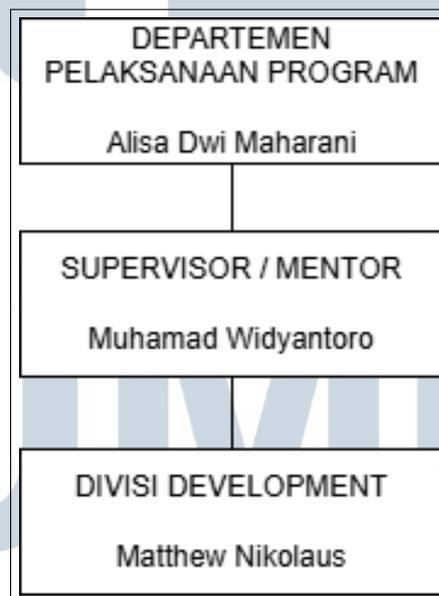


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama menjalani kegiatan magang di PT Winnicode Garuda Teknologi, posisi yang dijalankan adalah sebagai *Full Stack Developer* pada divisi *development*. Posisi ini memiliki peran penting dalam mendukung proses pengembangan *website* perusahaan, khususnya *website* portal berita. Program magang ini dilaksanakan di bawah supervisi Bapak Muhamad Widyantoro, S.Kom., sebagai mentor yang bertanggung jawab dalam memberikan tugas, melakukan evaluasi terhadap hasil pekerjaan, memberikan arahan teknis secara berkala, serta memberikan bimbingan selama proses magang. Gambar 3.1 menunjukkan struktur kedudukan di PT Winnicode Garuda Teknologi.



Gambar 3.1. Struktur Kedudukan di PT Winnicode Garuda Teknologi

Sumber: [4]

3.2 Tugas yang Dilakukan

Dalam masa kerja magang sebagai *full stack developer* di PT Winnicode Garuda Teknologi, tugas utama yang dijalankan adalah mengembangkan *website* portal berita dari tahap perancangan hingga implementasi akhir. Seluruh proses

pengembangan dilakukan secara mandiri, mulai dari pembuatan desain *wireframe* dan antarmuka pengguna menggunakan Figma, hingga pembangunan sisi *front-end* dan *back-end* menggunakan *framework* Next.js dengan bahasa pemrograman TypeScript dan *database* menggunakan PostgreSQL. Beberapa fitur utama yang berhasil dikembangkan meliputi:

1. Halaman utama berisi daftar berita terbaru.
2. Halaman kategori, untuk menampilkan berita berdasarkan topik.
3. Halaman detail berita/artikel, dengan tampilan konten lengkap.
4. Halaman *user profile*, pengguna dapat *upload profile picture*, edit nama, atau menghapus akun.
5. Sistem autentikasi, termasuk halaman *login* dan *register* untuk pengguna.
6. Fitur *search* berita/artikel, untuk mencari berita yang spesifik sesuai kebutuhan.
7. *Comment section* di setiap halaman detail berita/artikel, agar pembaca dapat saling berinteraksi.
8. Desain responsif, sehingga *website* dapat diakses dengan baik melalui perangkat *mobile* maupun *desktop*.

Dalam proses pengembangannya, perusahaan memberikan fleksibilitas penuh dalam pemilihan teknologi dan *tools*, dengan tetap mengacu pada kebutuhan fungsional dan pengembangan.

3.3 Uraian Pelaksanaan Magang

Selama pelaksanaan kerja magang, terdapat beberapa tanggung jawab pekerjaan yang harus dilakukan. Pelaksanaan kerja magang tersebut diuraikan dalam timeline proyek tabel 3.1.

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

Minggu Ke-	Pekerjaan yang dilakukan
1	Memulai tahap perencanaan dan riset studi kasus portal berita, riset teknologi yang digunakan (Next.js, Tailwind CSS, PostgreSQL, dst).
2	Merancang <i>flowchart</i> dan <i>use case diagram</i> , membuat daftar fitur utama, dan riset UI/UX desain berita modern.
3	Mendesain <i>wireframe</i> untuk halaman utama, halaman kategori, dan halaman berita/artikel.
4	Melanjutkan desain <i>wireframe</i> untuk halaman detail berita, halaman profil, dan halaman <i>login/register</i> .
5	Finalisasi desain UI/UX, <i>review</i> dan revisi desain (memperbaiki <i>spacing</i> dan komponen).
6	Inisialisasi proyek <i>front-end</i> dengan <i>framework</i> Next.js. Setup struktur <i>folder</i> , <i>routing</i> , dan <i>layout</i> dasar (<i>header/footer</i>). <i>Install dependency styling</i> dan <i>library</i> pendukung.
7	Mengembangkan dan implementasi <i>front-end</i> halaman utama (berita utama dan terbaru), halaman kategori berita, serta halaman <i>user profile</i> .
8	Mengembangkan dan implementasi <i>front-end</i> halaman <i>login/register</i> , halaman detail berita, dan integrasi tampilan komentar.
9	Memastikan UI responsif, simulasi koneksi ke <i>back-end</i> dengan <i>dummy</i> data. Finalisasi tahap pengembangan <i>front-end</i> , <i>review</i> dan revisi <i>front-end</i> halaman <i>website</i> .
10	Mulai melakukan tahap pengembangan <i>back-end</i> <i>website</i> . Setup <i>back-end</i> , konfigurasi database menggunakan PostgreSQL dan ORM (<i>Object-Relational Mapping</i>) menggunakan Prisma. Implementasi fitur <i>login</i> , <i>register</i> , dan autentikasi token.
11	Mengembangkan <i>back-end</i> halaman utama dengan menggunakan API dari perusahaan untuk <i>fetch</i> berita/artikel, API <i>endpoint</i> untuk berita terbaru dan fitur pencarian.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
12	Mengembangkan <i>back-end</i> halaman <i>user profile</i> , setup <i>upload profile picture</i> , edit <i>user name</i> , opsi untuk <i>delete account</i> , serta implementasi fitur komentar.
13	Finalisasi tahap pengembangan <i>back-end website</i> . <i>Review</i> dan revisi <i>back-end website</i> .
14	Mulai melakukan tahap integrasi dan testing <i>website</i> . Integrasi autentikasi <i>front-end</i> dan <i>back-end</i> , integrasi tampilan berita dinamis dan fitur komentar.
15	Melakukan <i>debugging</i> dan finalisasi. Uji responsivitas di berbagai ukuran layar (<i>mobile</i> dan <i>desktop</i>), uji alur pengguna dari <i>login</i> hingga eksplorasi berita dan interaksi.

3.3.1 Analisis Perancangan

Proses kerja magang diawali dengan tahap perencanaan dan riset yang akan menjadi dasar dalam pembuatan proyek selama magang berlangsung. Tahap ini dilakukan untuk memastikan bahwa seluruh proses pengembangan memiliki arah yang jelas dan sesuai dengan kebutuhan pengguna. Perancangan yang baik juga akan membantu meminimalkan risiko kesalahan pada tahap pengembangan dan mempercepat proses implementasi secara keseluruhan.

Berikut adalah teknologi dan *tools* yang digunakan dalam pengembangan *website* portal berita:

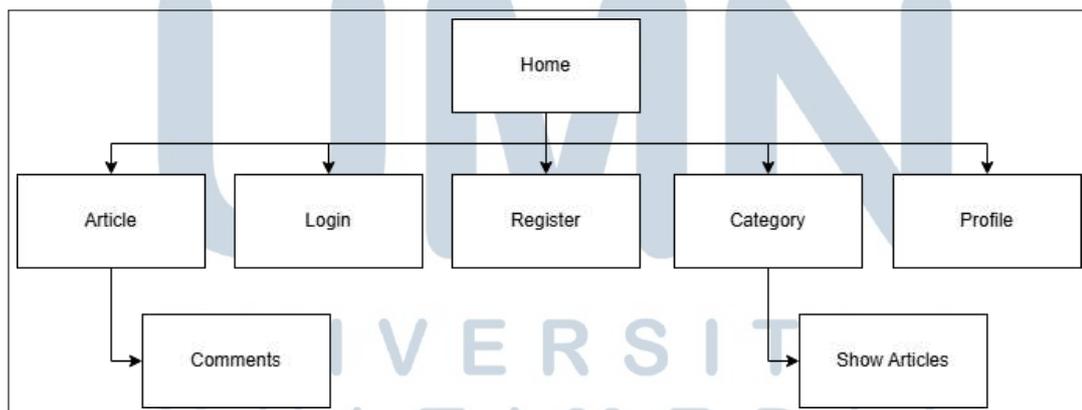
1. **Next.js**, digunakan sebagai *framework* utama untuk pengembangan *front-end* dan *back-end*. Selain itu, Next.js juga mendukung API *routes*, sehingga logika *back-end* dapat ditulis dalam satu proyek terpadu [5].
2. **Tailwind CSS**, adalah *utility-first CSS framework* yang memungkinkan pengembangan tampilan antarmuka secara cepat dan konsisten. Dengan Tailwind, *styling* dilakukan langsung dalam atribut HTML sehingga proses pengembangan rapi dan mudah dikustomisasi [6].
3. **TypeScript**, digunakan sebagai bahasa pemrograman utama dalam proyek ini. Dengan dukungan sistem *static typing*, TypeScript membantu

mengurangi *bug* sejak tahap pengembangan awal dan memudahkan proses *debugging* [7].

4. **Shadcn/ui**, merupakan *library* komponen UI yang dibangun di atas Tailwind CSS dan dirancang untuk integrasi cepat dengan Next.js dan TypeScript. *Library* ini menyediakan komponen-komponen UI seperti tombol, *form*, *card*, dan lainnya yang sudah siap pakai dan memiliki desain modern, konsisten, serta dapat dikustomisasi sesuai kebutuhan proyek [8].
5. **PostgreSQL**, dipilih sebagai database untuk menyimpan dan mengelola data *website* portal berita seperti data pengguna dan komentar [9].
6. **Prisma**, merupakan ORM (*Object-Relational Mapping*) yang digunakan untuk menghubungkan aplikasi dengan *database* PostgreSQL untuk mempermudah pengelolaan *database* [10].

A. Sitemap

Sitemap merupakan representasi visual dari struktur navigasi sebuah *website* yang bertujuan untuk menggambarkan hubungan antar halaman secara hierarkis. Gambar 3.2 menunjukkan *sitemap* yang telah dibuat pada pengembangan *website* portal berita ini.



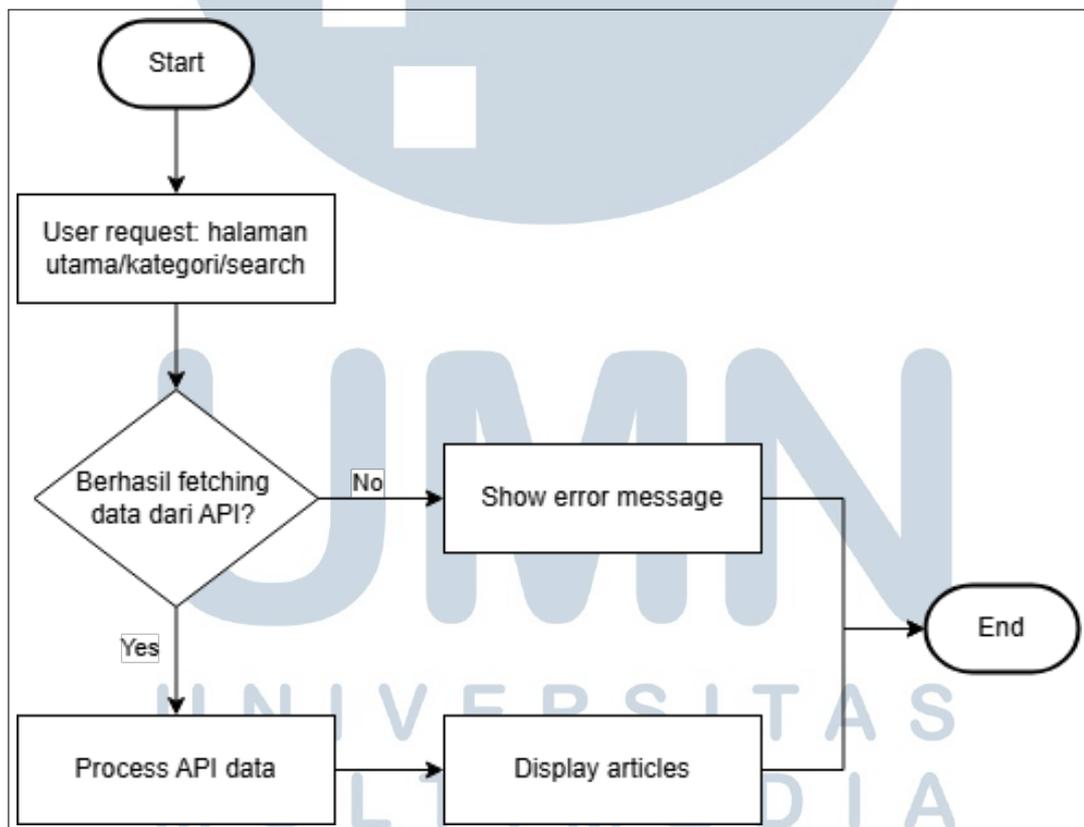
Gambar 3.2. *Sitemap Website Portal Berita*

Pada pengembangan *website* portal berita ini, *sitemap* disusun untuk memberikan gambaran alur navigasi pengguna dari halaman utama (*Home*) menuju halaman-halaman lain seperti Artikel, *Login*, *Register*, Kategori, dan Profil. Melalui halaman Artikel, pengguna dapat melihat detail berita dan memberikan komentar

melalui fitur komentar. Pada halaman Kategori, pengguna dapat menelusuri berita berdasarkan kategori tertentu, sedangkan pada halaman Profil, pengguna dapat melakukan pengelolaan akun seperti mengunggah foto profil, mengubah nama, atau menghapus akun. Secara keseluruhan, *sitemap* ini membantu dalam proses perancangan serta memastikan alur penggunaan aplikasi berjalan dengan logis dan mudah dipahami oleh pengguna.

B. Flowchart

Flowchart merupakan diagram alur yang digunakan untuk memvisualisasikan langkah-langkah atau proses dalam sistem. Gambar 3.3 merupakan sebuah *flowchart* dari proses pengambilan (*fetching*) data artikel atau berita dari API yang disediakan oleh PT Winnicode Garuda Teknologi.

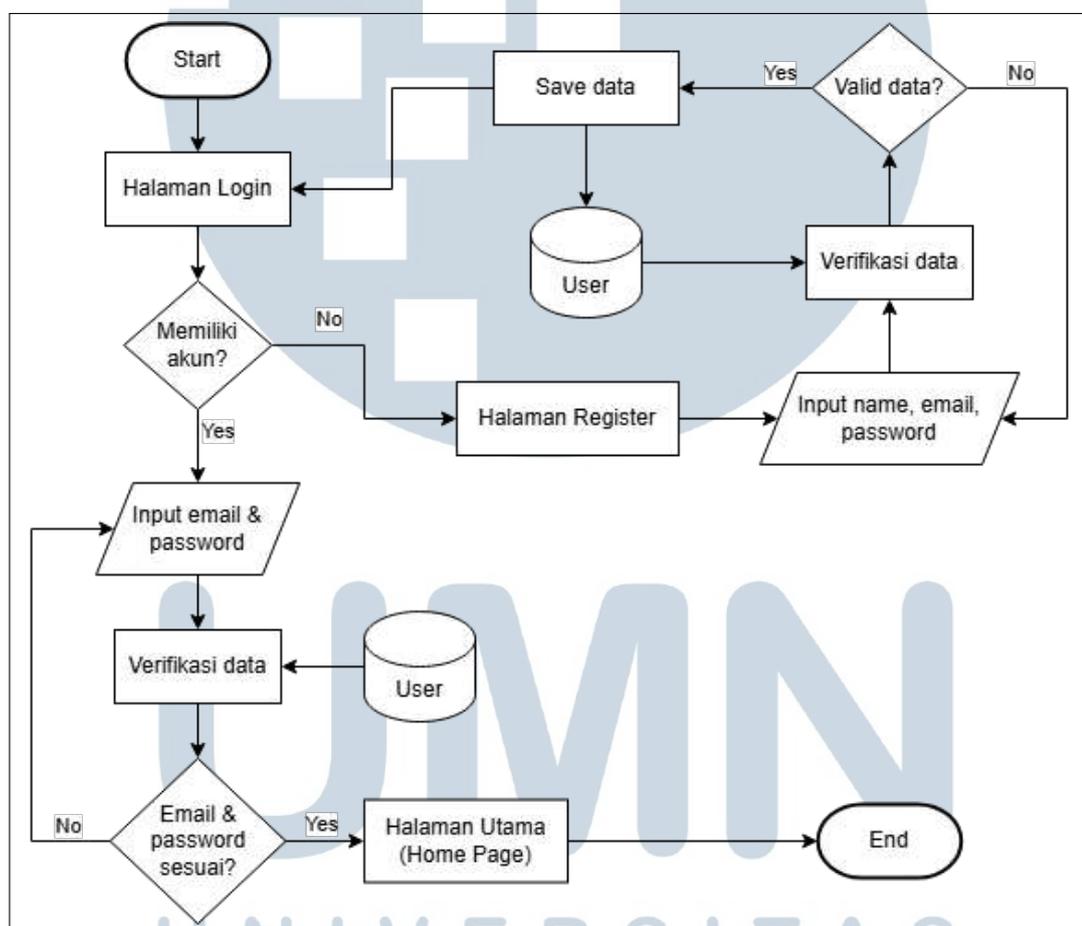


Gambar 3.3. *Flowchart Fetching* Artikel Berita dari API

Alur proses dari *flowchart* di atas dimulai saat pengguna melakukan permintaan untuk mengakses salah satu halaman yang menampilkan daftar artikel. Kemudian, sistem mencoba melakukan *fetching* data dari API PT Winnicode

Garuda Teknologi. Jika proses *fetching* berhasil, maka data dari API akan diproses terlebih dahulu sesuai kebutuhan *front-end*, lalu hasilnya akan ditampilkan pada halaman berupa daftar artikel. Jika proses *fetching* gagal karena koneksi buruk atau respons *error* dari API, maka sistem akan menampilkan pesan *error* kepada pengguna. *Flowchart* ini penting sebagai panduan untuk pengembangan logika aplikasi dalam menangani data dinamis yang bergantung pada API.

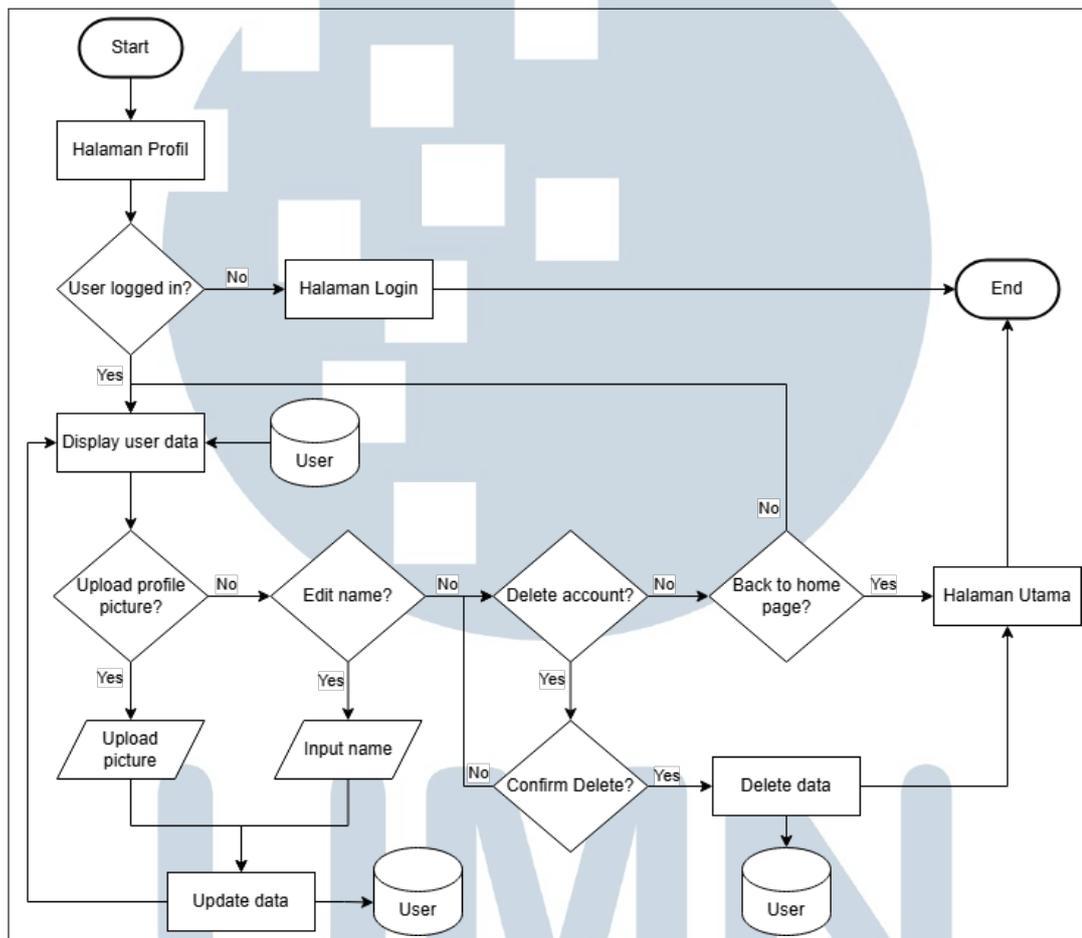
Gambar 3.4 merupakan *flowchart* yang dibuat untuk menggambarkan proses *login* dan *register* pengguna pada sistem *website* portal berita ini.



Gambar 3.4. *Flowchart Login dan Register*

Flowchart ini menggambarkan dua alur utama, yaitu proses *login* dan *register*. Ketika pengguna mengakses halaman *login*, pengguna dapat memilih untuk *login* jika sudah memiliki akun atau *register* jika belum memiliki akun. *Flowchart* ini membantu dalam mendeskripsikan alur autentikasi pengguna, sekaligus menjadi dasar pengembangan logika *back-end* dan *front-end* yang berkaitan dengan proses *login* dan pendaftaran akun.

Gambar 3.5 merupakan *flowchart* yang dibuat untuk menjelaskan alur proses interaksi pengguna pada halaman profil dalam *website* portal berita ini. Halaman profil memungkinkan pengguna untuk melakukan beberapa tindakan seperti *upload* foto profil, mengubah nama, dan menghapus akun.

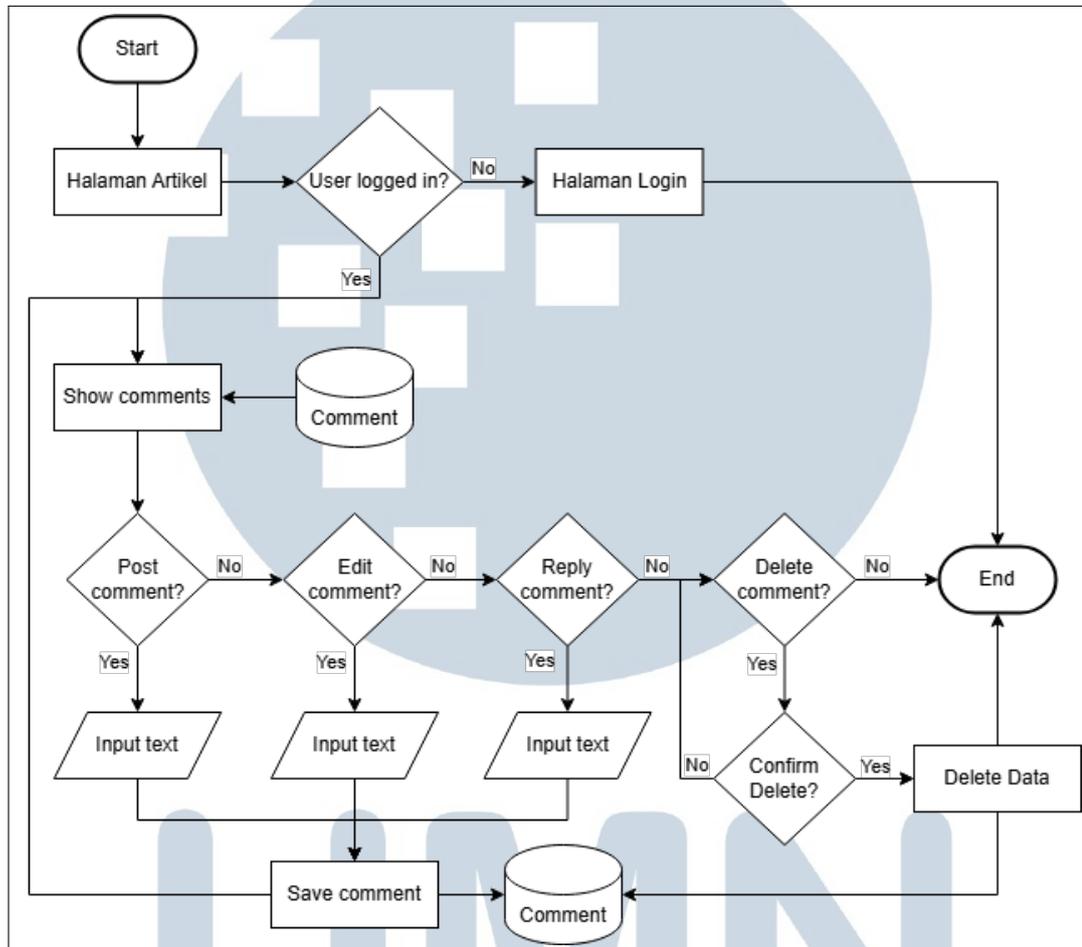


Gambar 3.5. *Flowchart* Halaman Profil

Proses dimulai saat pengguna mengakses halaman profil. Sistem akan memeriksa apakah pengguna sudah *login*. Jika belum, maka pengguna akan diarahkan ke halaman *login*. Jika sudah *login*, maka data pengguna akan ditampilkan. Selanjutnya, pengguna diberikan beberapa opsi, seperti *upload* foto profil, *edit* nama, menghapus akun, atau kembali ke halaman utama. *Flowchart* ini berguna dalam merancang dan mengembangkan fitur manajemen akun, dengan memastikan semua tindakan penting seperti penghapusan akun memerlukan konfirmasi untuk mencegah kesalahan pengguna.

Gambar 3.6 menunjukkan *flowchart* yang menggambarkan alur proses

interaksi pengguna dengan fitur komentar yang terdapat di halaman artikel *website* portal berita. Fitur ini memungkinkan pengguna untuk berpartisipasi secara aktif dengan menanggapi isi artikel melalui komentar.



Gambar 3.6. Flowchart Fitur Komentar

Proses dari *flowchart* di atas dimulai saat pengguna membuka halaman artikel. Sistem akan mengecek apakah pengguna sudah *login*. Jika belum, maka pengguna akan diarahkan ke halaman *login* untuk melakukan *login* terlebih dahulu. Jika sudah *login*, maka komentar-komentar yang sudah ada akan ditampilkan. Pengguna dapat melakukan beberapa aksi, seperti menulis komentar baru, mengubah isi komentar yang telah dibuat sebelumnya, menanggapi atau *reply* komentar pengguna lain, atau menghapus komentar yang telah dibuat pengguna tersebut.

Flowchart ini membantu memastikan bahwa setiap aksi pengguna pada fitur komentar dilakukan secara terstruktur dan aman, terutama pada tindakan sensitif

seperti penghapusan data. Selain itu, *flowchart* ini menjadi pedoman penting dalam proses perancangan dan implementasi fitur interaktif yang bersifat *real-time* dan dinamis.

C. Struktur Tabel

Berikut adalah struktur tabel yang digunakan dalam *website* portal berita ini. Tabel 3.2 menunjukkan struktur tabel *User* yang digunakan untuk menyimpan data pengguna, seperti informasi identitas, kredensial *login*, dan data profil.

Tabel 3.2. Struktur Tabel *User*

Nama	Tipe Data	Keterangan
id	text	<i>Primary key, generated</i> dengan <i>cuid()</i>
name	text	Nama pengguna
email	text	<i>Email</i> unik pengguna
password	text	Kata sandi pengguna
image	bytea	Gambar profil pengguna
imageType	text	Tipe <i>file</i> gambar profil
createdAt	timestamp	Tanggal akun pengguna dibuat
updatedAt	timestamp	Tanggal akun pengguna diperbarui terakhir

Tabel 3.3 merupakan struktur tabel *Comment* yang digunakan untuk menyimpan data komentar yang diberikan pengguna terhadap artikel.

Tabel 3.3. Struktur Tabel *Comment*

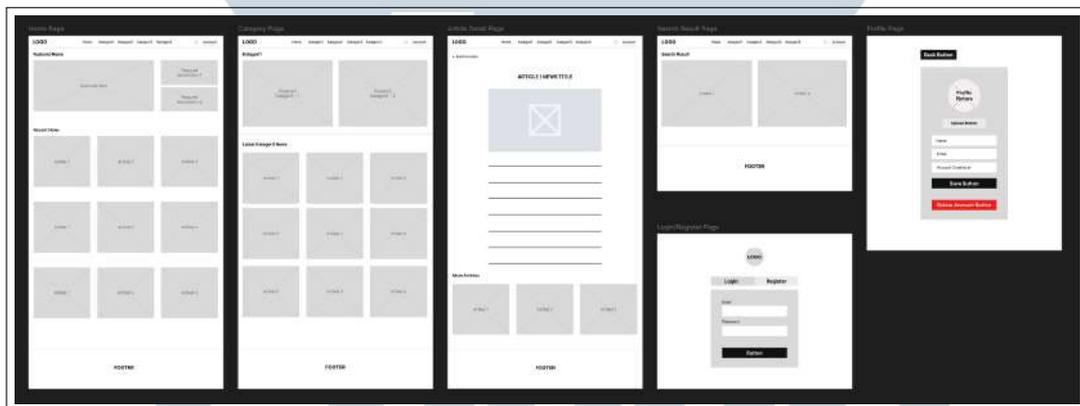
Nama	Tipe Data	Keterangan
id	text	<i>Primary key, generated</i> dengan <i>cuid()</i>
content	text	Isi komentar
articleId	text	ID artikel terkait
userId	text	<i>Foreign key</i> , untuk pemilik komentar
parentId	text	ID komentar induk, untuk komentar <i>reply</i>
Lanjut pada halaman berikutnya		

Tabel 3.3 Struktur Tabel *Comment* (lanjutan)

Nama	Tipe Data	Keterangan
createdAt	timestamp	Tanggal komentar dibuat
updatedAt	timestamp	Tanggal komentar diperbarui

D. Wireframe

Wireframe adalah representasi visual sederhana dari struktur antarmuka pengguna sebuah *website*. Pada tahap ini, desain belum berfokus pada warna atau elemen estetika, tetapi lebih pada penempatan dan fungsi elemen-elemen utama yang akan digunakan pengguna. Gambar 3.7 merupakan *wireframe* dari halaman-halaman utama pada *website* portal berita yang dikembangkan.

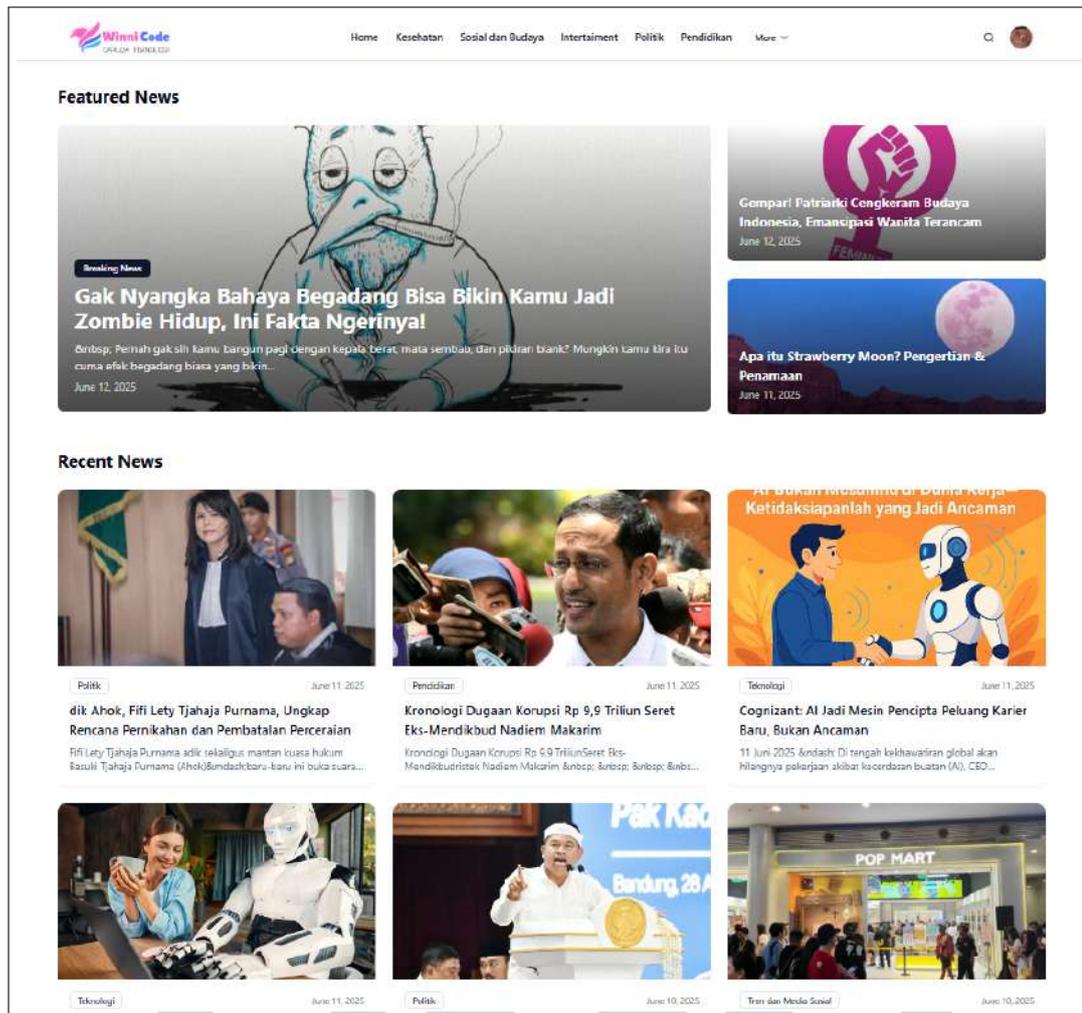


Gambar 3.7. *Wireframe Website Portal Berita*

Wireframe ini dibuat sebagai pedoman awal untuk pengembangan tampilan antarmuka *website*, dan membantu untuk memahami alur navigasi serta fungsi-fungsi utama dari setiap halaman sebelum masuk ke tahap implementasi visual secara penuh.

3.3.2 Implementasi

Hasil implementasi antarmuka *website* portal berita dapat dilihat pada tampilan halaman-halaman *website* berikut. Gambar 3.8 merupakan tampilan awal ketika pengguna membuka *website* portal berita ini. Halaman utama dirancang untuk menampilkan informasi berita dengan struktur yang jelas dan menarik.



Gambar 3.8. Halaman Utama

Pada bagian atas halaman terdapat tiga artikel unggulan (*featured news*) yang ditampilkan secara lebih besar agar dapat menarik perhatian pengguna. Artikel utama berada di sisi kiri dengan ukuran yang lebih besar, dan dua artikel lainnya ditampilkan di sisi kanan. Di bawah *featured news*, terdapat bagian *Recent News* yang memuat daftar artikel lainnya dalam bentuk *grid*. Tampilan ini dibuat responsif agar tetap nyaman diakses melalui berbagai ukuran perangkat, baik *desktop* maupun *mobile*.

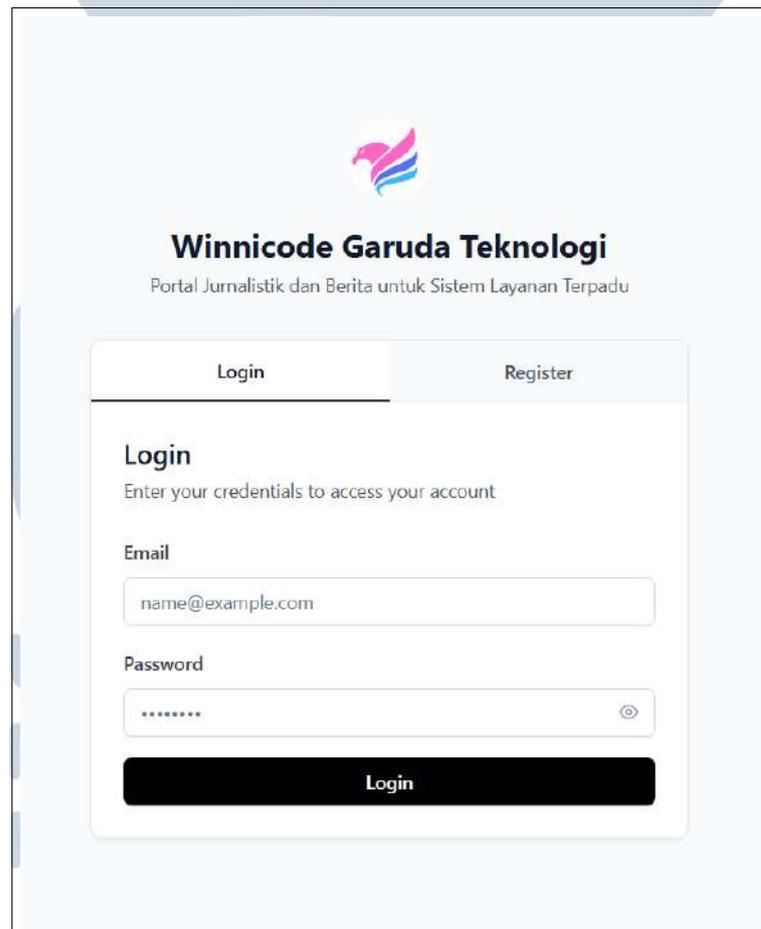
Gambar 3.9 menunjukkan menu navigasi atau *header* dari *website* portal berita yang telah dikembangkan.



Gambar 3.9. Menu Navigasi (*Header*)

Bagian atas halaman (*header*) memuat logo PT Winnicode Garuda Teknologi, menu navigasi menuju kategori-kategori berita, serta menu *More* untuk kategori tambahan. Di sisi kanan *header* terdapat ikon pencarian dan tombol akun (*profile picture* jika sudah *login*). Jika pengguna belum *login*, tombol ini akan mengarah ke halaman *login/register*. Namun jika sudah *login*, tombol ini akan menampilkan menu untuk ke halaman profil.

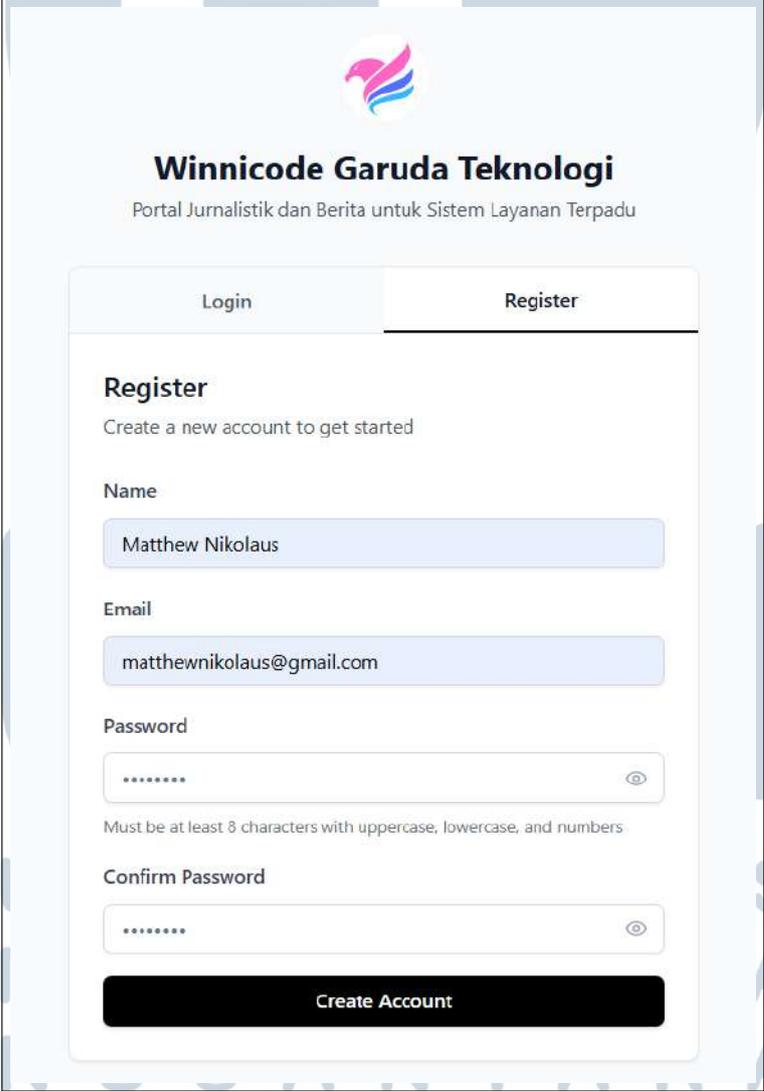
Gambar 3.10 menunjukkan halaman *login* yang telah dikembangkan. Pengguna diminta untuk memasukkan alamat *email* dan kata sandi yang telah terdaftar di *database*.



Gambar 3.10. Halaman *Login*

Pada halaman ini, terdapat dua *tab* navigasi utama, yaitu *Login* dan *Register* yang memungkinkan pengguna untuk berpindah antara formulir *login* dan pendaftaran akun baru. Sistem juga dilengkapi dengan validasi dan penanganan *error*, seperti menampilkan notifikasi ketika *email* yang dimasukkan tidak terdaftar di *database* atau *password* tidak sesuai.

Gambar 3.11 merupakan halaman *register* yang memungkinkan pengguna untuk membuat akun baru. Terdapat beberapa *input* yang harus diisi pengguna seperti nama, *email*, *password*, dan konfirmasi *password* yang akan disimpan ke *database*.



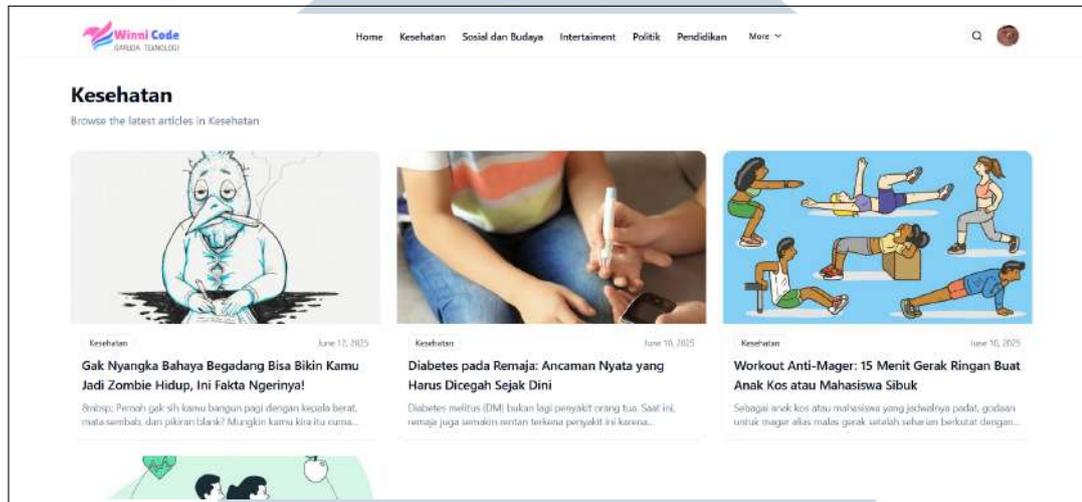
The image shows a web registration form for 'Winnicode Garuda Teknologi'. The form is titled 'Register' and includes the following fields and elements:

- Name:** Matthew Nikolaus
- Email:** matthewnikolaus@gmail.com
- Password:** Masked with dots. A note below the field states: 'Must be at least 8 characters with uppercase, lowercase, and numbers'.
- Confirm Password:** Masked with dots.
- Button:** 'Create Account' (black background, white text).

Gambar 3.11. Halaman *Register*

Gambar 3.12 merupakan halaman kategori yang menampilkan artikel-artikel

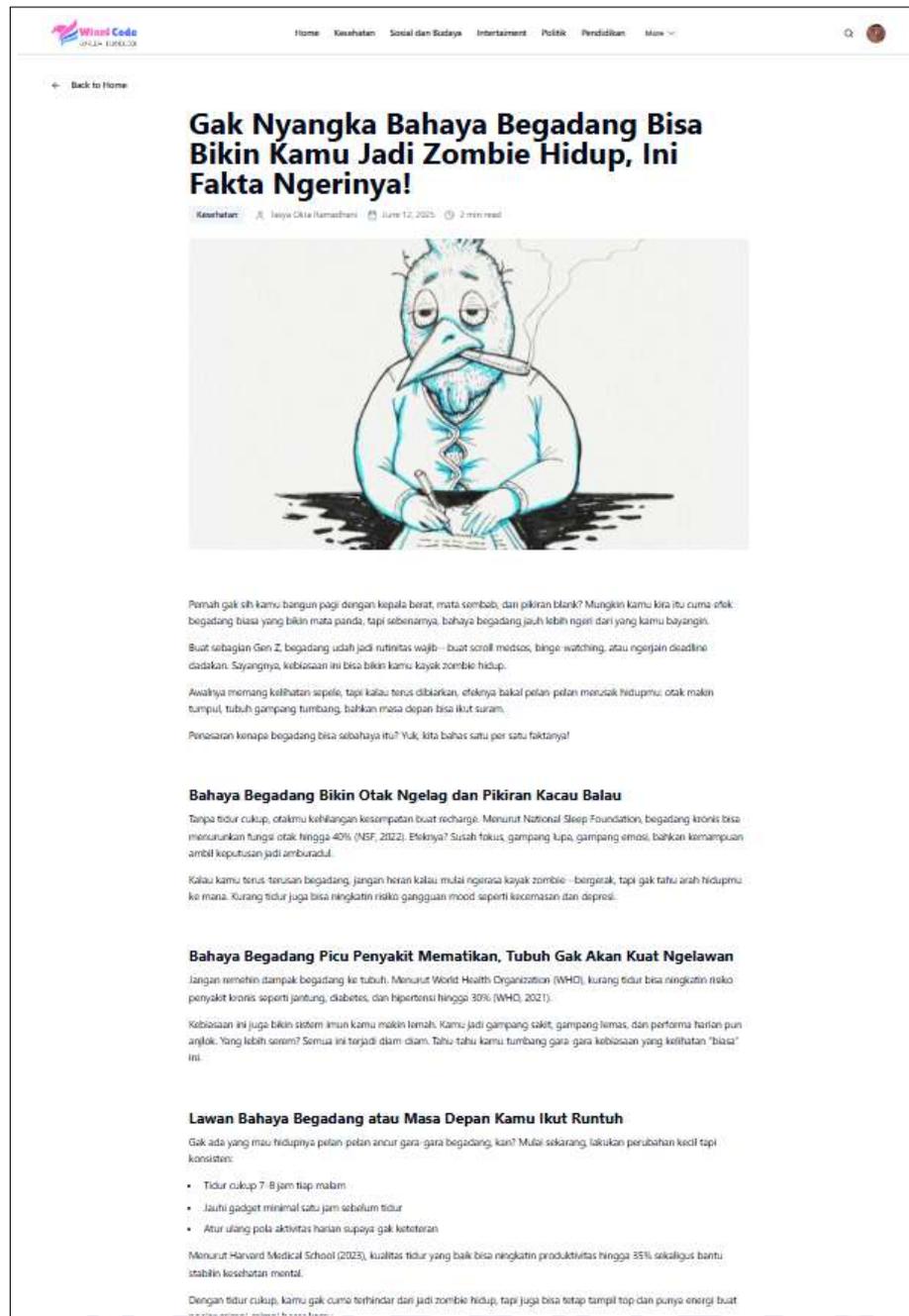
yang dikelompokkan berdasarkan topik atau kategori tertentu. Pada gambar 3.12 ditampilkan halaman kategori Kesehatan, yang dapat diakses melalui menu navigasi.



Gambar 3.12. Halaman Kategori

Gambar 3.13 menunjukkan halaman artikel yang ditampilkan saat pengguna mengklik salah satu berita dari halaman utama atau halaman kategori. Tampilan ini dirancang agar mudah dibaca dan fokus pada kenyamanan pengguna dalam mengakses konten yang panjang.

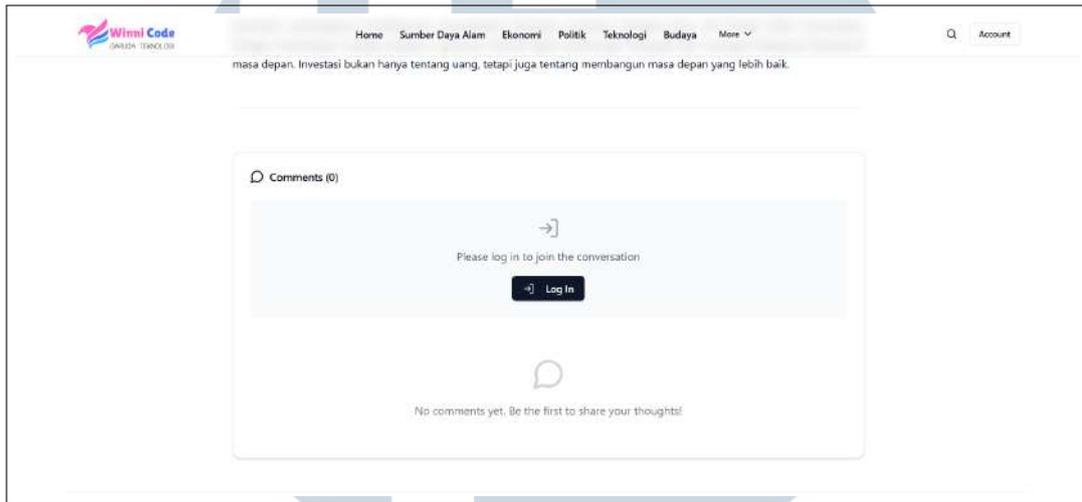
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.13. Halaman Detail Artikel

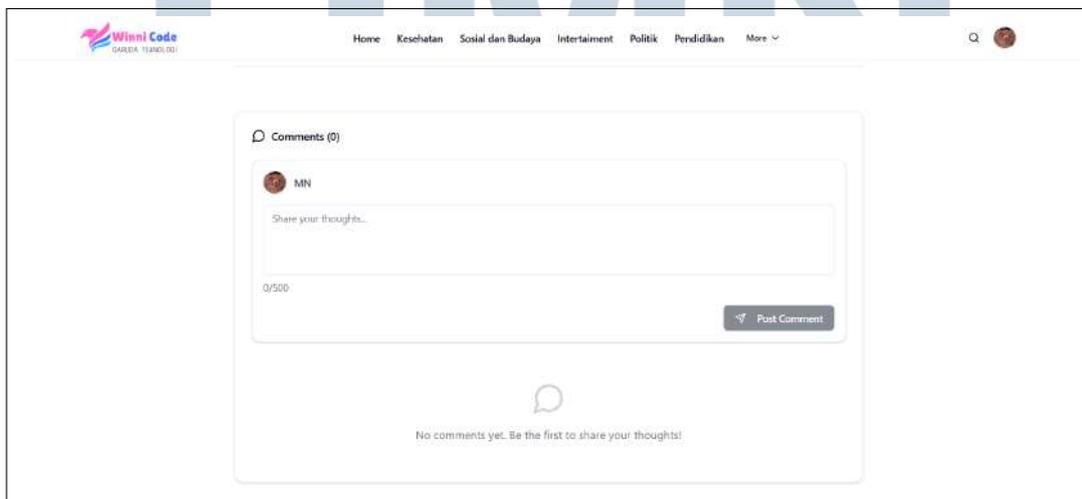
Judul artikel tertera dengan ukuran besar dan tebal di bagian atas untuk menarik perhatian pembaca. Di bawah judul, terdapat informasi penting seperti kategori artikel, penulis artikel, tanggal publikasi, serta estimasi waktu baca. Pemisahan antar paragraf dan penggunaan *heading/subheading* membuat struktur konten jelas dan mudah dipahami. Di bagian atas halaman, terdapat tombol *Back to Home* untuk memudahkan pengguna kembali ke halaman utama.

Di bawah isi artikel, terdapat bagian komentar yang memungkinkan pengguna untuk memberikan tanggapan, berdiskusi, serta berinteraksi satu sama lain terkait konten artikel yang dibaca. Gambar 3.14 menunjukkan tampilan komentar ketika pengguna belum *login*. Pengguna akan diminta untuk *login* terlebih dahulu sebelum dapat berpartisipasi dalam komentar.



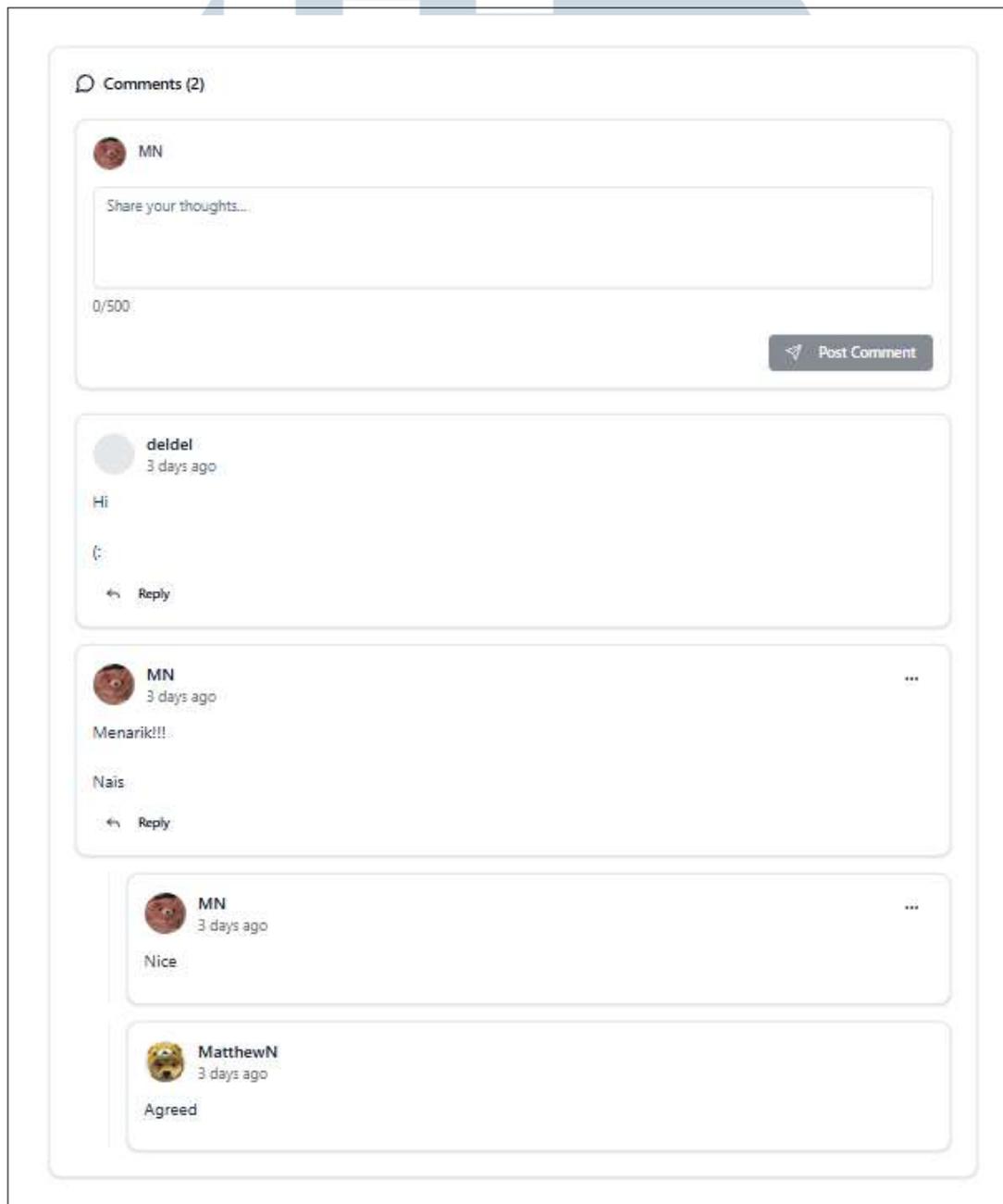
Gambar 3.14. Tampilan Komentar ketika Belum *Login*

Gambar 3.15 menunjukkan tampilan komentar ketika sebuah artikel belum memiliki komentar dari pengguna. Sistem akan menampilkan pesan informatif seperti *"No comments yet. Be the first to share your thoughts!"* untuk mengajak pengguna memberikan komentar.



Gambar 3.15. Tampilan Komentar ketika Belum Ada Komentar

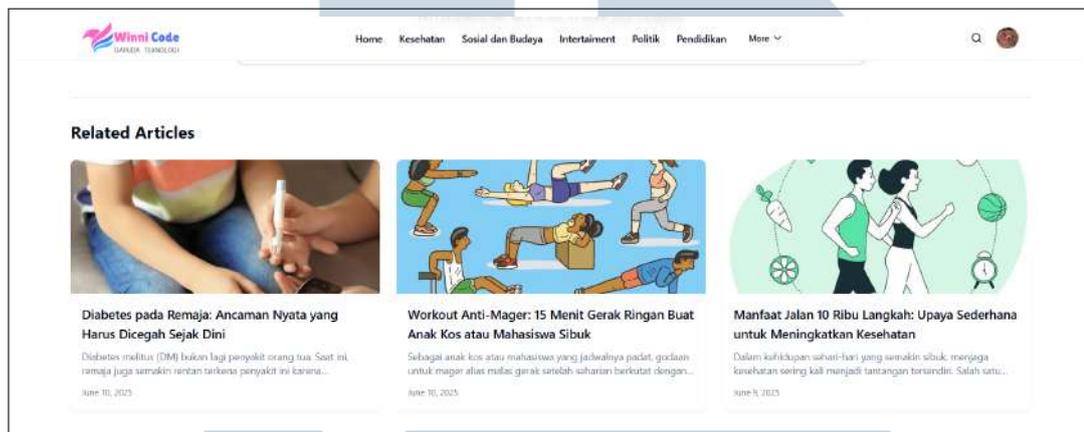
Gambar 3.16 menunjukkan tampilan komentar ketika sebuah artikel sudah terdapat komentar. Ketika sudah terdapat komentar, sistem akan menampilkan daftar komentar yang dilengkapi dengan waktu komentar dikirim dan fitur berbalas atau *reply* di setiap komentar. Pengguna juga dapat *edit* atau menghapus komentar sendiri yang telah dikirim.



Gambar 3.16. Tampilan Komentar ketika Sudah Ada Komentar

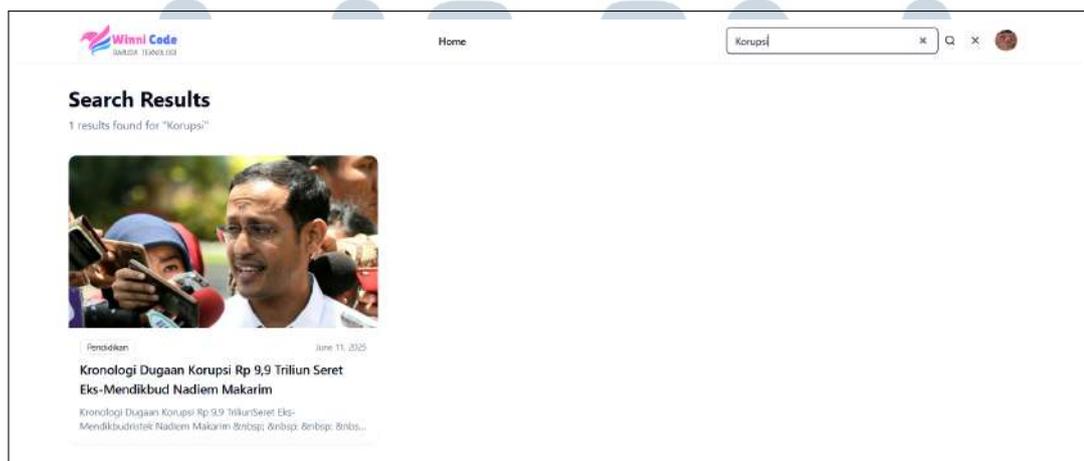
Di bawah kolom komentar, terdapat bagian *Related Articles* yang

menyarankan artikel-artikel lain yang relevan dengan kategori artikel yang sedang dibaca. Gambar 3.17 menunjukkan tampilan bagian *Related Articles*. Dengan bagian ini, pengguna yang telah selesai membaca artikel utama dapat langsung menelusuri berita lanjutan yang sejenis dengan kategori.



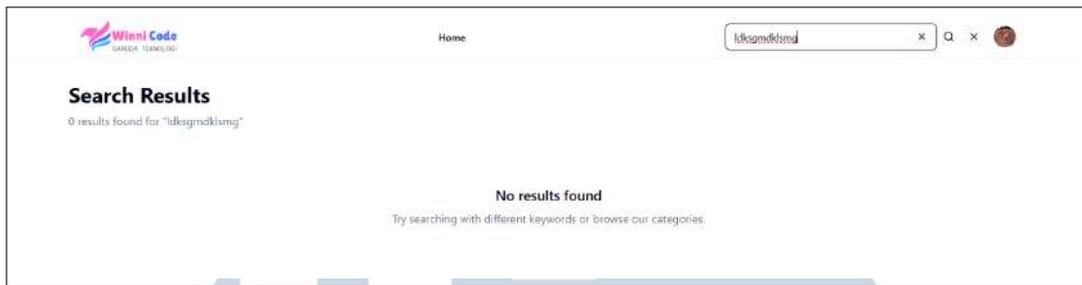
Gambar 3.17. Tampilan *Related Articles*

Gambar 3.18 merupakan tampilan hasil pencarian dengan kata kunci “Korupsi”. Halaman pencarian atau *search* akan ditampilkan saat pengguna mengetikkan kata kunci di kolom pencarian yang tersedia di bagian menu navigasi *website* dan menekan *enter* atau mengklik tombol ikon *search*.



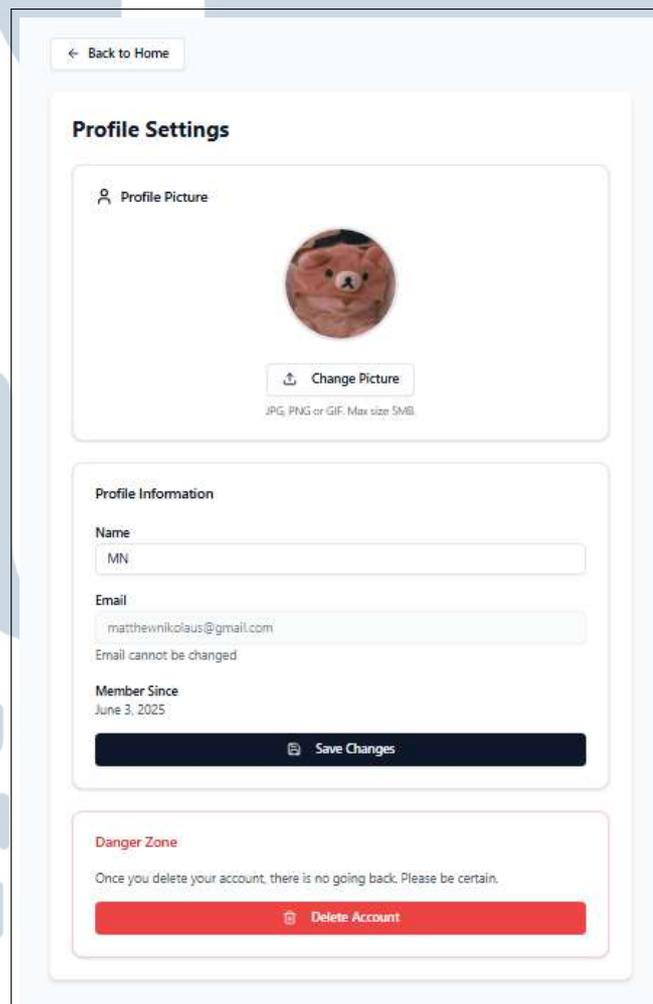
Gambar 3.18. Tampilan Fitur dan Halaman *Search*

Gambar 3.19 menunjukkan tampilan hasil pencarian ketika tidak ada artikel yang cocok dengan kata kunci. Halaman pencarian akan menampilkan pesan *No results found*.



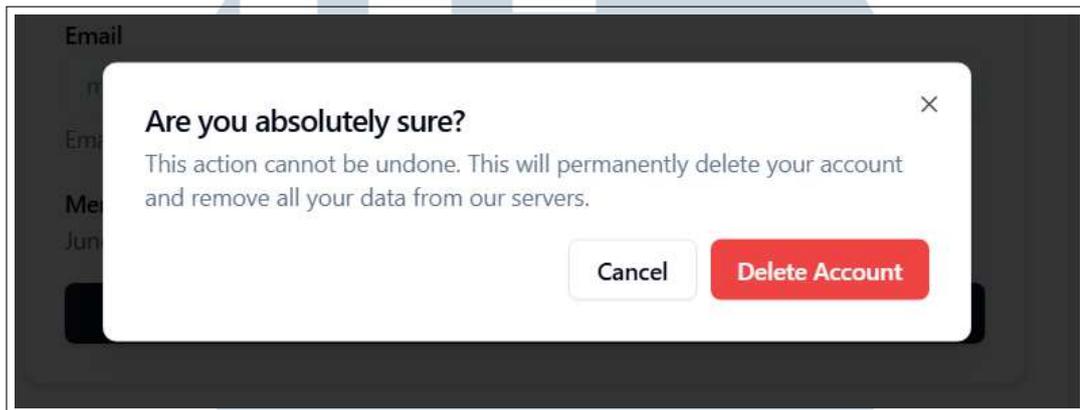
Gambar 3.19. Tampilan Fitur dan Halaman *Search* ketika *No Results Found*

Gambar 3.20 merupakan halaman profil pengguna yang memungkinkan pengguna untuk mengelola informasi data akun. Pada halaman ini, pengguna dapat mengubah foto profil, mengubah nama, melihat email yang digunakan akun pengguna, melihat tanggal akun dibuat, atau menghapus akun.



Gambar 3.20. Halaman Profil

Gambar 3.21 menunjukkan tampilan *pop-up* dengan pesan konfirmasi ketika pengguna mengklik tombol *Delete Account* di halaman profil pengguna. Jika pengguna menekan tombol *Delete Account* pada pesan konfirmasi, maka data akun pengguna akan dihapus dari *database*. Jika pengguna menekan tombol *cancel*, maka *pop-up* pesan konfirmasi akan ditutup dan penghapusan akun dibatalkan.



Gambar 3.21. Tampilan *Warning Message Delete Account*

3.4 Kendala dan Solusi yang Ditemukan

Selama proses pengembangan proyek, ditemukan beberapa kendala yang menghambat kelancaran implementasi antarmuka *website* portal berita ini. Berikut adalah kendala-kendala yang dihadapi selama proses kerja magang:

1. Kesalahan format API saat melakukan *fetching* data. Kendala ini muncul karena kurangnya pengalaman dalam menggunakan dan memahami struktur API yang disediakan. Hal ini menyebabkan data tidak dapat ditampilkan dengan benar di halaman *website*.
2. Gambar dari API tidak muncul di halaman *website*. Gambar-gambar yang diambil dari API milik perusahaan tidak dapat ditampilkan di halaman *website*. Setelah ditelusuri, hal ini disebabkan oleh kebijakan CORS (*Cross-Origin Resource Sharing*) yang memblokir pemuatan gambar dari domain yang berbeda.
3. Kurangnya pengalaman dalam desain UI/UX. Keterbatasan pengalaman dalam bidang UI/UX menyebabkan kesulitan dalam merancang tampilan antarmuka yang menarik, fungsional, dan sesuai dengan standar desain modern.

4. *Profile picture* di *header* tidak langsung *update* setelah *upload* gambar baru. Setelah pengguna mengunggah gambar profil baru, gambar di bagian *header* tidak langsung berubah dan tetap menampilkan gambar lama akibat *cache browser*.

Berikut adalah solusi-solusi yang berhasil diterapkan untuk mengatasi kendala yang ditemukan:

1. Setelah melakukan diskusi dengan *supervisor* dan melakukan riset mandiri menggunakan Postman sesuai saran dari *supervisor*, struktur respons API berhasil dianalisis. Dengan memahami format data yang dikembalikan oleh server, akhirnya proses *fetching* data berhasil dilakukan dengan benar.
2. Dengan melakukan riset lebih lanjut, ditemukan bahwa masalah ini dapat diatasi dengan menambahkan atribut *referrerPolicy="no-referrer"* pada elemen gambar. Atribut ini mencegah *browser* mengirim informasi *referrer* saat melakukan *request* gambar, yang kemudian memungkinkan gambar untuk *loading* dengan benar di halaman *website*.
3. Melakukan riset mengenai prinsip-prinsip dasar UI/UX melalui referensi *online* dan mendapatkan inspirasi dari desain *website* populer. Selain itu, *feedback* rutin dari *supervisor* membantu dalam pembuatan desain. Pembuatan *wireframe* dengan menggunakan Figma juga dimanfaatkan untuk merancang dan memvisualisasikan antarmuka sebelum diimplementasikan ke dalam kode.
4. Setelah melakukan riset, solusi yang diterapkan adalah menambahkan parameter versi ke URL gambar dengan menggunakan *timestamp updatedAt* dari data pengguna. Dengan demikian, URL menjadi unik setiap kali gambar diperbarui, sehingga *browser* tidak menggunakan *cache* lama dan selalu menampilkan gambar terbaru.