelajari Laravel Framework dan Persiapan Tools Development

Pada minggu kedua magang, difokuskan pada proses mempelajari *Laravel framework* dan mempersiapkan *tools development* yang diperlukan untuk pengembangan situs berita. Kegiatan ini merupakan tahap fundamental yang menentukan kualitas dan efisiensi pengembangan aplikasi nantinya[1]. Persiapan yang matang di tahap ini akan memudahkan proses implementasi fitur-fitur situs berita di minggu-minggu berikutnya.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.1. Arsitektur *Model-View-Controller (MVC)* pada *Laravel Framework* [1]

Pada minggu kedua magang, difokuskan pada proses mempelajari *Laravel framework* dan mempersiapkan *tools development* yang diperlukan untuk pengembangan situs berita. Kegiatan ini merupakan tahap fundamental yang menentukan kualitas dan efisiensi pengembangan aplikasi nantinya[1]. Persiapan yang matang di tahap ini akan memudahkan proses implementasi fitur-fitur situs berita di minggu-minggu berikutnya.

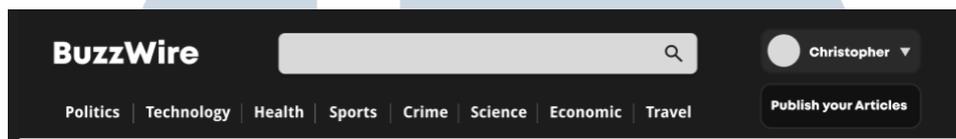
3.3.3 Perancangan tampilan antar muka dan Konfigurasi Sistem (Minggu 3-5)

Setelah menyelesaikan tahap pembelajaran *Laravel* dan persiapan *tools development*, dilakukan tahap perancangan tampilan antarmuka dan konfigurasi sistem yang mencakup tiga aktivitas utama dalam tiga minggu berturut-turut. Tahap ini merupakan transisi dari perencanaan teoritis menuju implementasi praktis, di mana konsep-konsep yang telah dipelajari mulai diterapkan dalam bentuk desain visual dan struktur teknis aplikasi[5].

Pada minggu ketiga, dilakukan pembuatan desain *UI/UX* menggunakan *Figma* sebagai *tools* utama. Proses desain dimulai dengan pembuatan *wireframe* untuk menentukan tata letak dan struktur halaman situs berita. Dirancang tampilan antarmuka untuk berbagai halaman seperti *homepage* yang menampilkan daftar artikel terbaru, halaman detail artikel, halaman kategori berita, serta tampilan antarmuka admin untuk manajemen konten. Dalam proses ini, diterapkan prinsip-

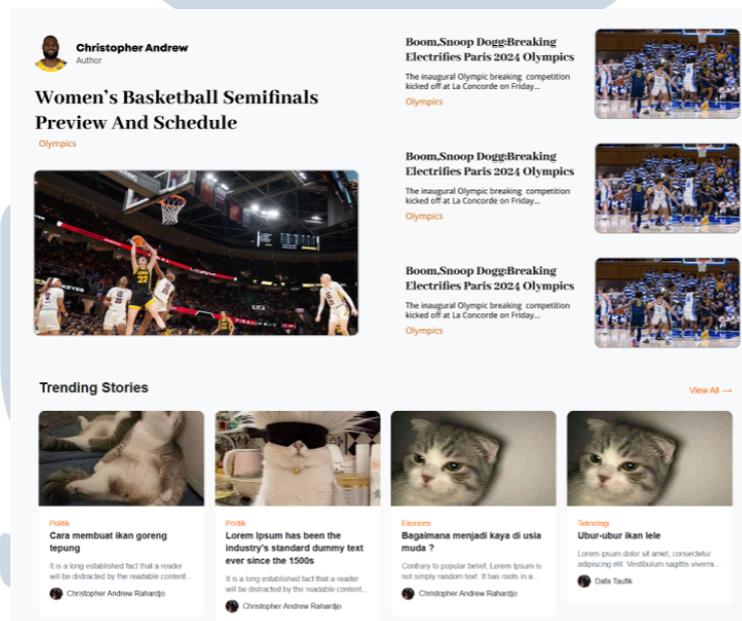
prinsip *user experience design* yang mengutamakan kemudahan navigasi dan aksesibilitas[5]. Desain yang dibuat juga mempertimbangkan aspek *responsive design* agar situs dapat diakses dengan baik di berbagai perangkat, mulai dari *desktop* hingga *mobile*.

Hasil desain UI/UX yang telah dibuat dapat dilihat pada beberapa *screenshots* berikut, dengan dokumentasi lengkap tersedia di platform *Figma*.



Gambar 3.2. Desain Navbar

Gambar 3.2 menampilkan struktur navigasi utama situs berita dengan logo di sebelah kiri, menu kategori berita di tengah, bar pencarian, dan area user profile di sebelah kanan. Desain menggunakan tata letak horizontal yang responsive dan konsisten dengan identitas merek.



Gambar 3.3. Desain Kartu Artikel

Gambar 3.3 menampilkan implementasi desain kartu artikel yang dirancang untuk menampilkan konten berita dalam format yang terstruktur dan mudah dibaca. Desain ini menggunakan sistem grid tata letak yang membagi konten menjadi beberapa area utama.

Pada bagian utama (main content area), terdapat artikel featured dengan judul "Women's Basketball Semifinals Preview And Schedule" yang dilengkapi dengan gambar berkualitas tinggi sebagai thumbnail utama. Kartu artikel ini menerapkan hierarki visual yang jelas dengan ukuran font yang berbeda untuk judul dan metadata artikel.

Di bagian bawah terdapat section "Trending Stories" yang menampilkan artikel-artikel populer dalam format kartu yang lebih kecil. Setiap trending story kartu terdiri dari thumbnail image, judul artikel yang ringkas, dan informasi metadata seperti waktu publikasi. Tata letak ini memungkinkan pengguna untuk dengan cepat memindai (scanning) berbagai konten yang tersedia.

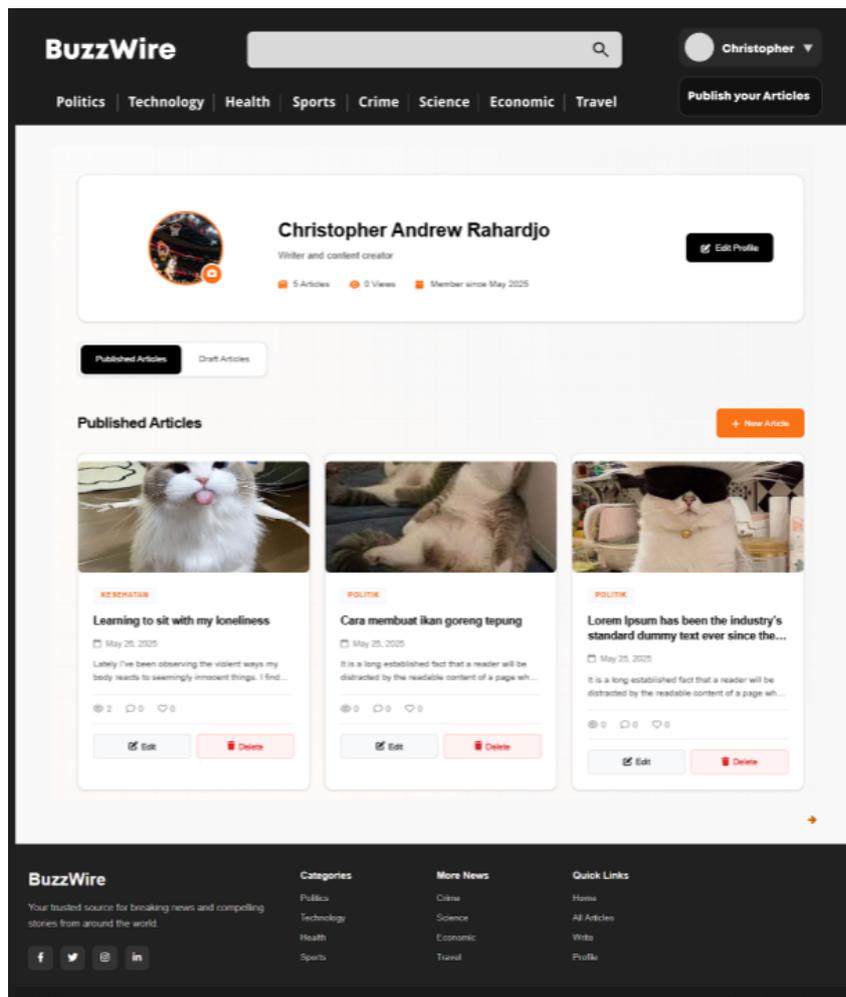
Sidebar di sebelah kanan menampilkan artikel-artikel terkait dengan topik serupa (dalam hal ini tentang Olympics) dalam format vertical list kartu. Setiap kartu dalam sidebar memiliki konsistensi desain dengan thumbnail, judul, dan brief description yang singkat.



Gambar 3.4. Desain Footer

Gambar 3.4 menampilkan desain footer situs BuzzWire yang terdiri dari informasi singkat tentang situs, ikon media sosial, daftar kategori berita, serta tautan navigasi cepat. Desain ini bertujuan untuk mempermudah pengguna dalam menjelajahi konten dan terhubung melalui berbagai platform.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



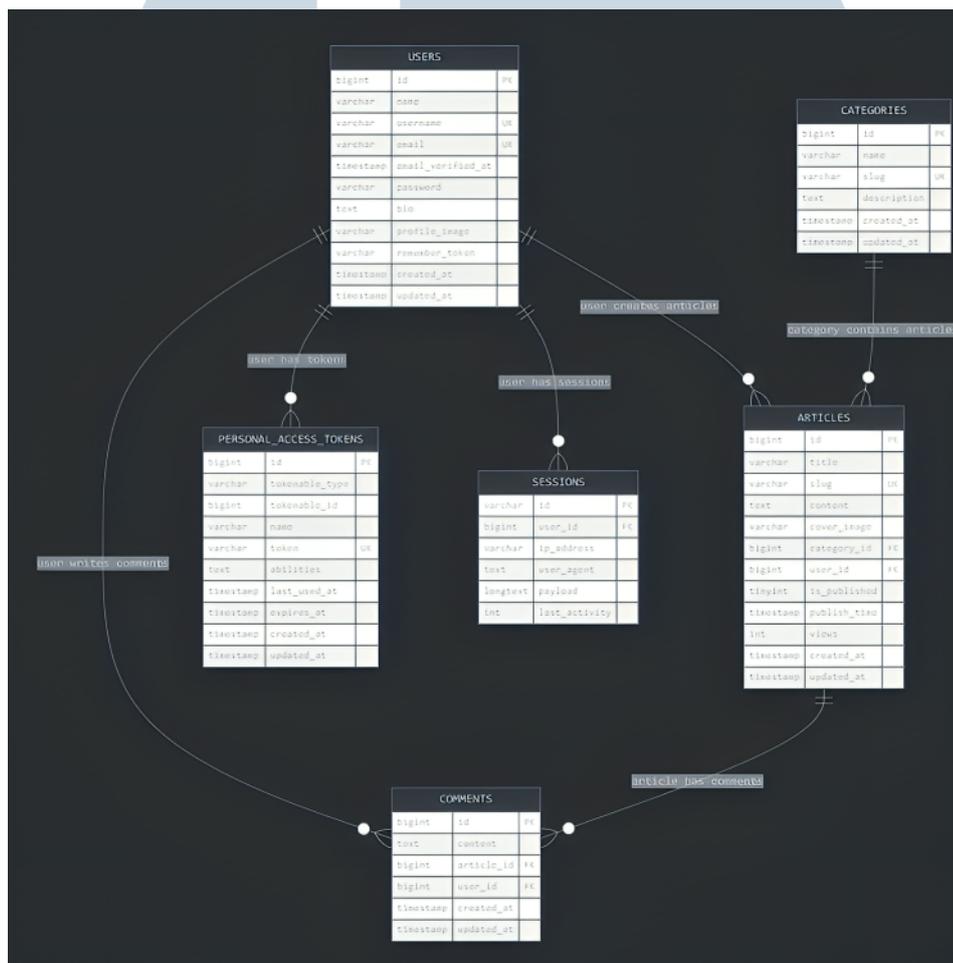
Gambar 3.5. Desain Profile Page

Gambar 3.5 menampilkan desain halaman profil pengguna pada platform penulisan artikel. Halaman ini menampilkan informasi dasar pengguna, seperti nama, peran, jumlah artikel, serta tanggal bergabung. Tersedia dua tab utama untuk menampilkan artikel yang telah dipublikasikan dan yang masih dalam bentuk draf. Setiap artikel ditampilkan dalam bentuk kartu lengkap dengan judul, kategori, tanggal publikasi, jumlah interaksi, serta tombol untuk mengedit atau menghapus artikel.

Minggu keempat dialokasikan untuk *setup* proyek Laravel dan konfigurasi *environment development*. Dibuat proyek Laravel baru menggunakan *Composer*, kemudian dilakukan konfigurasi dasar seperti pengaturan *environment variables*, koneksi *database*, dan struktur folder proyek. Tahap ini juga mencakup instalasi *dependencies* yang diperlukan dan konfigurasi *Laravel Mix* untuk kompilasi *asset* CSS dan JavaScript. Diimplementasikan arsitektur MVC Laravel yang telah

dipelajari sebelumnya dengan membuat struktur folder dan file yang terorganisir.

Pada minggu kelima, dilakukan fokus pada perancangan dan implementasi struktur *database*. Berdasarkan analisis kebutuhan aplikasi situs berita, dibuat *entity relationship diagram (ERD)* yang menggambarkan relasi antar tabel utama: *users* untuk data pengguna, *articles* untuk konten berita, *categories* untuk klasifikasi artikel, dan *comments* untuk interaksi pembaca.



Gambar 3.6. Entity Relationship Diagram (ERD) situs Berita BuzzWire

Implementasi struktur *database* dilakukan menggunakan *query SQL* langsung melalui *phpMyAdmin*, dibuat tabel-tabel tersebut beserta *field-field* yang diperlukan seperti *foreign key* untuk relasi antar tabel. Dengan selesainya tahap ini, fondasi teknis aplikasi sudah siap untuk tahap pengembangan *frontend* dan *backend* selanjutnya.

Melalui aktivitas perancangan tampilan antarmuka dan konfigurasi sistem, diperoleh kompetensi dalam penggunaan Figma untuk membuat *wireframe* dan

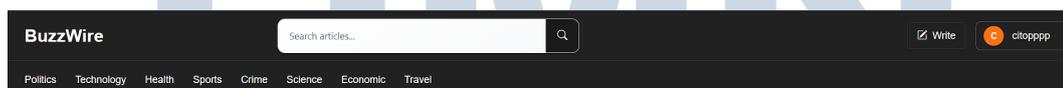
desain *UI/UX*, pemahaman prinsip *responsive design*, serta kemampuan merancang hierarki visual yang *user-friendly*. Dari aspek teknis, dikuasai *setup* proyek *Laravel*, konfigurasi *environment development*, implementasi arsitektur *MVC*, dan perancangan struktur database menggunakan *Entity Relationship Diagram (ERD)*. Pengalaman ini juga mengembangkan kemampuan analisis kebutuhan aplikasi dan perencanaan tahapan pengembangan yang sistematis.

3.3.4 Pengembangan Antarmuka Pengguna (Minggu 6-8)

Setelah menyelesaikan tahap perancangan tampilan antarmuka dan konfigurasi sistem, dilakukan fase pengembangan antarmuka pengguna yang merupakan implementasi dari desain *UI/UX* yang telah dibuat. Tahap ini mencakup konversi desain visual menjadi kode *HTML*, *CSS*, dan *JavaScript* yang terintegrasi dengan *Laravel*, serta sangat menentukan pengalaman pengguna dalam berinteraksi dengan situs berita.

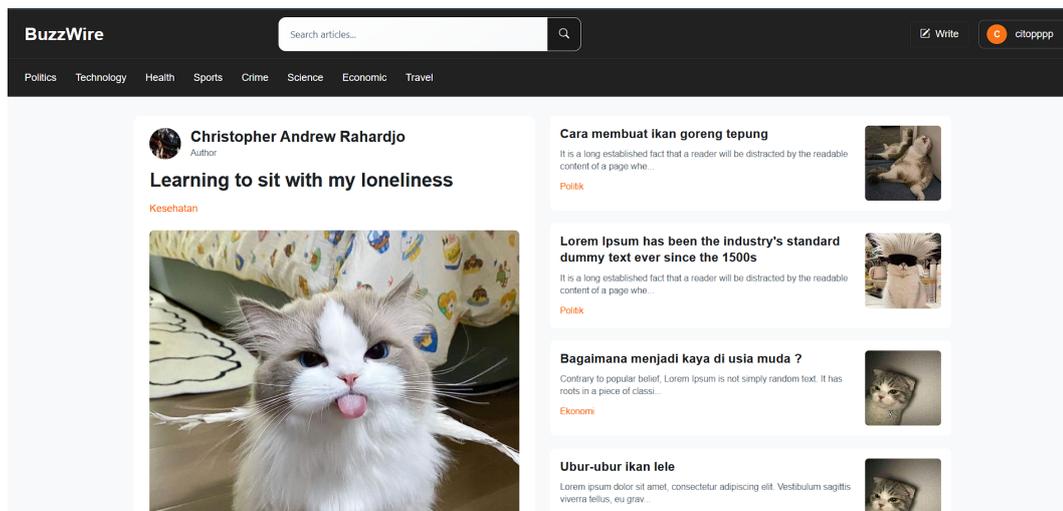
Pada minggu keenam, dikembangkan tampilan dasar meliputi header, footer, dan *tata letak* utama menggunakan *Laravel Blade templating engine*. Dibuat *master tata letak* yang dapat digunakan sebagai *template* untuk semua halaman, dengan header berisi navigasi intuitif dan menu kategori, serta footer dengan informasi kontak. *CSS framework Bootstrap* diimplementasikan untuk memastikan tampilan *responsive* dan konsisten di berbagai ukuran layar[6]. Komponen-komponen *reusable* seperti *navbar* dan *sidebar* juga dibuat untuk efisiensi *development*.

Hasil implementasi *tata letak* dasar dapat dilihat pada *screenshot* berikut:



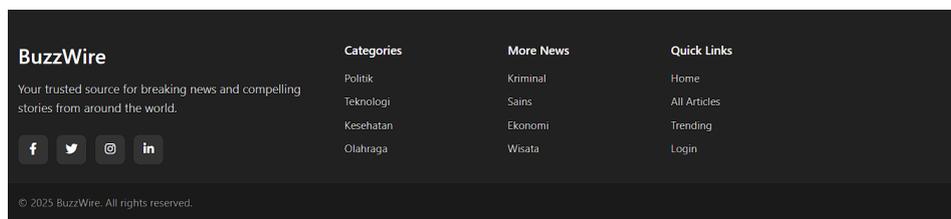
Gambar 3.7. Implementasi Navigation Bar

Gambar 3.7 menampilkan struktur navigasi utama situs berita dengan logo di sebelah kiri, menu kategori berita di tengah, bar pencarian, dan area user profile di sebelah kanan. Desain menggunakan tata letak horizontal yang *responsive* dan konsisten dengan identitas merek.



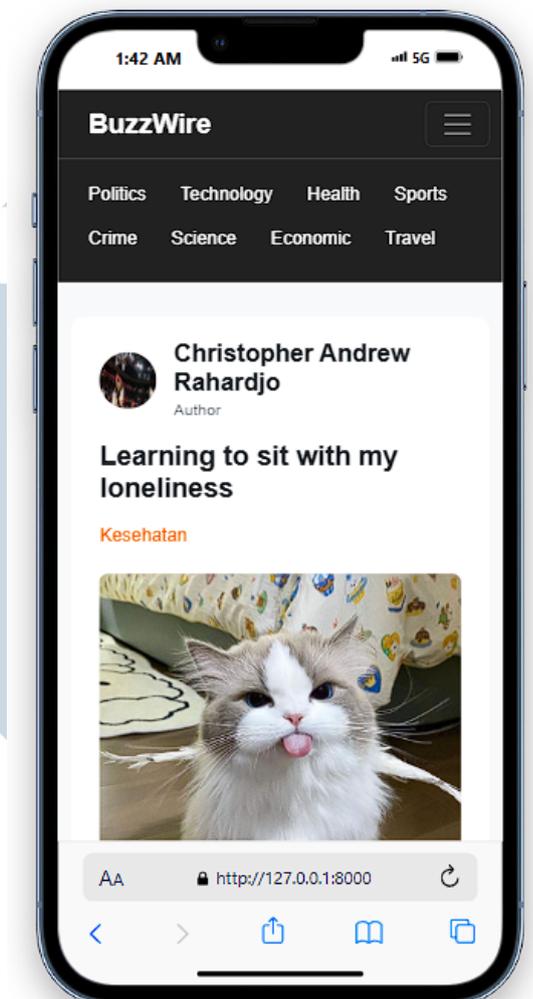
Gambar 3.8. tata letak Home Page

Gambar 3.8 menunjukkan tata letak halaman utama (home page) dari situs BuzzWire. Halaman ini menampilkan satu artikel utama secara lengkap di bagian kiri, sementara di sebelah kanan terdapat daftar artikel lainnya dalam format ringkas. Navigasi kategori berada di bagian atas, disertai fitur pencarian dan tombol untuk menerbitkan artikel. Desain ini dirancang untuk memberikan fokus pada konten utama sekaligus memudahkan pengguna menemukan artikel terkait.



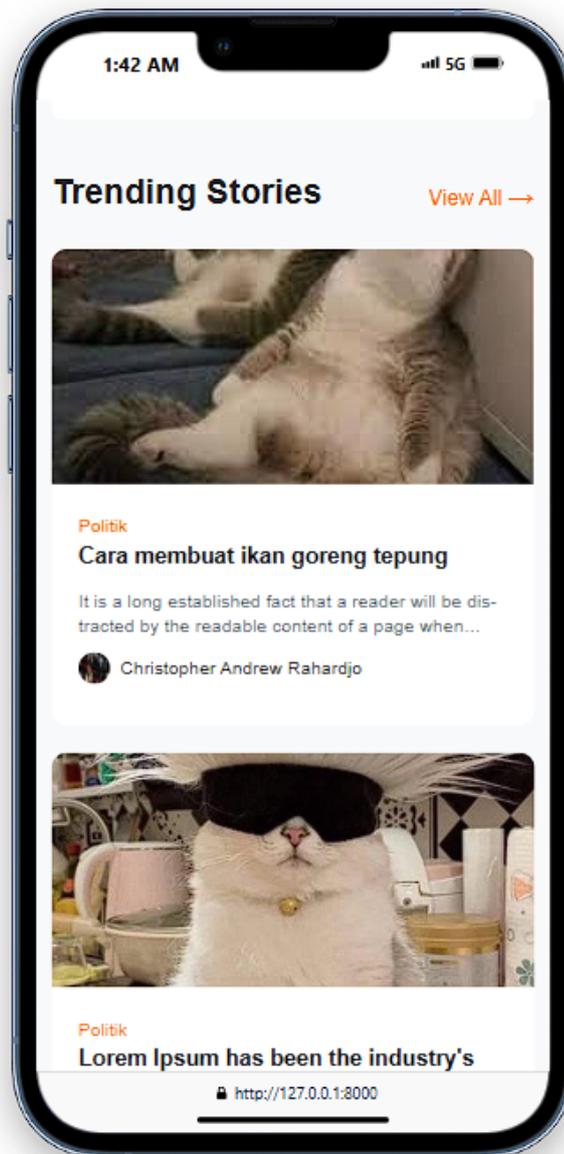
Gambar 3.9. Implementasi Footer dengan Informasi Kontak

Gambar 3.9 menampilkan desain footer situs BuzzWire yang terdiri dari informasi singkat tentang situs, ikon media sosial, daftar kategori berita, serta tautan navigasi cepat. Desain ini bertujuan untuk mempermudah pengguna dalam menjelajahi konten dan terhubung melalui berbagai platform.



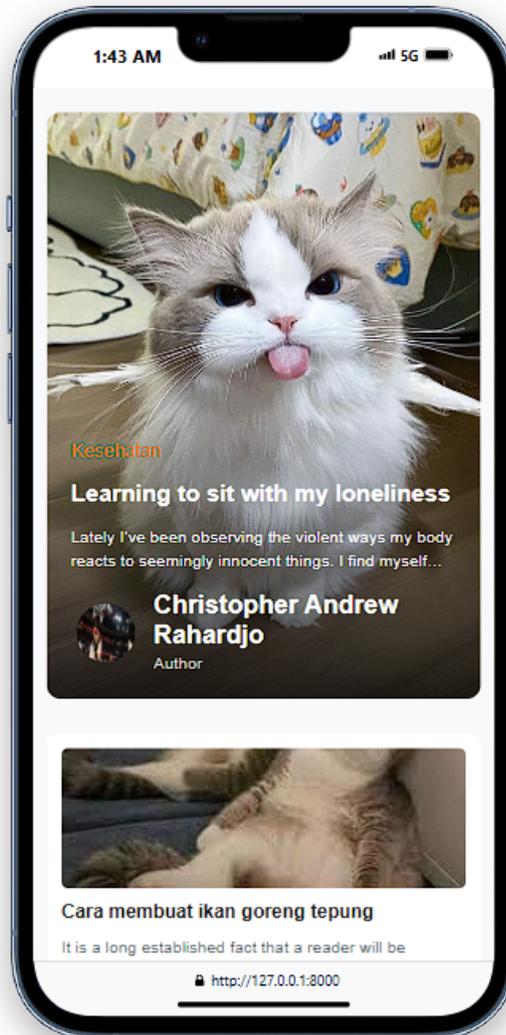
Gambar 3.10. Tampilan Responsive pada Mobile Device 1

Gambar 3.13 memperlihatkan tampilan responsif situs BuzzFeed pada perangkat mobile. Navigasi kategori disusun secara horizontal di bawah logo situs dan tombol menu, menyesuaikan ukuran layar kecil. Konten artikel ditampilkan secara terpusat dengan elemen-elemen penting seperti nama penulis, kategori, judul, dan gambar artikel tetap terbaca dan terstruktur dengan baik. Desain ini menunjukkan bahwa situs telah dioptimalkan untuk pengalaman pengguna yang nyaman di perangkat seluler.



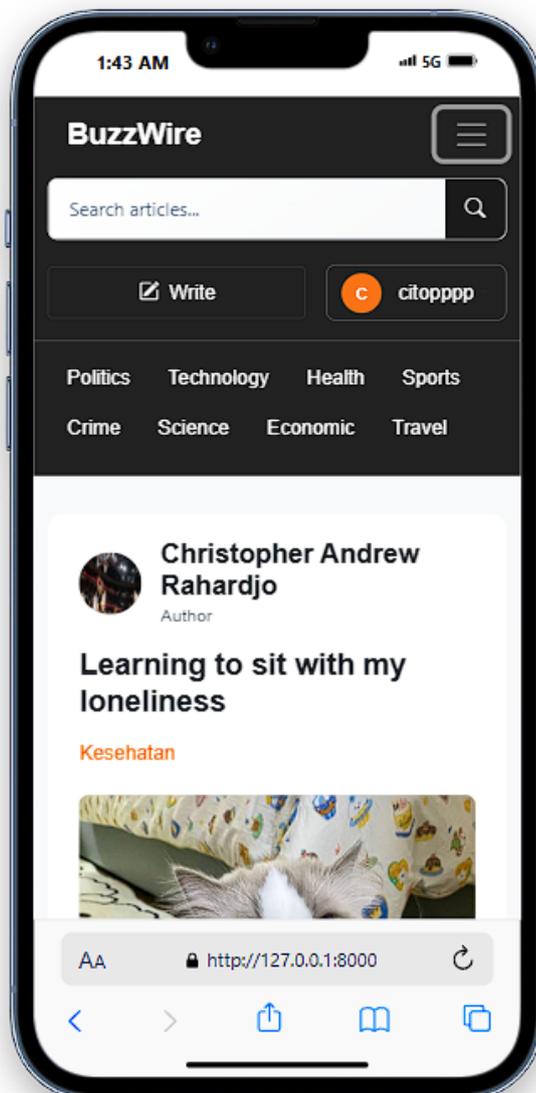
Gambar 3.11. Tampilan Responsive pada Mobile Device 2

Gambar 3.13 menampilkan tampilan section “Trending Stories” pada perangkat mobile. Setiap artikel ditampilkan dalam format kartu dengan gambar, kategori, judul, ringkasan isi, dan nama penulis secara berurutan. Elemen-elemen ini ditata secara vertikal agar mudah dibaca dan diakses pada layar kecil. Desain ini memperlihatkan bahwa fitur responsif berjalan baik, memastikan kenyamanan pengguna saat menjelajahi konten populer melalui smartphone.



Gambar 3.12. Tampilan Responsive pada Mobile Device 3

UNIVERSITAS
MULTIMEDIA
NUSANTARA



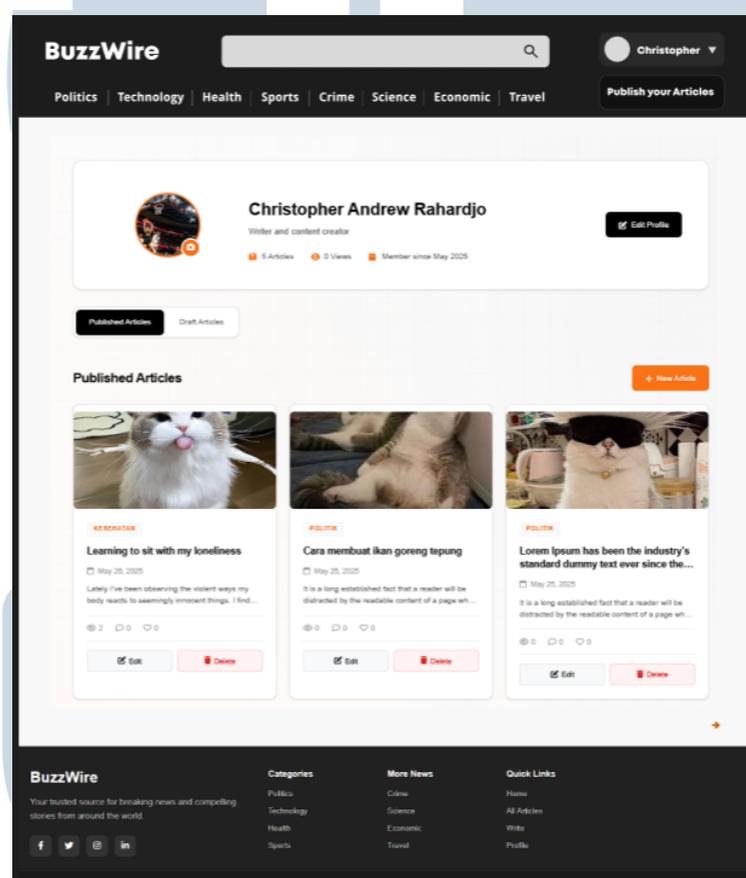
Gambar 3.13. Tampilan Responsive pada Mobile Device 4

Gambar 3.13 menunjukkan antarmuka utama aplikasi Buzzwire pada perangkat mobile dengan tampilan navigasi yang telah dioptimalkan. Elemen penting seperti kolom pencarian, tombol untuk menulis artikel, dan akses profil tetap tampil rapi di bagian atas layar. Navigasi kategori tersusun horizontal dan mudah diakses melalui scroll. Desain ini memastikan pengalaman pengguna tetap intuitif dan efisien meskipun di layar berukuran kecil.

Pada minggu kedelapan, dikembangkan *dashboard user* untuk manajemen artikel dengan pendekatan *user-centric design*. Tampilan antarmuka ini dirancang dengan *section* profil lengkap yang menampilkan foto, nama, bio, dan statistik artikel *user*. *Dashboard* dilengkapi dengan navigasi *tab* untuk memisahkan

Published Articles dan *Draft Articles*, serta tombol *New Article* yang *prominent* untuk membuat konten baru. Tampilan artikel menggunakan *card-based tata letak* yang menampilkan *thumbnail*, kategori (seperti KESEHATAN, POLITIK), judul, dan tanggal publikasi[7]. *User* dapat mengelola semua artikel mereka dari satu tampilan antarmuka yang intuitif tanpa pembagian *role* khusus, dimana setiap *user* memiliki kemampuan penuh untuk *create*, *read*, *update*, dan *delete* artikel.

Implementasi *dashboard user* dan sistem manajemen artikel dapat dilihat pada *screenshot* berikut:



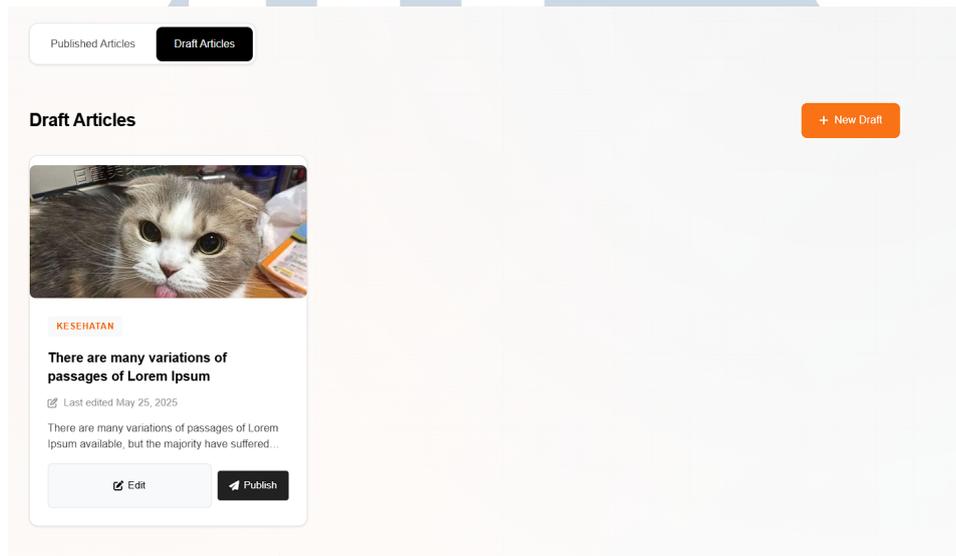
Gambar 3.14. Section Profil User dengan Foto, Bio, dan Statistik

Gambar 3.14 menampilkan halaman profil pengguna dalam platform BuzzWire. Di bagian atas halaman terdapat komponen identitas pengguna berupa foto profil, nama lengkap, bio singkat, dan statistik aktivitas, seperti jumlah artikel, total tampilan, dan tanggal bergabung.

Pengguna memiliki akses cepat untuk mengelola kontennya melalui tab *Published Articles* dan *Draft Articles*. Setiap artikel yang dipublikasikan ditampilkan dalam bentuk kartu yang mencakup gambar utama, kategori, judul

artikel, tanggal publikasi, dan ringkasan isi. Selain itu, tersedia tombol aksi untuk Edit dan Delete pada masing-masing artikel.

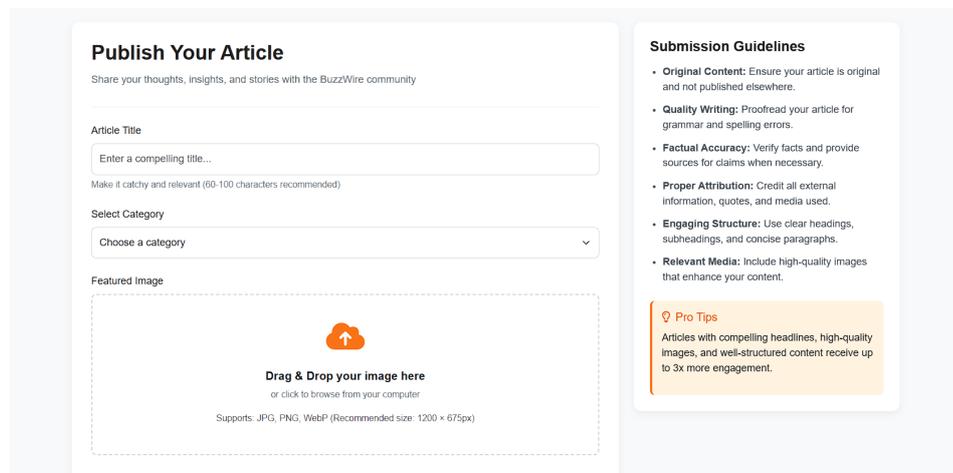
Tampilan halaman ini juga mencakup navigasi atas yang lengkap serta footer informatif dengan berbagai tautan cepat, menjadikan antarmuka pengguna ini intuitif dan responsif.



Gambar 3.15. Tab Draft Articles untuk Articles Belum Dipublikasi

Gambar 3.15 menampilkan tampilan tab Draft Artikel pada halaman profil pengguna, yang digunakan untuk mengelola artikel yang belum dipublikasikan. Dalam tab ini, setiap draft ditampilkan dalam bentuk kartu yang memuat gambar, kategori, judul, tanggal terakhir diedit, dan ringkasan artikel. Pengguna dapat memilih untuk mengedit atau langsung mempublikasikan artikel melalui tombol yang tersedia. Selain itu, terdapat tombol + New Draft untuk membuat artikel baru. Tampilan ini memudahkan pengguna menyusun dan menyimpan artikel sebelum diterbitkan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.16. Tampilan antar muka Manajemen Artikel

Gambar 3.16 menampilkan tampilan Publish Your Article yang dimana memperbolehkan user untuk menentukan judul artikel, category, cover image dan juga isi content dari artikel yang akan dipublish. Di bawahnya juga terdapat opsi untuk langsung publish ataupun didraft.

Melalui aktivitas pengembangan *frontend* dan implementasi *responsive design*, diperoleh kompetensi dalam penggunaan *HTML*, *CSS*, dan *JavaScript* untuk membangun antarmuka yang responsif di berbagai perangkat. Dikuasai teknik implementasi *mobile-first design*, optimasi tata letak untuk layar kecil, dan penggunaan *CSS Grid* serta *Flexbox* untuk struktur halaman yang fleksibel. Dari aspek manajemen konten, dikembangkan kemampuan merancang *dashboard* berbasis *user-centric design*, implementasi sistem *CRUD* untuk artikel, dan pengelolaan *tab navigation* yang intuitif. Pengalaman ini juga meningkatkan pemahaman tentang *user experience* pada platform pengelolaan konten dan kemampuan mengintegrasikan desain visual dengan fungsionalitas teknis.

3.3.5 Implementasi Logika Aplikasi (CRUD dan Integrasi NewsAPI) (Minggu 9-12)

Setelah menyelesaikan pengembangan antarmuka pengguna, dilakukan tahap implementasi logika aplikasi dan fitur interaktif yang merupakan inti dari fungsionalitas situs berita. Tahap ini mencakup pengembangan *backend logic* menggunakan *Laravel framework* dengan penerapan arsitektur *MVC* dan implementasi fitur-fitur yang memungkinkan *user* berinteraksi dengan aplikasi secara penuh[1].

Pada minggu kesembilan, diimplementasikan sistem CRUD (Create, Read, Update, Delete) untuk artikel yang menjadi fitur utama situs berita. Menggunakan *Eloquent ORM* Laravel, dibuat *controller ArticleController* yang menangani semua operasi CRUD dengan method `store()` untuk menyimpan artikel baru, `show()` untuk menampilkan detail artikel, `update()` untuk mengedit artikel, dan `destroy()` untuk menghapus artikel. Juga diimplementasikan fitur *draft article* yang memungkinkan *user* menyimpan artikel sebagai draft sebelum dipublikasi, serta *form validation* untuk memastikan data yang diinput sesuai dengan *requirement*[8]. Sistem ini terintegrasi dengan *dashboard user* yang telah dibuat sebelumnya, memungkinkan *user* mengelola artikel mereka dengan mudah. Implementasi sistem CRUD artikel dapat dilihat pada *screenshot* berikut:

Publish Your Article
Share your thoughts, insights, and stories with the BuzzWire community

Article Title
Why Most People Are Blind to Their Own Potential
Make it catchy and relevant (60-100 characters recommended)

Select Category
Kesehatan

Featured Image
Drag & Drop your image here
or click to browse from your computer
Supports .JPG, PNG, WebP (Recommended size: 1200 x 675px)

Submission Guidelines

- Original Content:** Ensure your article is original and not published elsewhere.
- Quality Writing:** Proofread your article for grammar and spelling errors.
- Factual Accuracy:** Verify facts and provide sources for claims when necessary.
- Proper Attribution:** Credit all external information, quotes, and media used.
- Engaging Structure:** Use clear headings, subheadings, and concise paragraphs.
- Relevant Media:** Include high-quality images that enhance your content.

Pro Tips
Articles with compelling headlines, high-quality images, and well-structured content receive up to 3x more engagement.

Gambar 3.17. Form Create Article dengan Validation

Article Content
Remove Image

Most people go through life unaware of the enormous reservoir of potential that lies dormant within them. They wake up, follow routines, check boxes, and go to sleep with a quiet frustration, never quite knowing what they're missing. But here's the uncomfortable truth: our limitations are more often illusions than realities. The boundaries we believe in aren't made of steel — they're made of stories. The Invisible Cage We Build There's a psychological concept called learned helplessness. It's when you stop trying after repeated failures — even when success becomes possible later. You no longer explore new paths, because you've internalized the belief that you'll fail again. You're not just living with fear. You're living inside it. The Stories That Shape Us Life often unfolds as a collection of stories we repeat in our minds — and more often than not, those stories hold us back instead...

Write your article using the editing tools above. Min 500 words recommended.

Save Draft Publish Article

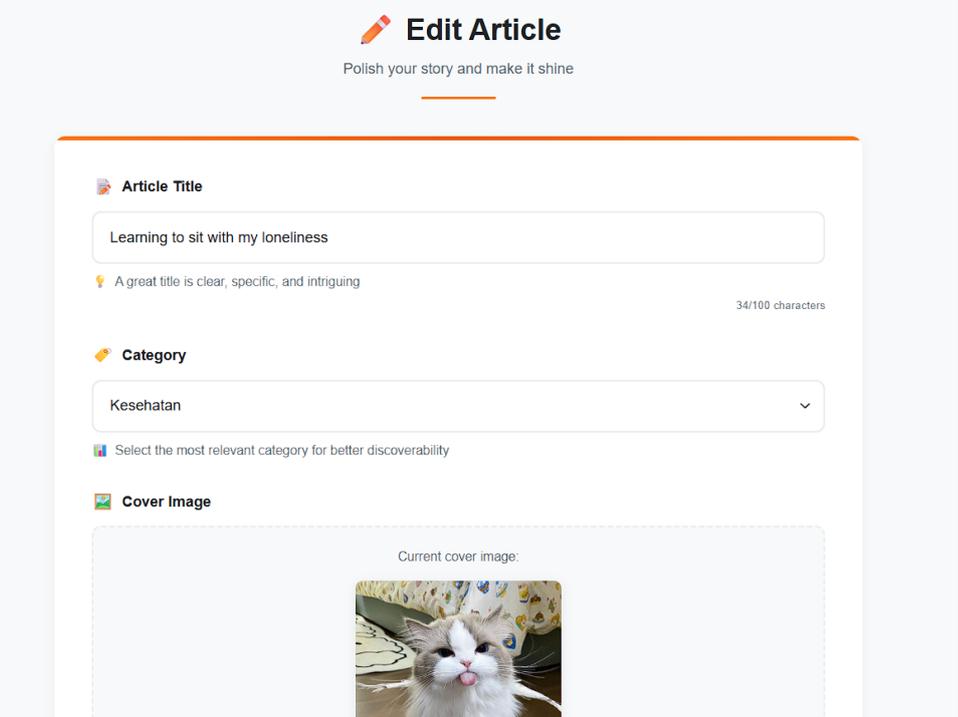
Submission Guidelines

- Original Content:** Ensure your article is original and not published elsewhere.
- Quality Writing:** Proofread your article for grammar and spelling errors.
- Factual Accuracy:** Verify facts and provide sources for claims when necessary.
- Proper Attribution:** Credit all external information, quotes, and media used.
- Engaging Structure:** Use clear headings, subheadings, and concise paragraphs.
- Relevant Media:** Include high-quality images that enhance your content.

Pro Tips
Articles with compelling headlines, high-quality images, and well-structured content receive up to 3x more engagement.

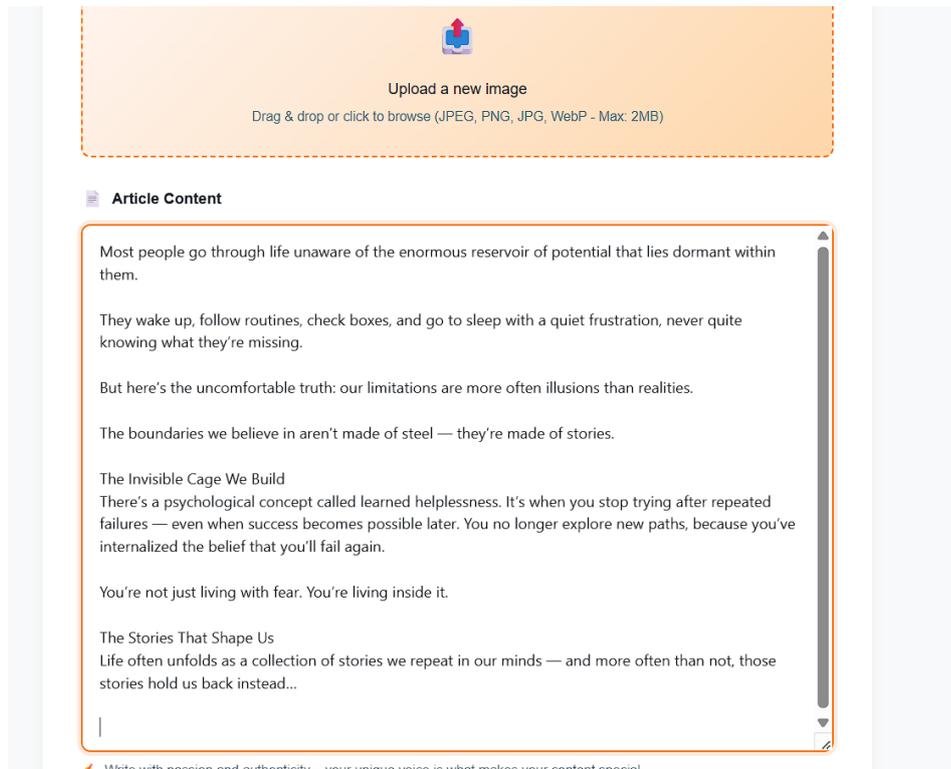
Gambar 3.18. Form Create Article dengan Validation dan juga fitur Draft

Gambar 3.17 dan gambar 3.18 menampilkan form "Publish Your Article" yang memerlukan validasi ketat. Setiap input seperti "Article Title" (yang direkomendasikan 50-100 karakter) dan "Featured Image" (dengan spesifikasi ukuran dan format) akan divalidasi di sisi server menggunakan rules Laravel. Jika data yang dimasukkan pengguna tidak memenuhi kriteria (misalnya, judul terlalu pendek atau format gambar salah), Laravel secara otomatis akan menampilkan pesan error di dekat field yang relevan dan mencegah data disimpan ke database, memastikan integritas dan kualitas konten berita. Fitur save draft juga disediakan agar dapat langsung dipublish ataupun disimpan dari draft.

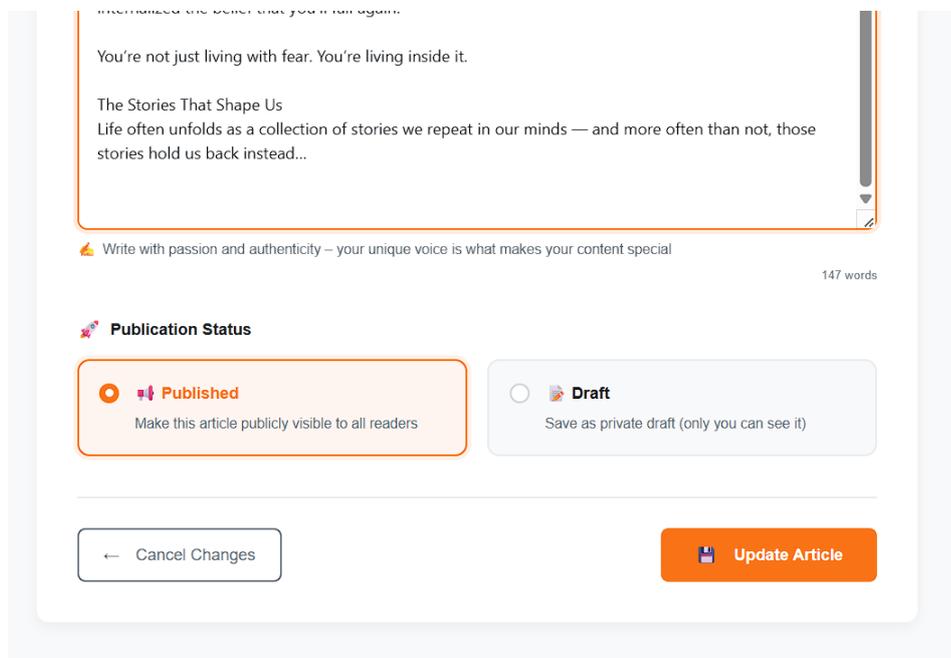


Gambar 3.19. Tampilan antar muka Edit Article dengan Data Existing(1)

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.20. Tampilan antar muka Edit Article dengan Data Existing(2)



Gambar 3.21. Tampilan antar muka Edit Article dengan Data Existing(3)

Tampilan Edit Article menggunakan **Route Model Binding** Laravel yang

secara otomatis mengambil data artikel dari database berdasarkan ID di URL. Method `edit(Article $article)` menerima object artikel yang sudah berisi semua data existing seperti title “Learning to sit with my loneliness”, kategori “Kesehatan”, cover image kucing putih, dan konten artikel.

Sistem melakukan **authorization check** dengan `if ($article->user_id !== Auth::id())` untuk memastikan hanya penulis artikel yang bisa mengedit. Form secara otomatis ter-isi dengan data existing melalui **Form Model Binding** Laravel, sehingga semua field menampilkan nilai dari database tanpa perlu coding manual.

Sistem juga melakukan **category validation** dengan memastikan selalu ada 8 kategori fixed sebelum menampilkan dropdown. Ketika user mengklik “Update Article”, method `update()` memproses perubahan dengan **validation rules** yang ketat, kemudian melakukan **direct assignment** ke object artikel seperti `$article->title = $request->title`.

File upload handling dilakukan dengan menghapus gambar lama jika ada upload baru, menggunakan filename unik dengan timestamp dan random string. Sistem mengimplementasikan **publishing logic** yang mencatat waktu publikasi pertama kali dengan `Carbon::now()`.

Setelah `$article->save()` yang menggunakan **Eloquent dirty checking** untuk update hanya field yang berubah, user di-redirect dengan **conditional flash message** yang berbeda untuk status published atau draft. Keseluruhan proses memastikan data existing tetap aman dengan otorisasi di kedua method `edit` dan `update`.

Pada minggu 11 sampai 12, dilakukan implementasi integrasi dengan NewsAPI untuk memperkaya konten situs berita dengan menggabungkan artikel yang diupload user dengan berita terkini dari sumber-sumber internasional terpercaya. Integrasi ini bertujuan untuk memberikan pengalaman menjelajah internet yang lebih komprehensif, dimana user dapat mengakses tidak hanya konten buatan komunitas lokal tetapi juga berita global yang relevan dalam satu platform[9].

Implementasi dimulai dengan pembuatan **NewsApiService** sebagai service class yang menangani semua komunikasi dengan external API. Servis ini menggunakan **Guzzle HTTP Client** Laravel untuk melakukan request ke NewsAPI endpoints, dengan konfigurasi API key yang disimpan secara aman dalam file `.env`. Arsitektur service pattern diterapkan untuk memisahkan logika pemanggilan API dari controller, meningkatkan pemeliharaan dan pengujian kode.

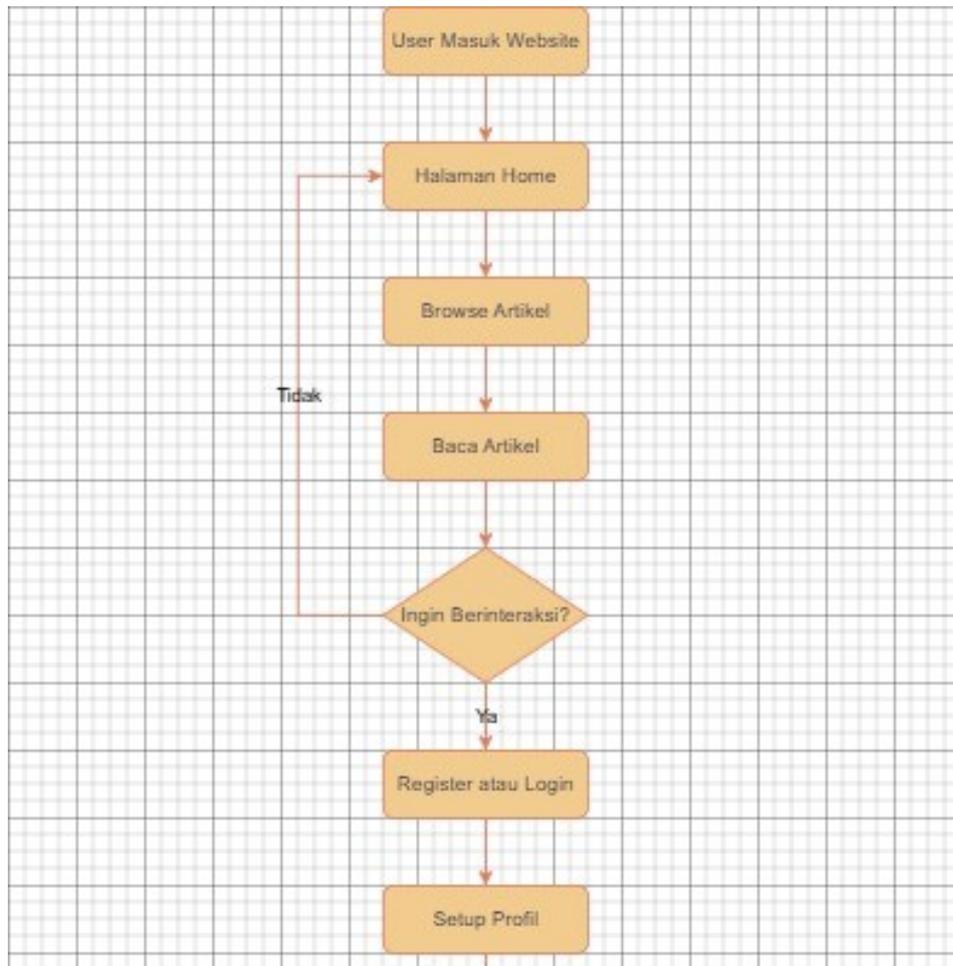
Melalui implementasi logika aplikasi dan integrasi *API*, diperoleh kompetensi mendalam dalam penggunaan *Eloquent ORM* untuk operasi *CRUD*, pemahaman arsitektur *MVC Laravel*, dan implementasi *form validation* yang ketat. Dikuasai teknik *Route Model Binding*, *authorization check*, dan *file upload handling* dengan sistem keamanan yang robust. Dari aspek integrasi eksternal, dikembangkan kemampuan menggunakan *Guzzle HTTP Client* untuk komunikasi dengan *NewsAPI*, penerapan *service pattern* untuk pemisahan logika, dan manajemen *API key* yang aman melalui *environment configuration*. Pengalaman ini juga meningkatkan pemahaman tentang *data validation*, *error handling*, dan implementasi fitur *draft system* untuk manajemen konten yang fleksibel.

3.3.6 Pengujian dan Penyelesaian proyek (Minggu 13-16)

Fase akhir pengembangan situs berita untuk PT. Winnicode Garuda Teknologi, yang berlangsung dari Minggu 13 hingga Minggu 16, adalah periode krusial yang berfokus pada pengujian menyeluruh, perbaikan bug, penyusunan dokumentasi, dan presentasi hasil akhir. Tahap ini memastikan aplikasi berfungsi secara optimal, stabil, dan siap untuk diluncurkan.

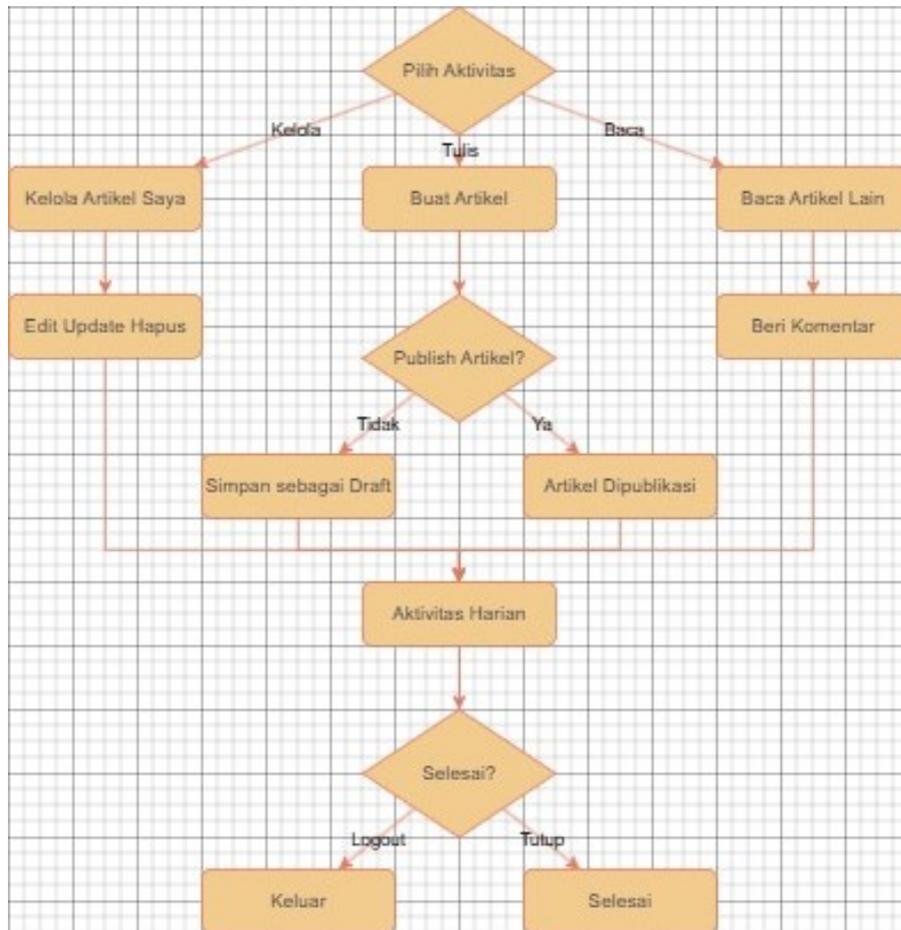
Pada Minggu 13, seluruh fitur yang telah diimplementasikan menjalani pengujian komprehensif. Fokus utamanya adalah mengidentifikasi potensi bug, error, atau anomali dalam fungsionalitas. Pengujian ini mencakup validasi fungsionalitas setiap fitur, seperti operasi *CRUD* artikel, sistem login/registrasi, pengelolaan kategori berita, fitur pencarian, dan sistem komentar, untuk memastikan semuanya bekerja sesuai rancangan. Selain itu, pengujian antarmuka pengguna (*UI testing*) dilakukan untuk memverifikasi tata letak, responsivitas desain, dan konsistensi visual di berbagai perangkat dan ukuran layar. Pengujian integrasi memastikan bahwa modul-modul yang berbeda, seperti interaksi antara artikel dan komentar atau pengguna dengan artikel, dapat bekerja sama dengan baik. Terakhir, pengujian performa dasar dilakukan untuk mengevaluasi responsivitas aplikasi dalam kondisi normal.

Setelah pengujian intensif di Minggu 13, Minggu 14 didedikasikan sepenuhnya untuk perbaikan semua bug dan error yang ditemukan. Prioritas diberikan pada bug kritis yang secara signifikan memengaruhi fungsionalitas inti aplikasi atau pengalaman pengguna. Setiap perbaikan diuji ulang secara cermat untuk memastikan masalah telah teratasi tanpa menimbulkan masalah baru.



Gambar 3.22. System User Flow Diagram

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.23. Blog System User Flow Diagram

Minggu 15 berpusat pada penyusunan dokumentasi Flow Penggunaan Aplikasi Platform Artikel. Dokumentasi ini menjelaskan alur kerja pengguna dari awal hingga akhir sesuai flowchart yang telah dibuat, meliputi langkah-langkah user masuk website, browse artikel, proses authentication, hingga aktivitas utama seperti membuat artikel, mengelola konten, dan berinteraksi melalui komentar untuk guest user dan registered user.

Sebagai tahap finalisasi proyek, Minggu 16 melibatkan upload aplikasi ke hosting dan presentasi hasil akhir. Proses deployment ke server hosting yang telah disiapkan mencakup konfigurasi database di server, pengaturan lingkungan, dan memastikan semua dependensi terinstal dengan benar. Setelah deployment, pengujian pasca-deployment dilakukan dengan cepat di lingkungan produksi untuk memastikan aplikasi berjalan dengan baik. Terakhir, sebuah presentasi hasil akhir disiapkan dan disampaikan kepada PT. Winnicode Garuda Teknologi. Presentasi ini mencakup demonstrasi fungsionalitas situs, penjelasan tentang fitur-fitur utama,

dan menampilkan hasil akhir dari keseluruhan proyek pengembangan. Dengan selesainya tahap ini, proyek pengembangan situs berita dianggap selesai dan siap untuk digunakan.

Melalui fase pengujian dan penyelesaian proyek, diperoleh kompetensi dalam melakukan *comprehensive testing* yang mencakup *functional testing*, *UI testing*, *integration testing*, dan *performance testing* dasar. Dikuasai metodologi *bug tracking* dan *debugging* sistematis untuk mengidentifikasi serta memperbaiki *critical bugs* tanpa menimbulkan masalah baru. Dari aspek dokumentasi, dikembangkan kemampuan menyusun *user flow documentation* yang menjelaskan alur kerja aplikasi secara komprehensif. Pengalaman *deployment* ke *hosting server* memberikan pemahaman tentang konfigurasi *production environment*, manajemen *dependencies*, dan *post-deployment testing*. Tahap ini juga meningkatkan kemampuan *project presentation* dan komunikasi teknis dalam mendemonstrasikan hasil pengembangan kepada *stakeholder*.

3.4 Kendala dan Solusi yang Ditemukan

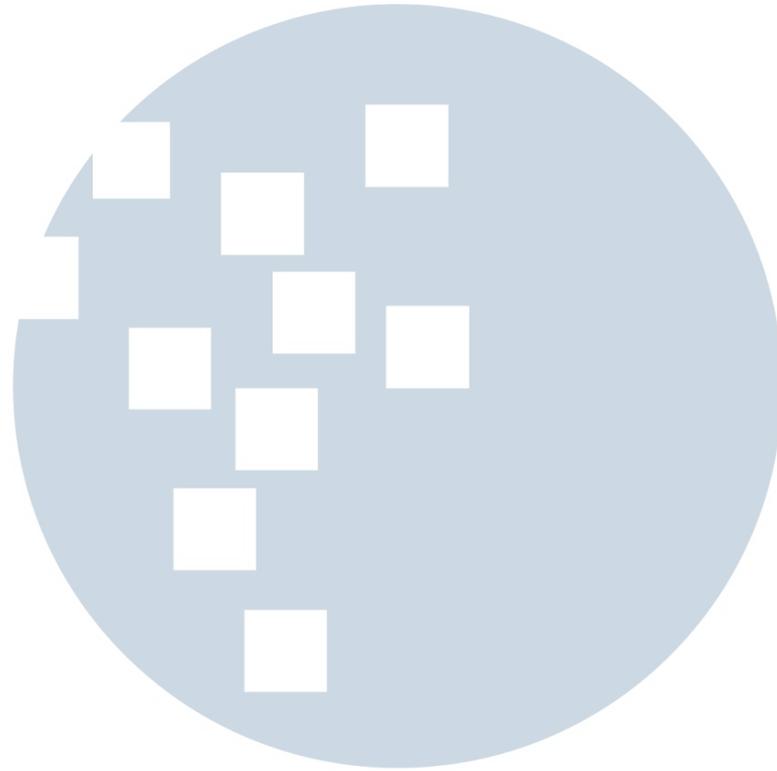
Kendala yang saya alami dalam mengerjakan proyek adalah:

1. Tabel *articles* dan *categories* dibuat tanpa memperhitungkan kebutuhan *slug*.
2. Terjadi *error* saat menambahkan kolom *slug* sebagai *field required* pada *data existing*.
3. Database menolak *constraint* pada *data existing* yang tidak memiliki nilai *slug*.
4. Terdapat inkonsistensi struktur data antara artikel lokal dan artikel dari *NewsAPI*.

Solusi yang saya terapkan untuk masalah *slug*:

1. Memodifikasi *migration* dengan membuat *field slug* sebagai *nullable* terlebih dahulu.
2. Membuat *script* untuk *generate slug* otomatis dari judul artikel dan nama kategori menggunakan *Laravel Str::slug() helper*.
3. Menambahkan validasi untuk memastikan *uniqueness slug* pada input baru.

4. Membuat *method normalizeApiArticle()* di *ArticleController* untuk standardisasi data antara artikel lokal dan *NewsAPI*.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA