

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama kegiatan magang, mahasiswa ditempatkan sebagai *Full Stack Developer* yang berkontribusi langsung dalam proses pengembangan sistem informasi koperasi berbasis web di bawah bimbingan Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Multimedia Nusantara. Penempatan ini berada dalam koordinasi tim pengembang internal yang berfokus pada implementasi solusi digital untuk mendukung operasional koperasi Wiyata Mandala.

Sebagai *Full Stack Developer*, mahasiswa bertanggung jawab dalam membangun serta memperbaiki beberapa fitur utama sistem koperasi, meliputi sisi antarmuka pengguna menggunakan *React.js* dan *Tailwind CSS*, serta pengolahan data di sisi server menggunakan *Express.js* dan *PostgreSQL*. Pengembangan dilakukan dengan pendekatan *modular*, yang mencakup *UI/UX Responsive*, fitur profil koperasi, dan sistem voucher berbasis koperasi.

Koordinasi kerja dilakukan bersama supervisor proyek dan rekan tim pengembang lainnya. Komunikasi rutin dilakukan melalui pertemuan daring menggunakan *Zoom* dan diskusi informal melalui *WhatsApp group*. Pembagian tugas disusun berdasarkan modul fitur, sehingga masing-masing anggota memiliki fokus kerja yang jelas namun tetap saling terintegrasi.

Penggunaan *GitHub* sebagai wadah kolaborasi memungkinkan pengelolaan kode yang sistematis, dengan pendekatan *branch* per fitur untuk meminimalisir konflik kode. Proses *review and merge* dilakukan secara berkala agar pengembangan tetap berjalan lancar dan sesuai target. Selain itu, mahasiswa juga ikut serta dalam beberapa sesi diskusi langsung dengan pembimbing dari LPPM untuk menyampaikan perkembangan fitur serta mendapatkan arahan terkait kebutuhan tambahan dari pihak koperasi.

Pada beberapa kesempatan, tim juga menjalin komunikasi langsung dengan pengurus koperasi guna memahami kebutuhan bisnis dan menyesuaikan alur sistem dengan praktik yang berlaku di lapangan. Pelaksanaan kerja magang dilakukan secara *Work From Home (WFH)* sesuai dengan arahan dari pembimbing proyek, dan berlangsung mulai tanggal 03 Februari hingga akhir 12 Juni 2025, dengan waktu

kerja aktif dari Senin hingga Jumat, pukul 08.00 hingga 17.00 WIB.

Seluruh kegiatan pengembangan tetap dilakukan secara profesional dengan pemantauan dan evaluasi berkala dari pihak pembimbing dan pengurus proyek. Koordinasi kerja dilakukan melalui komunikasi daring, menggunakan platform seperti *Google Meet* dan *WhatsApp Group*. Diskusi teknis maupun review desain dilakukan secara berkala, terutama saat peralihan modul atau sebelum integrasi fitur baru. Selain itu, dilakukan juga sesi evaluasi rutin bersama pembimbing dari LPPM, guna memastikan setiap bagian proyek diselesaikan sesuai dengan *timeline* dan kebutuhan koperasi.

Struktur organisasi proyek ini berada di bawah tanggung jawab langsung Bapak Winarno, selaku dosen pembimbing dari LPPM UMN, yang memberikan pengarahan, pengawasan, serta evaluasi terhadap progres pengembangan sistem. Di samping itu, terdapat koordinasi teknis dengan pihak koperasi untuk memastikan bahwa fitur-fitur yang dikembangkan relevan dengan kebutuhan operasional koperasi di lapangan.

Adapun alur koordinasi dalam proyek ini dapat dijabarkan sebagai berikut: Pembimbing Proyek (LPPM - Bapak Winarno):

1. Memberikan arahan umum terkait pengembangan sistem, melakukan evaluasi berkala, serta memastikan mahasiswa menjalankan tanggung jawab sesuai peran.
2. Tim Pengurus Koperasi Wiyata Mandala: Memberikan masukan kebutuhan fungsional sistem, serta melakukan pengujian dan validasi fitur yang dikembangkan.
3. Mahasiswa Magang (Full stack Developer): Melaksanakan pengembangan *UI/UX*, mengimplementasikan fitur-fitur yang diminta, serta melakukan *debugging* dan penyempurnaan sistem berdasarkan hasil evaluasi.

Dengan struktur yang jelas serta koordinasi yang terorganisir, proses pengembangan sistem koperasi ini dapat berjalan secara efektif dan berkontribusi nyata terhadap digitalisasi layanan koperasi, khususnya dalam meningkatkan transparansi dan efisiensi pengelolaan keuangan anggota.

3.2 Tugas yang Dilakukan

Tugas yang dikerjakan sepanjang pelaksanaan program magang ini berfokus pada perancangan antarmuka pengguna yang responsif (*UI*), penyesuaian desain *UX*

untuk seluruh page peran pengguna (Anggota dan Pengurus), serta pengembangan dan integrasi fitur Voucher dalam sistem informasi koperasi berbasis web.

Tahapan pekerjaan diawali dengan melakukan observasi terhadap struktur sistem eksisting dan identifikasi masalah dalam tampilan serta alur interaksi pengguna. Berdasarkan temuan tersebut, dilakukan redesain tampilan halaman utama untuk ketiga peran pengguna, yaitu Anggota, Pengurus, dan Admin, agar lebih intuitif, efisien, dan dapat diakses secara optimal di berbagai ukuran layar.

Untuk meningkatkan kenyamanan dan aksesibilitas pengguna, halaman-halaman utama seperti *dashboard* anggota, manajemen akun, riwayat simpan pinjam, serta pengelolaan data anggota dan laporan transaksi didesain ulang menggunakan pendekatan *UI/UX Responsive* dan teknologi berbasis komponen. Komponen-komponen seperti sidebar dinamis, tabel data interaktif, dan *form input* adaptif disusun ulang agar dapat diakses baik melalui desktop maupun perangkat *mobile*.

Pada sisi pengembangan fitur, dilakukan perancangan dan penambahan modul Voucher, yang mencakup pembuatan halaman daftar voucher, pengelolaan masa aktif, penukaran voucher oleh anggota, dan integrasi status penggunaan voucher oleh pengurus. Penyesuaian logika *backend* juga dilakukan untuk memastikan fungsi validasi dan distribusi voucher berjalan sesuai dengan permintaan pihak koperasi.

Secara keseluruhan, fokus pekerjaan selama magang ini menitikberatkan pada pengembangan antarmuka yang *user-friendly*, fleksibel, efisien dan integrasi fitur yang mendukung digitalisasi layanan koperasi secara menyeluruh, khususnya dalam hal peningkatan pengalaman pengguna dan efisiensi pengelolaan program voucher. Sistem database menggunakan *PostgreSQL* dan *database management* menggunakan *dashboard PGAdmin4*.

Struktur database *PostgreSQL* dimodifikasi untuk menyesuaikan dengan kebutuhan entitas voucher, termasuk relasi dengan tabel anggota, pengurus, dan log transaksi.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan magang yang dilakukan di LPPM UMN dilakukan selama 4 bulan lebih terhitung dari 03 Februari 2025 hingga 12 Juni 2025. Pelaksanaan magang dilaksanakan selama 5 hari kerja dari Senin hingga Jumat dengan jam kerja 7-8 Jam sehari.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Observasi awal sistem koperasi, mengenali kebutuhan fitur, setup environment pengembangan menggunakan <i>React.js</i> dan <i>Express.js</i> , serta memahami struktur database <i>PostgreSQL</i> .
2	Mendesain struktur halaman <i>dashboard</i> , membuat sketsa <i>UI</i> awal untuk halaman profil pengguna dan voucher, serta mulai membangun routing <i>Frontend</i> dasar.
3	Implementasi halaman profil pengguna, integrasi dengan <i>backend Express.js</i> , membuat fitur edit data pengguna serta validasi formulir di sisi <i>Fronten</i>
4	Finalisasi modul profil pengguna dan uji coba penyimpanan data ke database. Perapihan <i>UI</i> agar responsif di berbagai perangkat.
5	Mendesain <i>UI</i> halaman voucher anggota, mulai implementasi <i>backend</i> : pembuatan tabel voucher, pengelolaan data voucher di <i>PostgreSQL</i> .
6	Pengembangan fitur distribusi voucher berdasarkan saldo tabungan anggota, serta validasi masa berlaku dan ketentuan penggunaan voucher.
7	Implementasi <i>Fronten</i> halaman daftar voucher, menampilkan status voucher aktif dan kedaluwarsa, serta perbaikan tampilan responsif
8	Mendesain dan membangun <i>UI</i> struktur pembagian SHU (Sisa Hasil Usaha), mulai dari form input persentase pembagian hingga tampilan rekap SHU
9	Menghubungkan struktur SHU ke <i>backend</i> , menyimpan persentase ke database dan menampilkan hasil distribusi otomatis ke setiap anggota.
10	Uji coba dan <i>debugging</i> seluruh alur struktur SHU. Memastikan logika pembagian SHU sesuai ketentuan koperasi dan dapat dimodifikasi admin.

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

Minggu Ke -	Pekerjaan yang dilakukan
11	Peningkatan tampilan mobile untuk halaman-halaman utama: profil, voucher, dan SHU. Penerapan layout fleksibel dan komponen reusable.
12	Penyempurnaan tampilan responsif seluruh modul. Melakukan <i>testing</i> di berbagai resolusi layar dan memperbaiki bug layout.
13	Persiapan integrasi halaman tambahan, dokumentasi kode <i>Fronten</i> , serta perapihan komponen <i>UI</i> sesuai standar koperasi.
14	Finalisasi modul voucher: fitur edit dan nonaktifkan voucher, serta konfirmasi dari sisi admin.
15	Revisi minor berdasarkan feedback supervisor, melakukan penyesuaian teks, validasi data tambahan, serta backup database
16	Review dan dokumentasi sistem yang telah dikembangkan, menyusun catatan penggunaan fitur dan instruksi operasional
17	Penyusunan laporan magang, termasuk penyusunan Bab 3: uraian kegiatan, struktur teknologi, dan hasil kerja pada setiap modul utama

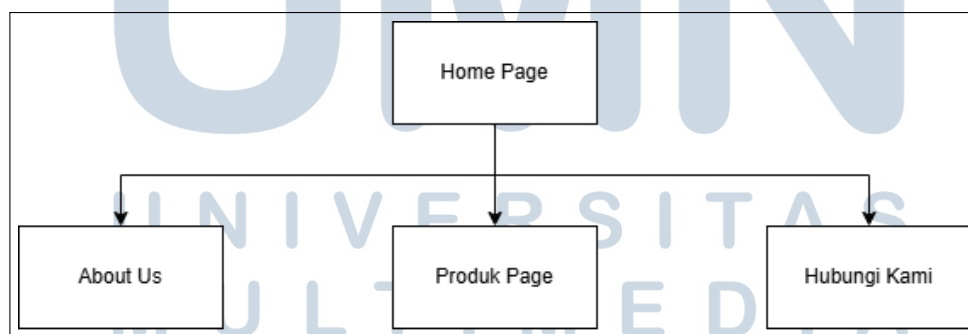
3.3.1 Mempelajari sistem koperasi dan mengenali kebutuhan fitur pengembangan koperasi

Pada tahap awal pelaksanaan magang, terlebih dahulu melakukan studi terhadap sistem kerja Koperasi Wiyata Mandala untuk memahami alur kerja, kebutuhan pengguna, dan proses bisnis yang ada. Tujuan dari tahap ini adalah agar pengembangan sistem berbasis web dapat disesuaikan dengan kebutuhan riil pengguna, serta memberikan dampak yang nyata dalam digitalisasi layanan koperasi. Mahasiswa mengidentifikasi beberapa kebutuhan utama yang belum terpenuhi dalam sistem koperasi sebelumnya, seperti tidak adanya sistem voucher yang terintegrasi dengan data tabungan anggota, serta belum adanya fitur distribusi Sisa Hasil Usaha (SHU) secara otomatis. Selain itu, tampilan antarmuka pengguna masih belum responsif dan belum memanfaatkan *Framework* modern. Berdasarkan hasil pengamatan dan diskusi bersama pembimbing lapangan serta pihak koperasi, mahasiswa menetapkan beberapa kebutuhan fitur yang perlu dikembangkan, antara lain:

1. Fitur distribusi SHU berdasarkan persentase yang ditentukan pengurus.
2. Fitur voucher belanja yang terhubung dengan saldo tabungan anggota dan memiliki sistem tiering.
3. Sistem autentikasi dan pengelolaan profil anggota koperasi.
4. Peningkatan antarmuka pengguna agar lebih modern dan responsif.

Untuk mendukung pengembangan fitur-fitur tersebut, mahasiswa memilih *React.js* sebagai library utama dalam pengembangan *Framework* karena kemampuannya dalam membangun antarmuka pengguna yang dinamis dan modular. mahasiswa juga menggunakan *Tailwind CSS* untuk mempercepat proses styling dengan utilitas yang efisien dan mendukung desain responsif. Di sisi *backend*, mahasiswa menggunakan *Express.js* sebagai *Framework* Node.js untuk membangun *RESTful API* yang dapat menangani permintaan dari *Framework* dengan efisien. Sedangkan untuk penyimpanan data, digunakan *PostgreSQL* sebagai database relasional karena kestabilan dan kemampuannya dalam menangani transaksi data yang kompleks, seperti penghitungan SHU dan validasi voucher anggota. Dengan pendekatan ini, mahasiswa dapat merancang dan mengembangkan sistem yang tidak hanya memenuhi kebutuhan teknis, tetapi juga mudah digunakan oleh anggota dan pengurus koperasi.

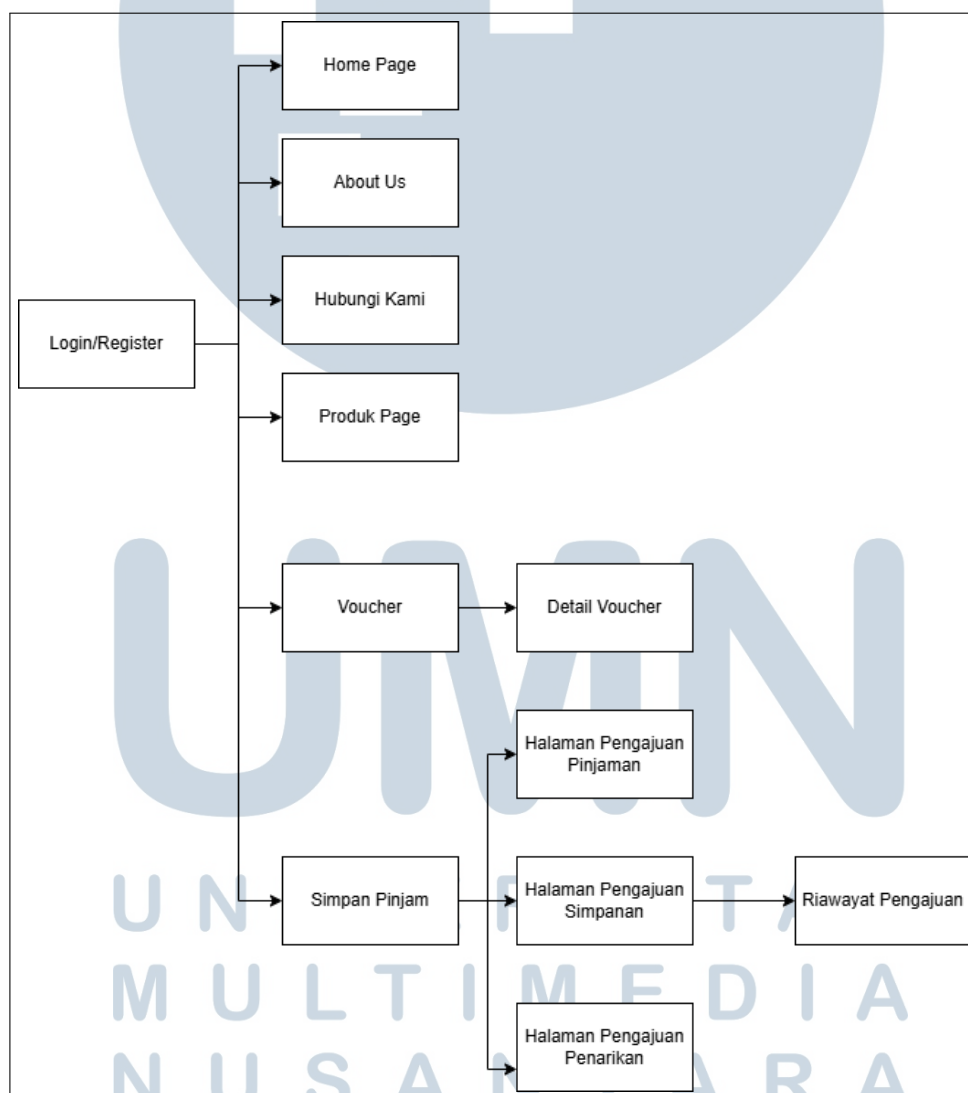
A Sitemap



Gambar 3.1. Sitemap Website Koperasi Wiyata Mandala Sebelum *Login*

Diagram ini menunjukkan halaman-halaman yang dapat diakses oleh pengguna tanpa *login*:

1. *Login/register*: Titik masuk utama bagi pengguna untuk membuat akun atau masuk ke sistem.
2. Home Page: Beranda publik koperasi yang menampilkan informasi umum.
3. About Us: Informasi mengenai latar belakang koperasi.
4. Produk Page: Menampilkan produk/jasa yang ditawarkan koperasi.
5. Hubungi Kami: Formulir kontak atau informasi untuk menghubungi pihak koperasi.

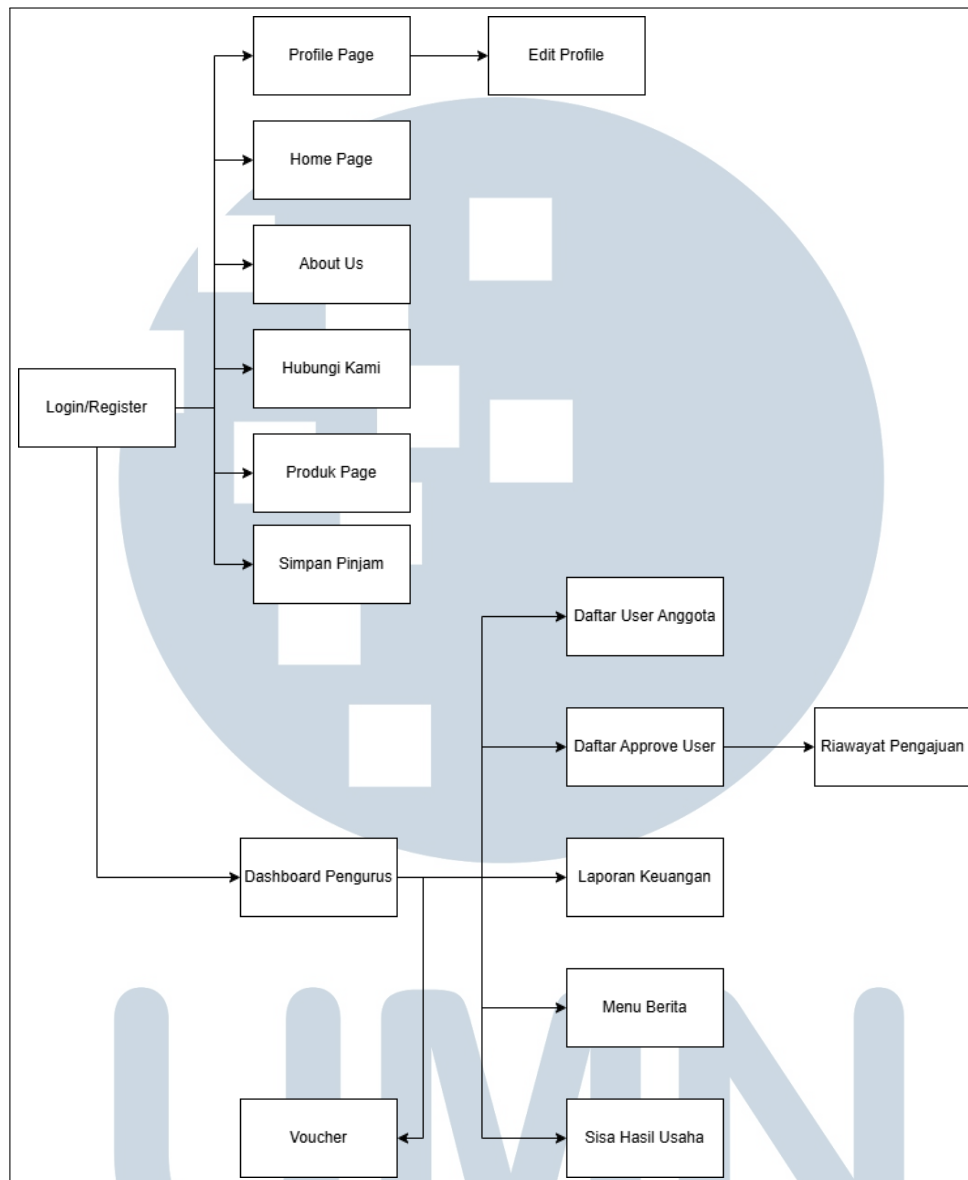


Gambar 3.2. Sitemap Website Koperasi Wiyata Mandala User Anggota

Setelah *login* sebagai anggota, pengguna memiliki akses ke berbagai fitur koperasi:

1. Profile Page dan Edit Profile: Menampilkan dan mengubah data diri anggota.
2. Simpan Pinjam: Halaman utama simpan pinjam yang menghubungkan ke :
 - Halaman Pengajuan Pinjaman
 - Halaman Pengajuan Simpanan
 - Halaman Pengajuan Penarikan
 - Riwayat Pengajuan
3. Halaman Voucher : tampilan Voucher yang bisa di gunakan anggota untuk belanja
 - Detail Voucher: Menampilkan detail voucher berdasarkan tier saldo.
4. Produk Page, Hubungi Kami, About Us, dan Home Page: Masih dapat diakses sebagai informasi tambahan.





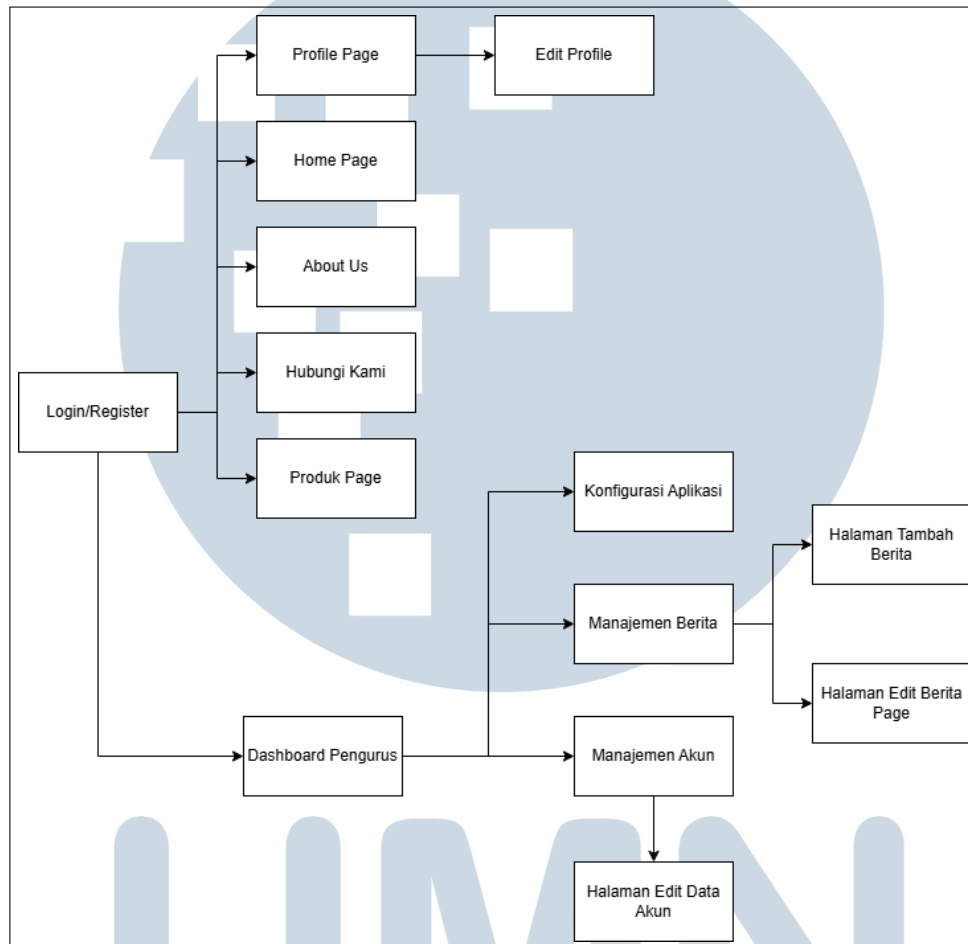
Gambar 3.3. Sitemap Website Koperasi Wiyata Mandala User Pengurus

Pengurus koperasi memiliki akses *dashboard* dengan fitur administratif:

1. *Dashboard* Pengurus: Menu utama.
2. Daftar User Anggota: Melihat data seluruh anggota.
3. Daftar Approve User: Validasi pendaftaran pengguna baru.
4. Riwayat Pengajuan: Monitoring pengajuan simpan/pinjam.
5. Laporan Keuangan: Rekapitulasi transaksi dan keuangan koperasi.

6. Menu Berita: Manajemen konten berita koperasi.

7. Sisa Hasil Usaha (SHU): Menampilkan data distribusi SHU anggota.



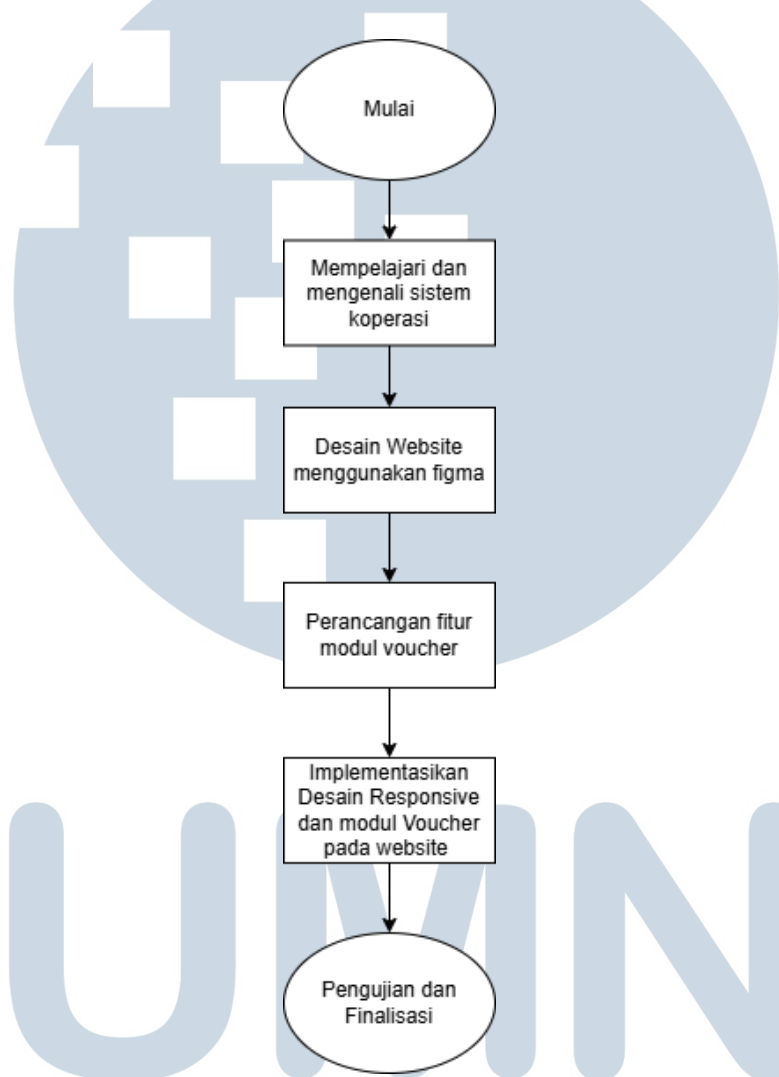
Gambar 3.4. Sitemap Website Koperasi Wiyata Mandala User Admin

Sitemap untuk level admin yang bertanggung jawab terhadap konfigurasi aplikasi:

1. Konfigurasi Aplikasi: Pengaturan teknis atau umum aplikasi.
2. Manajemen Akun:
 - Halaman Edit Data Akun: Ubah data akun anggota dan pengurus.
3. Manajemen Berita:
 - Halaman Tambah Berita: Buat artikel berita baru.
 - Halaman Edit Berita: Ubah isi berita yang sudah dibuat.

Admin memiliki peran lebih teknis dan struktural dalam pengelolaan sistem.

B Flowchart Proses Pengembangan Website Koperasi



Gambar 3.5. Flowchart Proses Pengembangan Website Koperasi Wiyata Mandala

Flowchart pada Gambar 3.5 menggambarkan tahapan proses yang dilakukan dalam pengembangan website Koperasi Wiyata Mandala, khususnya dalam pembuatan desain *UI/UX* yang responsif serta pengembangan fitur modul voucher. Berikut adalah uraian dari setiap tahapan dalam flowchart:

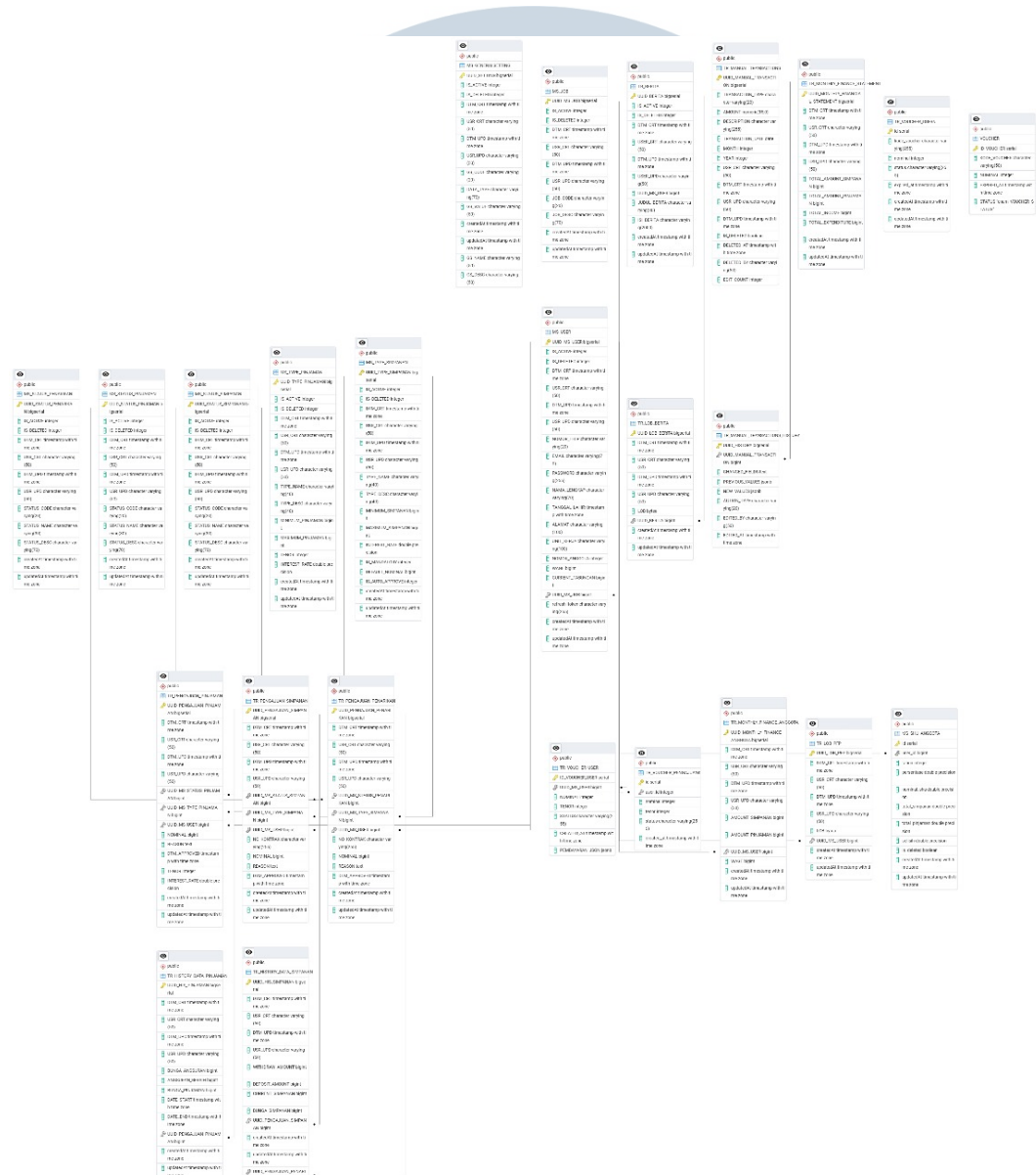
1. Mempelajari dan Mengenali Sistem Koperasi Pada tahap ini dilakukan observasi dan pengumpulan informasi mengenai bagaimana sistem koperasi

berjalan, termasuk proses tabungan, pencatatan transaksi, distribusi SHU, dan alur kerja pengurus serta anggota.

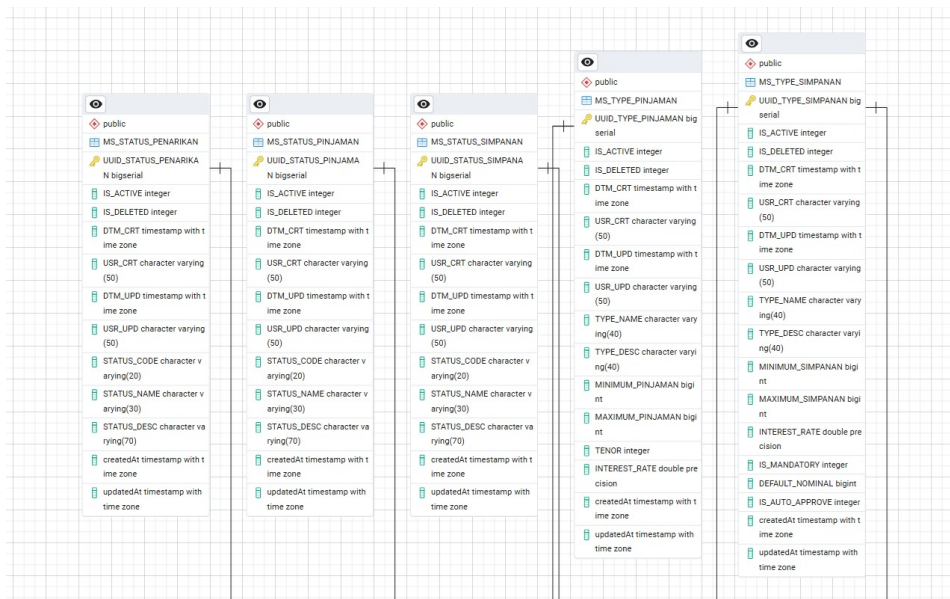
2. Desain Website Menggunakan Figma Setelah memahami sistem koperasi, dilakukan perancangan antarmuka menggunakan Figma. Desain ini mencakup halaman *login/register*, *dashboard* pengurus, halaman profil, halaman voucher, dan halaman pengajuan simpan pinjam. Fokus utama adalah pada aspek usability dan responsivitas antarmuka.
3. Perancangan Fitur Modul Voucher Modul voucher dirancang dengan mempertimbangkan logika bisnis koperasi. Anggota akan menerima voucher berdasarkan saldo tabungan, masa berlaku voucher, dan syarat transaksi (misalnya hanya untuk pembelian minimal Rp100.000 di Alfamart). Desain mencakup *UI* serta alur *backend* sederhana untuk validasi dan penyimpanan data.
4. Implementasi Desain Responsif dan Modul Voucher Setelah desain final disetujui, proses implementasi dimulai menggunakan teknologi seperti HTML, TailwindCSS, *React.js* untuk *Framework*, serta *Express.js* dan *PostgreSQL* untuk *backend*. Modul voucher diintegrasikan ke dalam *dashboard* anggota dan dikaitkan dengan data tabungan.
5. Pengujian dan Finalisasi Tahapan akhir adalah pengujian (*testing*) untuk memastikan bahwa seluruh fitur, termasuk *UI* responsif dan modul voucher, berjalan sebagaimana mestinya di berbagai perangkat (laptop, tablet, dan smartphone). Setelah itu dilakukan finalisasi dan dokumentasi hasil kerja.

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

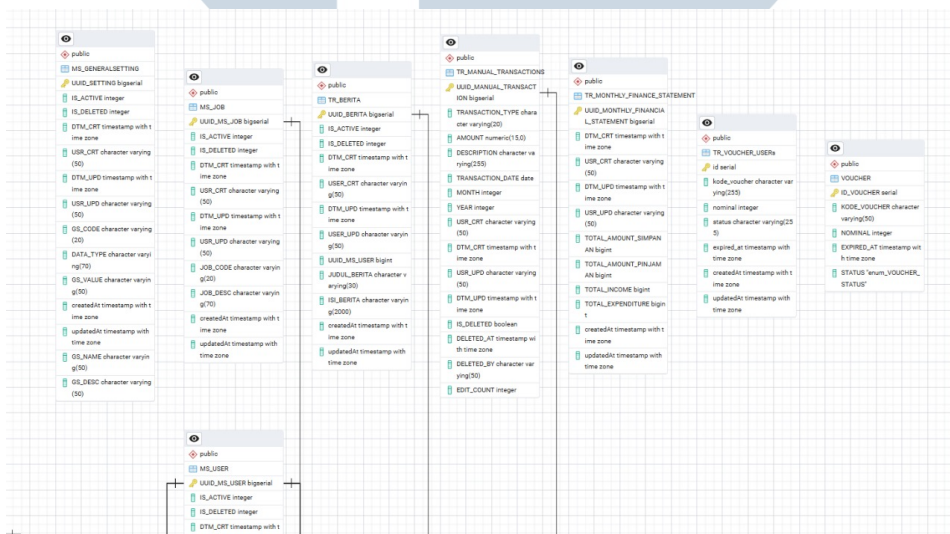
C Database



Gambar 3.6. ERD Website Koperasi Wiyata Mandala

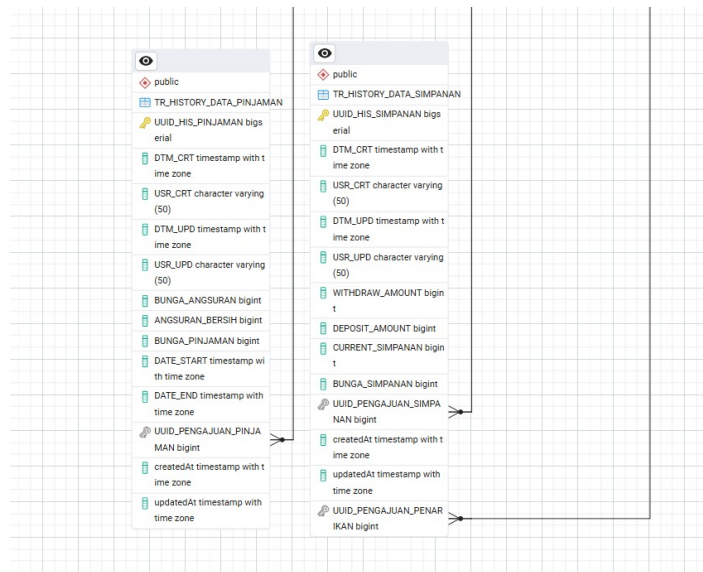


Gambar 3.7. ERD Website Koperasi Wiyata Mandala

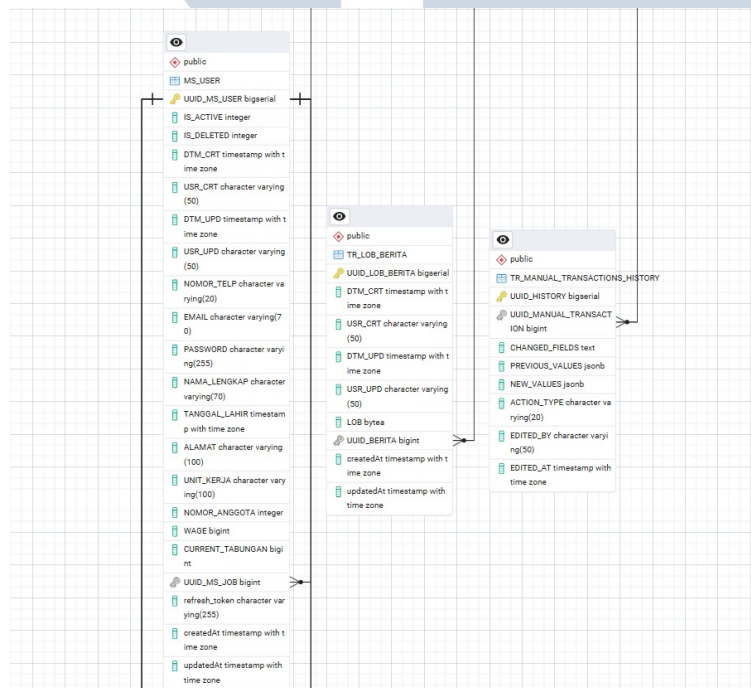


Gambar 3.8. ERD Website Koperasi Wiyata Mandala

UNIVERSITAS
MULTIMEDIA
NUSANTARA

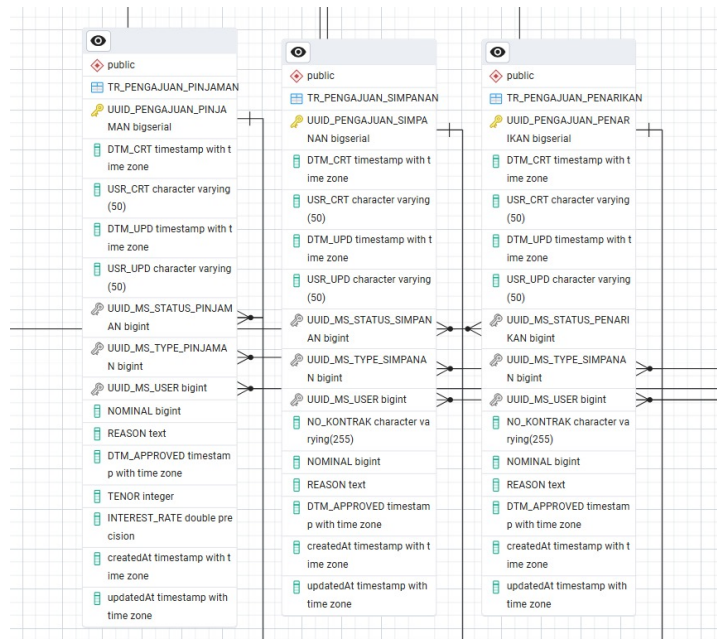


Gambar 3.9. ERD Website Koperasi Wiyata Mandala

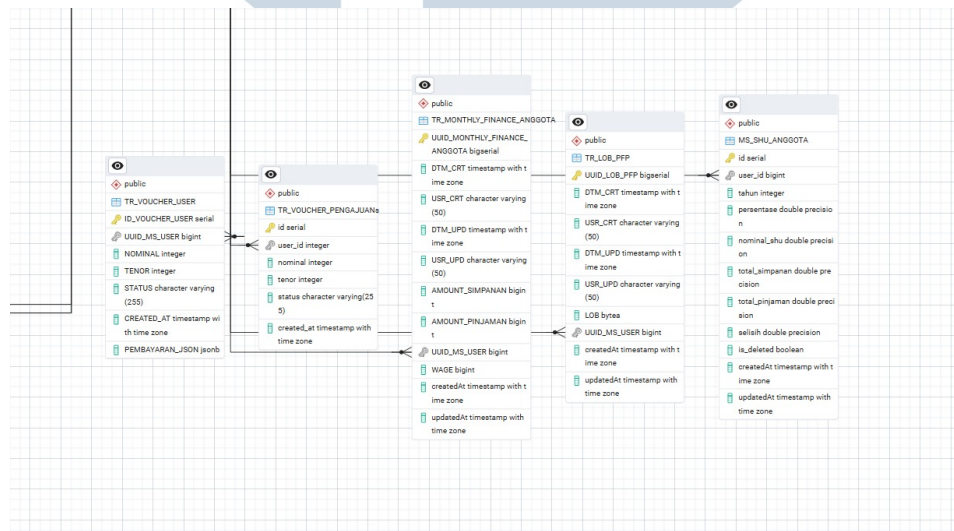


Gambar 3.10. ERD Website Koperasi Wiyata Mandala

MULTIMEDIA
NUSANTARA



Gambar 3.11. ERD Website Koperasi Wiyata Mandala



Gambar 3.12. ERD Website Koperasi Wiyata Mandala

3.4 Pengembangan UI/UX Responsive Koperasi Wiyata Mandala

Pada tahap ini pengembangan antarmuka pengguna (User Interface) dan pengalaman pengguna (User Experience) mulai dilakukan dengan standar UI/UX yang responsif dimana salah satu aspek penting dalam pengembangan sistem berbasis web atau aplikasi digital. Dalam proses magang di Koperasi Wiyata Mandala, mahasiswa terlibat dalam kegiatan pengembangan dan penyempurnaan

desain antarmuka agar dapat beradaptasi dengan berbagai ukuran layar perangkat, seperti desktop, tablet, dan ponsel pintar.

Pengembangan *UI/UX* responsif dilakukan dengan memperhatikan prinsip desain yang adaptif, pemilihan komponen visual yang konsisten, serta penerapan *Framework* atau library yang mendukung tampilan responsif, seperti Bootstrap atau CSS Grid. Tujuan utamanya adalah memastikan bahwa tampilan aplikasi tetap fungsional, menarik, dan mudah digunakan oleh pengguna di berbagai perangkat.

Dengan adanya pengembangan *UI/UX* yang responsif, diharapkan sistem koperasi dapat meningkatkan kenyamanan dan efisiensi dalam penggunaan aplikasi oleh anggota dan pengurus koperasi.

3.4.1 Pengembangan Navigasi Bar (Navbar)

Navbar merupakan bagian atas dari halaman website yang berfungsi sebagai navigasi utama antar halaman. Desain terbaru dibuat lebih modern, responsif, dan mudah digunakan.

- **Logo dan Nama Koperasi:** Diletakkan di sisi kiri navbar untuk memperkuat identitas koperasi.
- **Menu Navigasi:** Terdiri dari tautan seperti Beranda, Tentang Kami, Produk, dan Hubungi Kami.
- **Tombol Masuk:** Terletak di sisi kanan, berupa tombol *login* dengan ikon pengguna yang menonjol menggunakan warna biru terang.



Gambar 3.13. Tampilan Navbar Sebelum *revamp* pada Website Koperasi Wiyata Mandala



Gambar 3.14. Tampilan Navbar Sesudah *revamp* pada Website Koperasi Wiyata Mandala

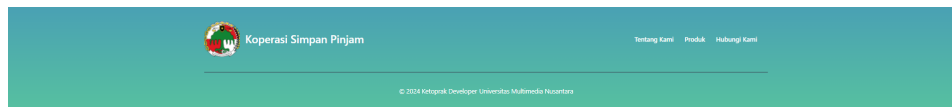
Berikut ringkasan komponennya:

- `<header>` menggunakan kelas `bg-gradient-to-b` untuk memberikan efek gradasi biru sebagai latar.
- Logo dan teks *Koperasi Wiyata Mandala* berada dalam elemen `<h1>` di sisi kiri.
- Daftar menu navigasi disusun dalam tag `<ul>` dan diberi efek hover underline.
- Tombol *login* berada di sisi kanan, dengan class khusus untuk efek animasi dan ikon pengguna menggunakan elemen `<i>` atau `svg`.

Struktur ini memastikan navbar tampil bersih, intuitif, dan tetap responsif di berbagai ukuran layar.

3.4.2 Footer Website

Footer merupakan bagian paling bawah dari halaman website yang berfungsi menampilkan identitas koperasi, navigasi tambahan, serta informasi hak cipta. Pada versi terbaru, tampilan footer dirancang dengan tema modern dan responsif menggunakan kombinasi warna gradasi biru yang profesional.



Gambar 3.15. Tampilan Footer Sebelum *revamp* pada Website Koperasi Wiyata Mandala



Gambar 3.16. Tampilan Footer Sesudah *revamp* pada Website Koperasi Wiyata Mandala

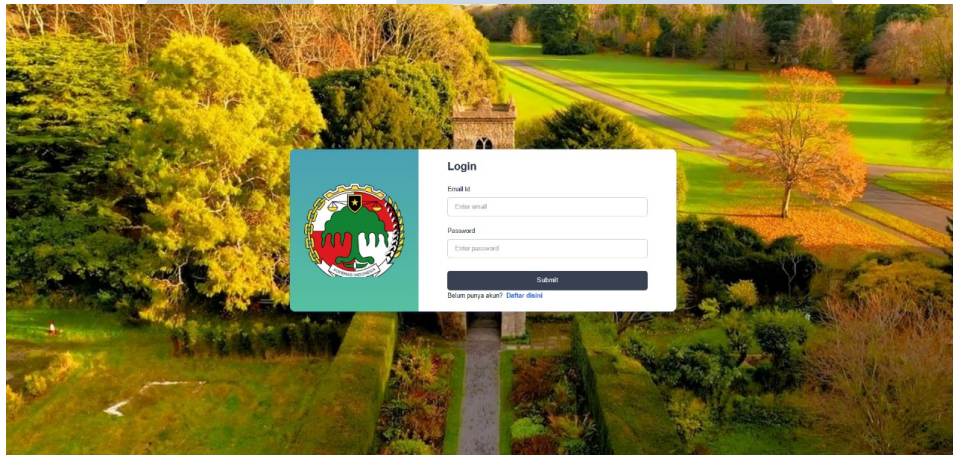
Berikut ini penjelasan singkat dari Halaman *Footer* :

- **Background:** Footer menggunakan latar belakang gradasi biru (bg-gradient-to-b) dari warna #2e86c1 ke #21618c.
- **Kontainer Utama:** Menggunakan `max-w-screen-xl` agar lebar konten tidak melebihi batas dan tetap responsif.
- **Bagian Kiri:** Menampilkan logo (`<img>`) dan nama *Koperasi Wiyata Mandala* di dalam `<a>` yang diarahkan ke halaman utama.
- **Navigasi Tengah:** Menggunakan `<ul>` dengan tiga tautan menu: Tentang Kami, Produk, dan Hubungi Kami. Tautan diberi efek hover underline.
- **Copyright:** Di bagian bawah terdapat teks hak cipta © 2024 LPPM Universitas Multimedia Nusantara.

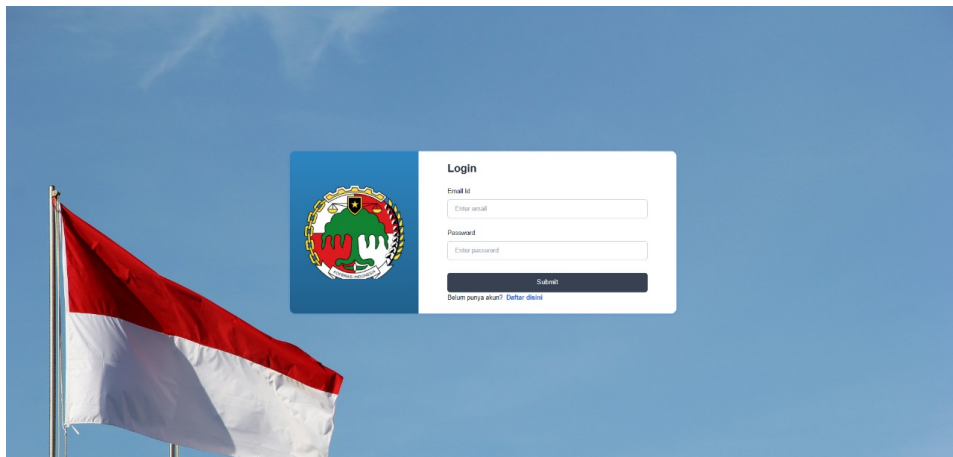
Struktur kode ini memastikan tampilan footer tetap rapi di berbagai ukuran layar, sekaligus memberikan akses cepat ke informasi penting bagi pengguna.

3.4.3 Halaman *Login*

Halaman *login* berfungsi sebagai pintu masuk bagi pengguna (anggota atau pengurus) ke dalam sistem informasi koperasi. Tampilan halaman *login* dirancang responsif dan modern dengan latar gradasi biru, serta menampilkan form *login* secara terpusat di layar.



Gambar 3.17. Tampilan Halaman *Login* Sebelum *revamp* pada Website Koperasi Wiyata Mandala



Gambar 3.18. Tampilan Halaman *Login* Sesudah *revamp* pada Website Koperasi Wiyata Mandala

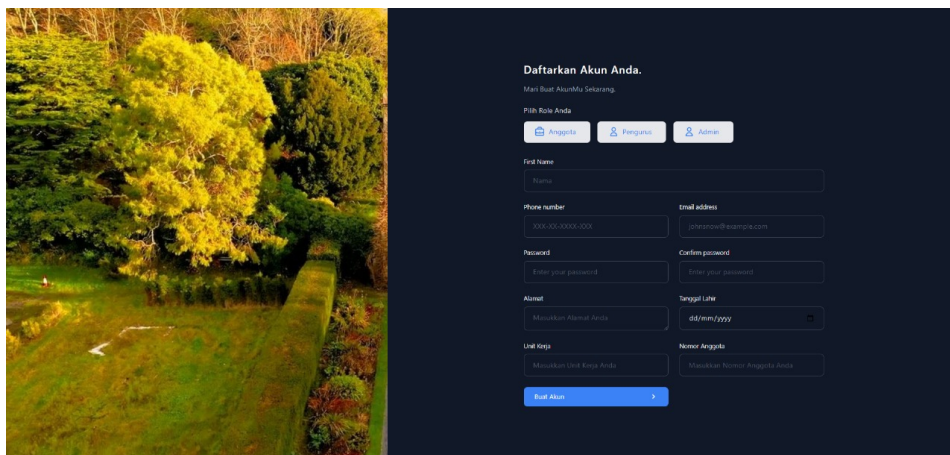
Berikut adalah bagian-bagian penting dalam kode:

- **Latar Belakang:** Menggunakan properti `style` untuk menampilkan gambar background dari variabel `$hijau`.

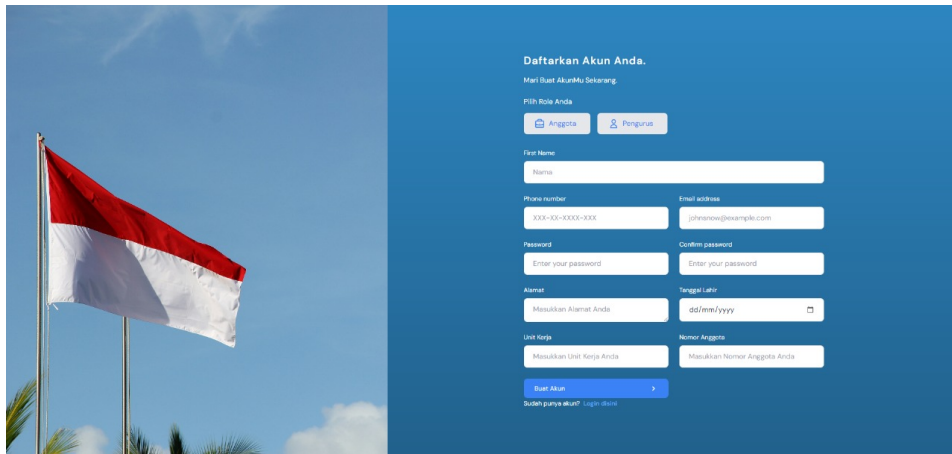
- **Struktur Grid:** Form *login* berada di sebelah kanan, sedangkan sebelah kiri digunakan untuk logo atau gambar dekoratif.
- **Input Form:** Terdapat dua input field untuk email dan password yang memiliki placeholder, validasi, dan state management melalui `setEmail` dan `setPassword`.
- **Tombol Submit:** Tombol Submit dikaitkan dengan event `onSubmit` yang memproses *login* saat diklik.
- **Link Registrasi:** Di bawah form terdapat tautan untuk berpindah ke halaman registrasi jika pengguna belum memiliki akun.

3.4.4 Halaman Registrasi

Halaman registrasi digunakan untuk mendaftarkan akun baru ke dalam sistem Koperasi Wiyata Mandala. Desain halaman dibuat dengan layout dua kolom — sebelah kiri berisi gambar visual, sementara sebelah kanan menampilkan form registrasi. Tampilan menggunakan latar gradasi biru dengan elemen *UI* yang modern dan responsif.



Gambar 3.19. Tampilan Halaman Registrasi Sebelum *revamp* pada Website Koperasi Wiyata Mandala



Gambar 3.20. Tampilan Halaman Registrasi Sesudah *revamp* pada Website Koperasi Wiyata Mandala

Berikut elemen-elemennya secara umum:

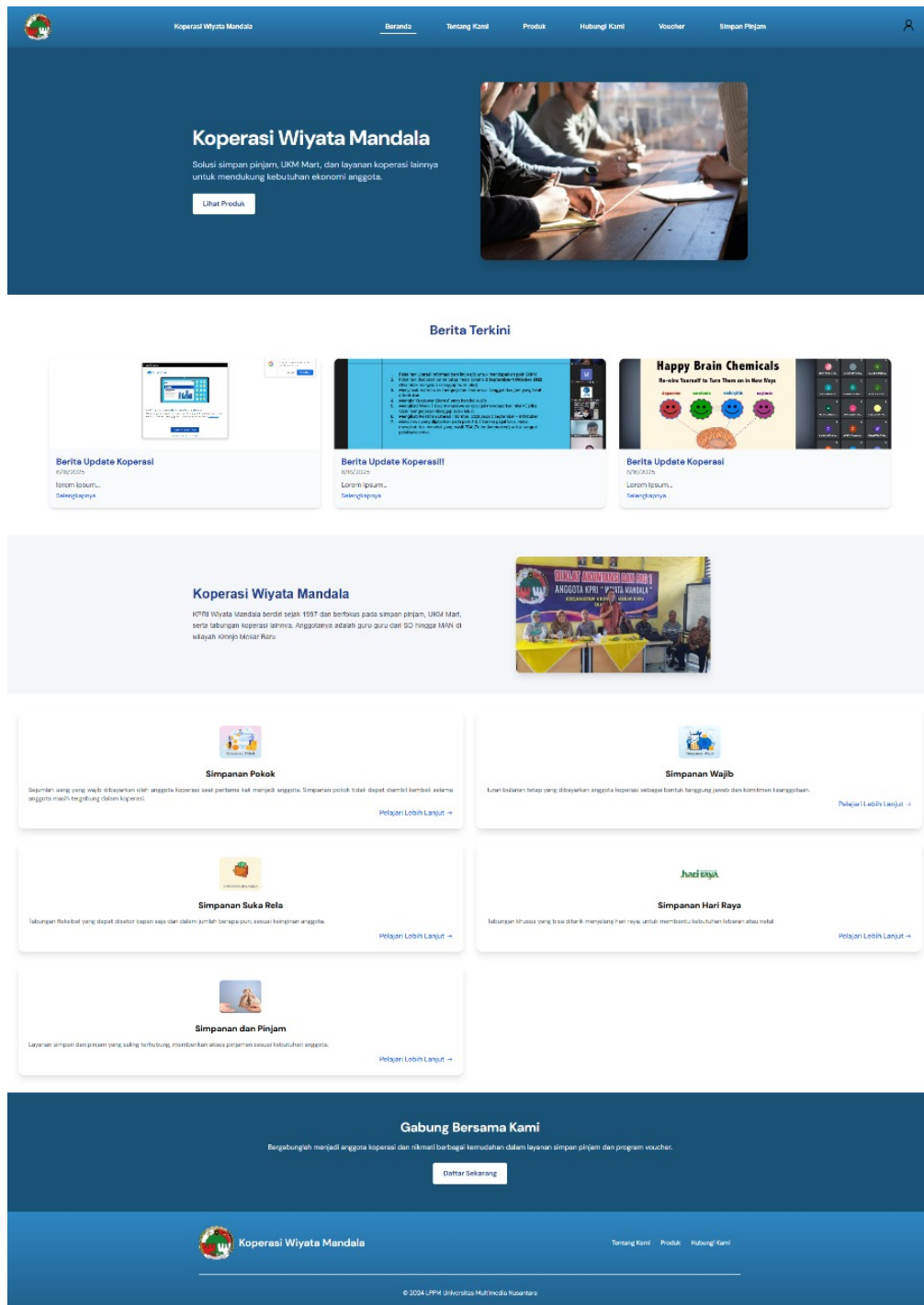
- **Latar Belakang:** Latar menggunakan efek gradasi dan gambar background yang diatur dengan *inline style*.
- **Struktur Layout :** Form registrasi berada di sebelah kanan dan dibuat responsif dengan sistem *grid*.
- **Input Formulir:** Terdapat input untuk:
 - Nama Lengkap
 - Nomor Telepon
 - Alamat Email
 - Password dan Konfirmasi Password
 - Alamat Tempat Tinggal
 - Tanggal Lahir (dengan *date picker*)
 - Unit Kerja
 - Nomor Anggota
- **Validasi dan Interaksi:** Setiap input menggunakan *state React* dan *event onChange* untuk menyimpan data pengguna. Semua field bersifat wajib diisi (*required*).
- **Tombol Submit:** Tombol digunakan untuk mengirim data registrasi. Di bawahnya terdapat link ke halaman *login* jika pengguna sudah memiliki akun.

3.4.5 Halaman Beranda

Halaman Beranda merupakan tampilan utama yang memberikan gambaran umum mengenai koperasi Wiyata Mandala. Halaman ini menampilkan informasi seputar layanan koperasi, berita terbaru, produk simpanan dan pinjaman, serta ajakan untuk bergabung. Desain dibuat responsif dan menarik dengan memanfaatkan *React* dan *Tailwind CSS*.



Gambar 3.21. Tampilan halaman Beranda sebelum *revamp* pada Website Koperasi Wiyata Mandala



Gambar 3.22. Tampilan halaman Beranda sesudah *revamp* pada Website Koperasi Wiyata Mandala

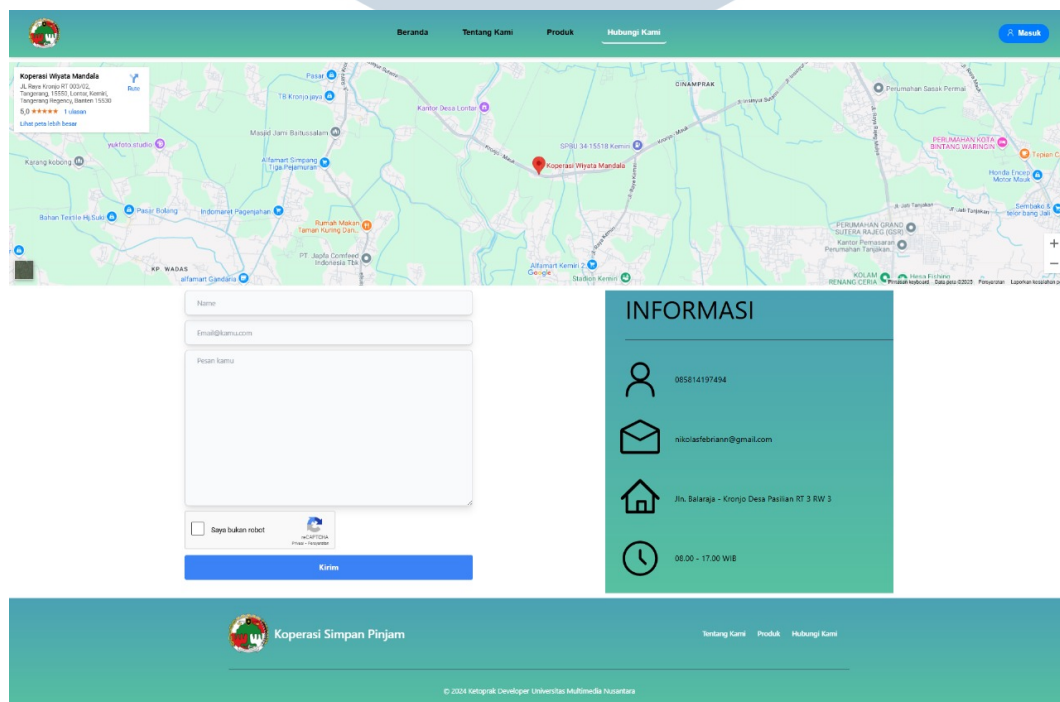
- **Carousel:** Menampilkan banner promosi koperasi dengan gambar latar dan slogan.
- **Berita Terkini:** Menampilkan daftar berita yang diambil dari endpoint *API*

menggunakan axios.

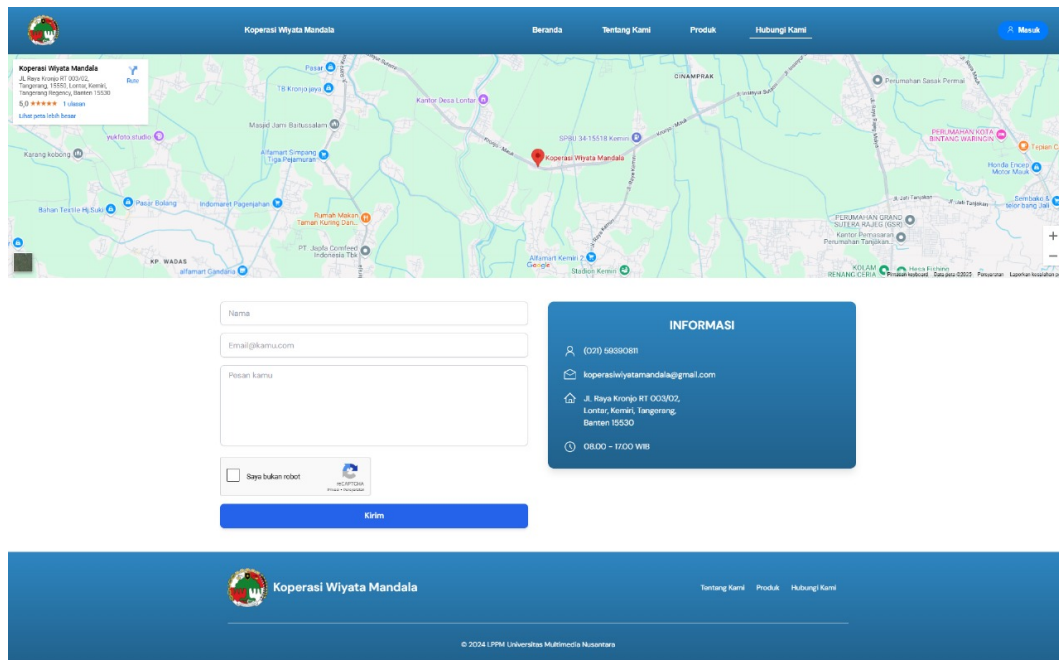
- **Informasi Koperasi:** Menjelaskan profil singkat koperasi termasuk sejarah singkat dan misi.
- **Daftar Produk:** Berisi informasi singkat mengenai jenis simpanan dan pinjaman yang tersedia.
- **Ajakan Bergabung:** Disediakan *call-to-action* (CTA) untuk mendorong pengunjung mendaftar menjadi anggota koperasi.

3.4.6 Halaman Hubungi Kami

Halaman Hubungi Kami digunakan untuk memfasilitasi komunikasi langsung antara pengguna dengan pengelola Koperasi Wiyata Mandala. Pengunjung dapat mengirimkan pesan, pertanyaan, atau saran melalui formulir yang tersedia. Selain itu, halaman ini juga menampilkan peta lokasi koperasi dan informasi kontak secara lengkap.



Gambar 3.23. Tampilan halaman Hubungi Kami sebelum *revamp* pada Website Koperasi Wiyata Mandala



Gambar 3.24. Tampilan halaman Hubungi Kami sesudah *revamp* pada Website Koperasi Wiyata Mandala

- **Peta Lokasi:** Ditampilkan menggunakan elemen `iframe` yang menampilkan Google Maps dengan titik lokasi koperasi.
- **Form Kontak:** Formulir berisi input untuk nama, email, dan pesan. Data dikirim menggunakan `axios` ke *endpoint* `/service-email`.
- **Validasi dan Status:** `State success` digunakan untuk menampilkan *alert* ketika pesan berhasil dikirim.
- **Informasi Kontak:** Ditampilkan di sisi kanan, mencakup nomor telepon, email, alamat lengkap, dan jam operasional.
- **Desain Responsif:** Menggunakan *Tailwind CSS* untuk memastikan tampilan tetap rapi dan nyaman diakses dari berbagai perangkat.

3.4.7 Halaman Tentang Kami

Halaman Tentang Kami bertujuan untuk memperkenalkan sejarah, visi, misi, serta tujuan dari Koperasi Wiyata Mandala. Halaman ini menjadi sarana bagi pengunjung untuk memahami latar belakang dan nilai-nilai yang diusung oleh

koperasi. Kontennya dirancang secara informatif dan ditampilkan dalam struktur yang rapi dengan desain responsif.



Gambar 3.25. Tampilan halaman Tentang Kami sebelum *revamp* pada Website Koperasi Wiyata Mandala

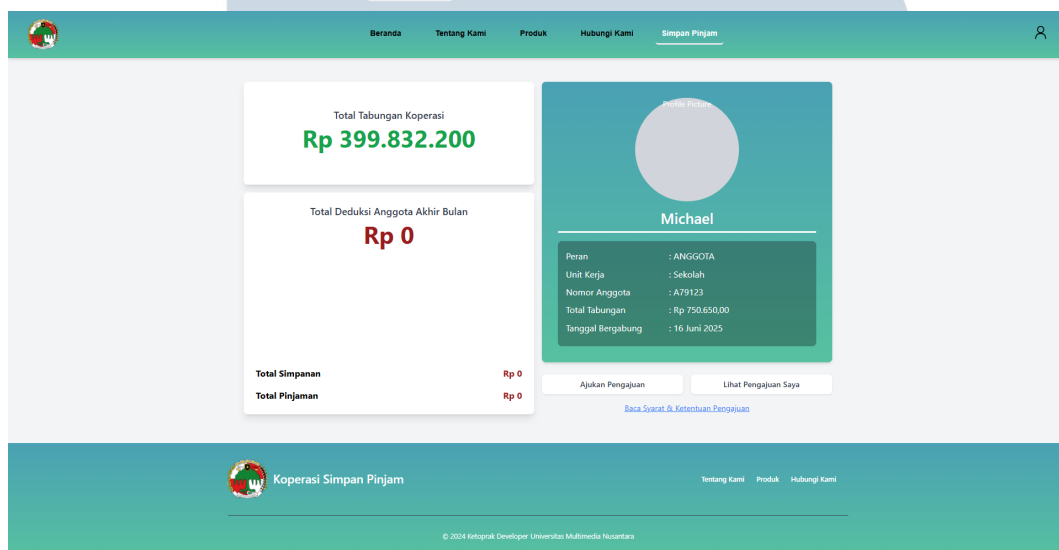


Gambar 3.26. Tampilan halaman Tentang Kami sesudah *revamp* pada Website Koperasi Wiyata Mandala

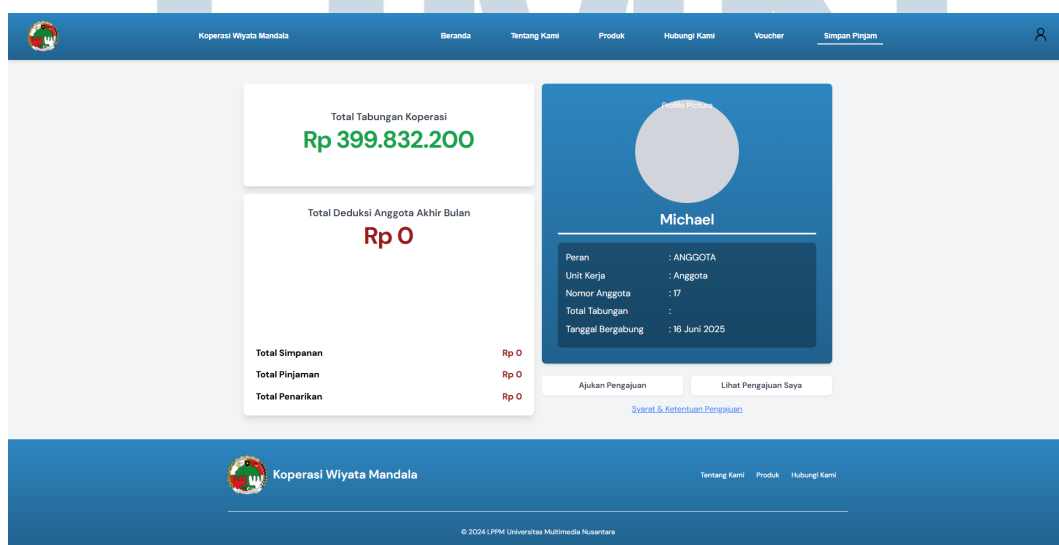
- **Deskripsi Koperasi:** Menjelaskan latar belakang, sejarah pendirian, dan jenis layanan yang disediakan oleh koperasi.
- **Tujuan Koperasi:** Ditampilkan dalam layout dua kolom dengan poin-poin yang merinci visi operasional koperasi.
- **Visi dan Misi:** Disusun secara berdampingan untuk menegaskan arah jangka panjang dan komitmen koperasi terhadap anggotanya.
- **Struktur Komponen:** Menggunakan *section*, *div*, dan *li* berbasis *React* dan *Tailwind CSS* untuk menjaga konsistensi visual dan responsivitas.
- **Gambar Pendukung:** Disematkan gambar dokumentasi yang memperkuat konteks penjelasan organisasi.

3.4.8 Halaman Simpan Pinjam Anggota dan Pengurus Koperasi Wiyata Mandala

Halaman ini berfungsi sebagai dasbor keuangan bagi anggota dan pengurus koperasi. Pengguna dapat melihat status tabungan, pinjaman, serta riwayat simpan pinjam anggota. Desainnya responsif dan menyesuaikan peran pengguna untuk menampilkan fitur yang relevan.



Gambar 3.27. Tampilan Halaman Simpan Pinjam Anggota sebelum *revamp* pada Website Koperasi Wiyata Mandala

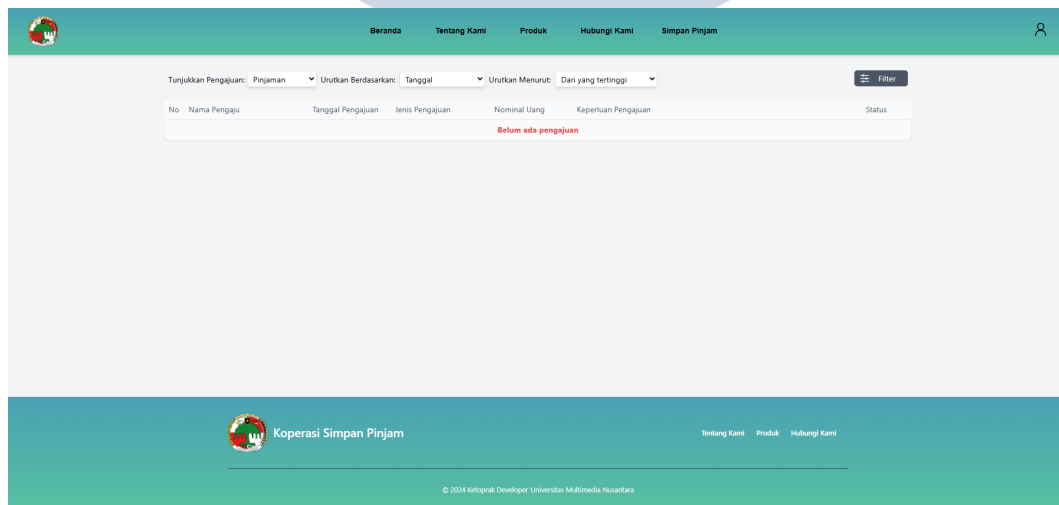


Gambar 3.28. Tampilan Halaman Simpan Pinjam Anggota sesudah *revamp* pada Website Koperasi Wiyata Mandala

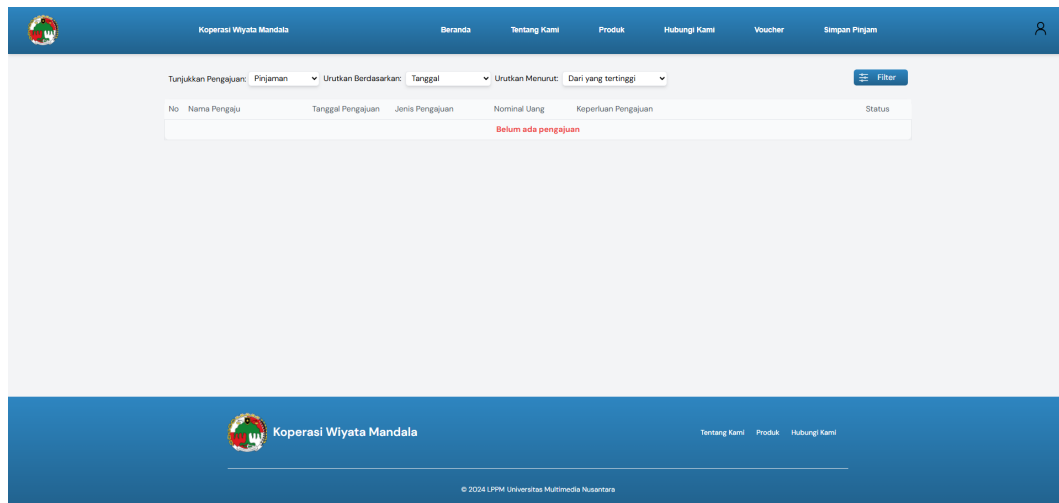
- **Ringkasan Tabungan:** Menampilkan total tabungan dan total potongan akhir bulan.
- **Profil Pengguna:** Memuat informasi nama, peran, unit kerja, nomor anggota, dan tanggal bergabung.
- **Aksi Pengguna:** Tombol untuk mengajukan simpanan/pinjaman, melihat riwayat pengajuan, dan membaca syarat ketentuan.
- **Peran Dinamis:** Tampilan fitur disesuaikan berdasarkan peran ANGGOTA atau PENGURUS.

3.4.9 Halaman Daftar Pengajuan Anggota Koperasi Wiyata Mandala Bagian Pengurus

Halaman ini menampilkan seluruh pengajuan yang dilakukan oleh anggota, baik simpanan, penarikan, maupun pinjaman. Fitur ini membantu pengurus dalam memantau dan memproses pengajuan anggota dengan efisien.



Gambar 3.29. Halaman Daftar Pengajuan Anggota sebelum *revamp* pada Website Koperasi Wiyata Mandala

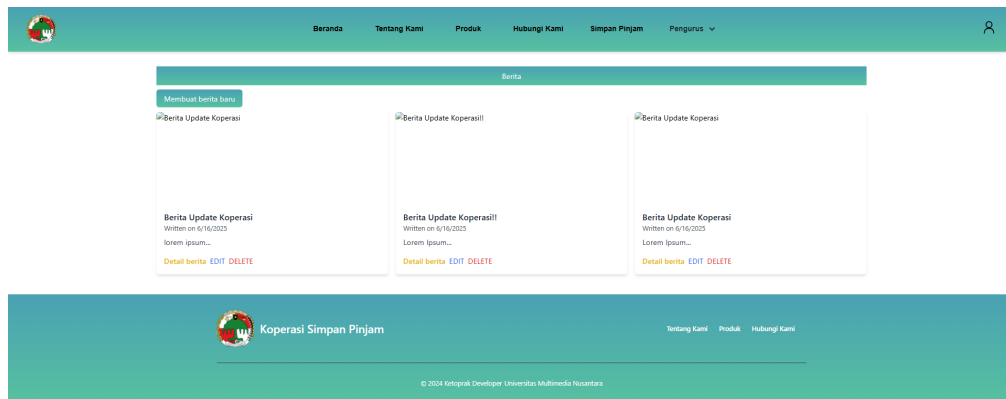


Gambar 3.30. Halaman Daftar Pengajuan Anggota sesudah *revamp* pada Website Koperasi Wiyata Mandala

- **SearchFilterBar:** Komponen pencarian dan filter yang mengatur filter, kriteria, dan aksi pencarian.
- **DataTable:** Komponen utama yang menampilkan tabel data pengajuan berdasarkan filter yang diterapkan.
- **Fitur Interaktif:** Mendukung *sorting and filtering* berdasarkan jenis pengajuan, nominal, dan tanggal.
- **Status Dinamis:** Setiap entri ditampilkan beserta statusnya seperti *Disetujui*, *Diproses*, atau *Ditolak*.

3.4.10 Halaman Menu Berita oleh Pengurus Koperasi Wiyata Mandala

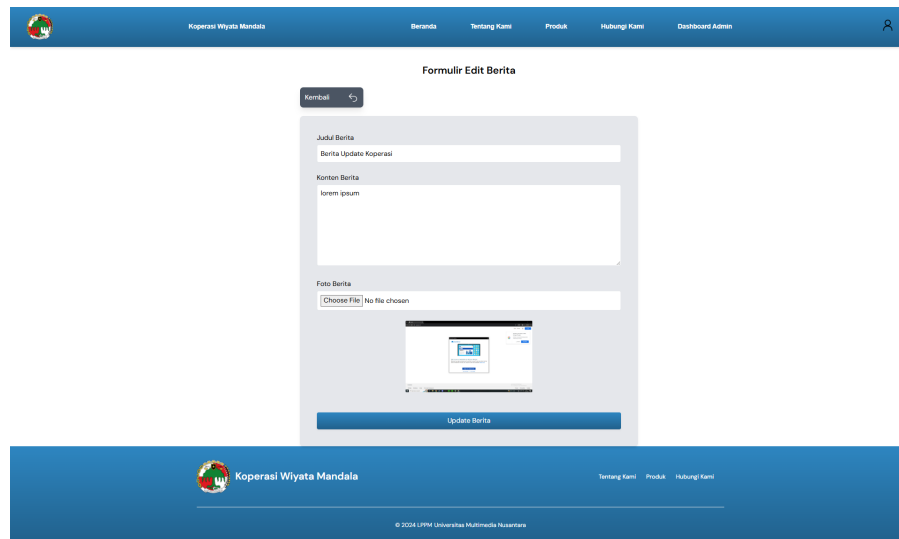
Halaman ini merupakan pusat pengelolaan konten informasi dan pengumuman penting yang dilakukan oleh pengurus koperasi. Fitur ini mempermudah penyampaian berita kepada seluruh anggota koperasi.



Gambar 3.31. Halaman Menu Berita oleh Pengurus sebelum *revamp* Koperasi Wiyata Mandala



Gambar 3.32. Halaman Menu Berita oleh Pengurus sesudah *revamp* Koperasi Wiyata Mandala



Gambar 3.33. Halaman Tambah Berita oleh Pengurus Koperasi Wiyata Mandala

- **Grid Berita:** Menampilkan daftar berita dalam format *grid* 3 kolom.
- **Kartu Berita:** Menyediakan gambar, judul, ringkasan isi, dan tanggal penulisan.
- **Aksi CRUD:** Setiap kartu berita memiliki tombol *Detail*, *Edit*, dan *Delete*.
- **Modal Konfirmasi:** Digunakan saat pengurus ingin menghapus berita, memastikan aksi dilakukan dengan sengaja.

Penjelasan: Formulir Tambah Berita

- **Input Form:** Tersedia input judul, nama penulis, isi berita, dan unggahan gambar.
- **Preview Gambar:** Menampilkan *preview* sebelum berita dikirim.
- **Konfirmasi Submit:** Menyediakan *modal* konfirmasi saat pengguna menekan tombol kirim.
- **Validasi Loading:** Menampilkan teks “Sedang Mengirim...” untuk menunjukkan *status loading*.

Antarmuka dibuat bersih dan responsif. Tombol aksi diberi warna berbeda untuk meningkatkan keterbacaan: biru untuk *edit*, merah untuk *delete*, kuning untuk *detail*. *Layout grid* menjaga kerapian tampilan berita.

Melalui fitur ini, pengurus dapat mengelola konten dengan efisien dan menyampaikan informasi penting kepada seluruh anggota koperasi dengan cepat dan terstruktur.

3.5 Perancangan dan pengenalan Fitur Voucher

Fitur voucher merupakan salah satu komponen penting dalam sistem koperasi Wiyata Mandala. Fitur ini memungkinkan anggota koperasi untuk mengajukan pinjaman dalam bentuk voucher yang dapat digunakan sebagai alat transaksi di mitra usaha seperti Alfamart. Pengajuan voucher dilakukan berdasarkan nominal dan tenor (jangka waktu cicilan), yang kemudian akan diverifikasi dan disetujui oleh pengurus koperasi. Proses ini mencakup perancangan antarmuka pengguna (UI), logika *backend*, serta penyimpanan data dalam basis data.

3.6 Implementasi Fitur Voucher

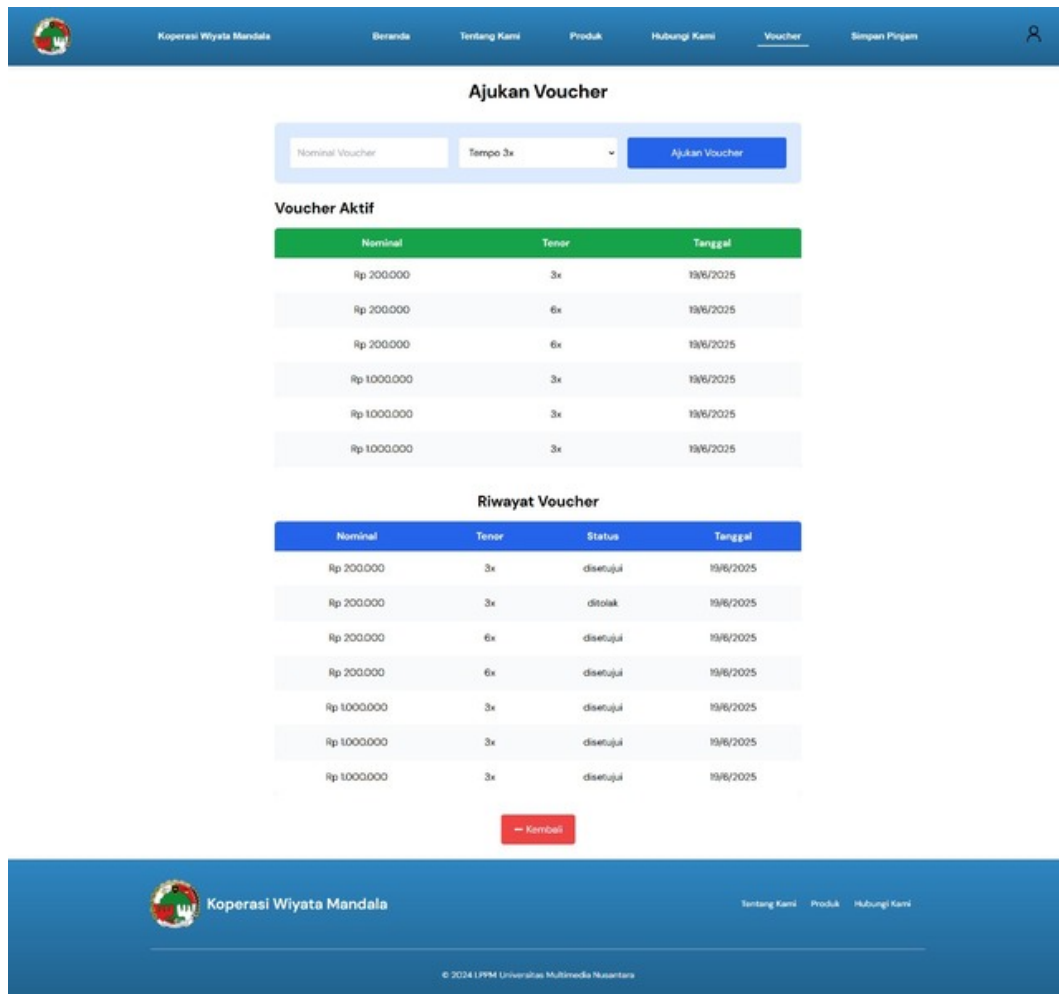
Fitur voucher merupakan salah satu fitur baru yang dikembangkan untuk mendukung layanan koperasi digital, memungkinkan anggota mengajukan voucher seperti pinjaman dengan tenor tertentu. Sistem ini terdiri dari antarmuka anggota dan pengurus yang terintegrasi dengan *backend* berbasis **Express.js** dan **PostgreSQL**.

3.6.1 Perancangan Antarmuka Pengguna

Antarmuka pengguna dibuat menggunakan **Vite + React.js**.

Anggota dapat:

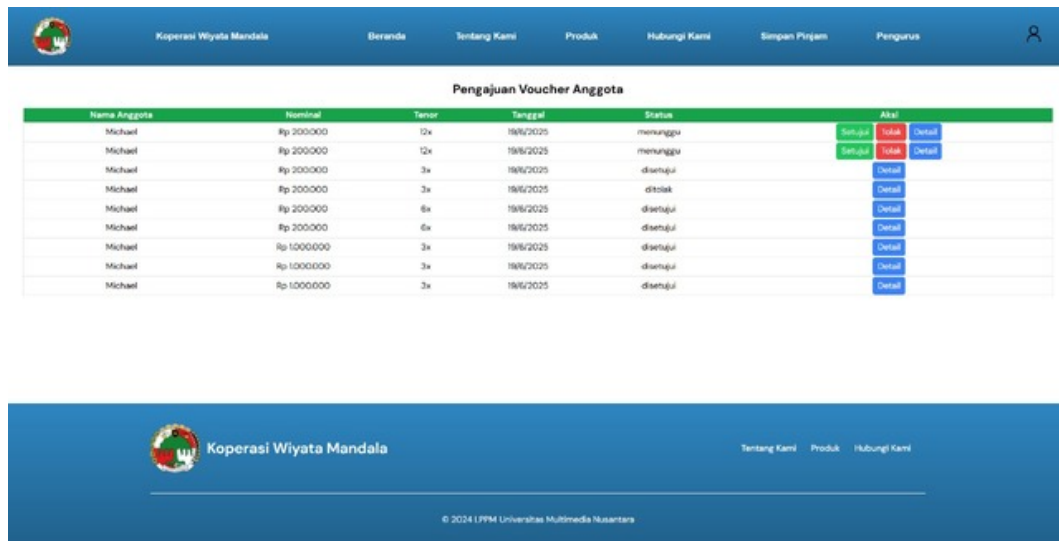
- Mengajukan voucher
- Melihat voucher aktif
- Melihat riwayat pengajuan voucher



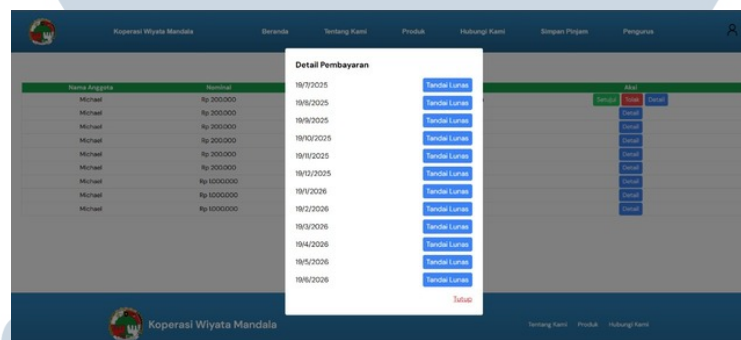
Gambar 3.34. Tampilan halaman voucher anggota (*frontend React*)

Pengurus dapat:

- Melihat daftar pengajuan voucher dari anggota
- Menyetujui/menolak pengajuan
- Melihat detail pembayaran berdasarkan tenor
- Menandai pelunasan voucher



Gambar 3.35. Tampilan halaman pengurus voucher (*Frontend React*)



Gambar 3.36. Detail pembayaran pada voucher pengurus (modal)

3.6.2 Implementasi *backend*

Backend dibangun menggunakan **Node.js** dan **Express.js**. Rute-rute penting antara lain:

- POST /pengurus-voucher/ajukan – anggota mengajukan voucher
- GET /pengurus-voucher/pengajuan/:user_id – ambil data voucher anggota
- GET /pengurus-voucher/pengajuan – untuk pengurus melihat semua pengajuan
- PUT /pengurus-voucher/pengajuan/:id – ubah status disetujui/ditolak

- GET /pengurus-voucher/detail/:id – ambil detail pembayaran per tenor
- PUT /pengurus-voucher/detail/:id/:index – tandai pelunasan pembayaran



Gambar 3.37. Implementasi *routing* untuk fitur voucher (*Express.js*)

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

1 import TR_VOUCHER_USER from "../models/TR_VOUCHER_USER.js";
2 import MS_USER from "../models/MS_USER.js";
3
4 // Ubah status disetujui/ditolak
5 export const updateStatusVoucher = async (req, res) => {
6   const { id } = req.params;
7   const { status } = req.body;
8   try {
9     const voucher = await TR_VOUCHER_USER.findByPk(id);
10    if (!voucher) return res.status(404).json({ message: "Voucher tidak ditemukan" });
11
12    voucher.STATUS = status;
13    await voucher.save();
14
15    res.json({ message: "Status berhasil diperbarui" });
16  } catch (err) {
17    console.error("Gagal update status:", err.message);
18    res.status(500).json({ message: "Gagal update status" });
19  }
20 };
21
22 // Ambil detail pembayaran per tenor
23 export const getDetailPembayaran = async (req, res) => {
24   const { id } = req.params;
25   try {
26     const voucher = await TR_VOUCHER_USER.findByPk(id);
27     if (!voucher) return res.status(404).json({ message: "Voucher tidak ditemukan" });
28
29     if (voucher.PEMBAYARAN_JSON?.length > 0) {
30       return res.json(voucher.PEMBAYARAN_JSON);
31     }
32
33     // Generate jika belum ada
34     const tanggalAwal = new Date(voucher.CREATED_AT);
35     const tenor = voucher.TENOR;
36     const pembayaran = [];
37
38     for (let i = 0; i < tenor; i++) {
39       const jatuhTempo = new Date(tanggalAwal);
40       jatuhTempo.setMonth(jatuhTempo.getMonth() + i + 1);
41       pembayaran.push({
42         tanggal: jatuhTempo,
43         status: "belum"
44       });
45     }
46
47     voucher.PEMBAYARAN_JSON = pembayaran;
48     await voucher.save();
49
50     res.json(pembayaran);
51   } catch (err) {
52     console.error("Gagal ambil detail:", err.message);
53     res.status(500).json({ message: "Gagal ambil detail pembayaran" });
54   }
55 };
56
57 // Update pelunasan
58 export const tandailunas = async (req, res) => {
59   const { id, index } = req.params;
60   try {
61     const voucher = await TR_VOUCHER_USER.findByPk(id);
62     if (!voucher) return res.status(404).json({ message: "Voucher tidak ditemukan" });
63
64     if (!Array.isArray(voucher.PEMBAYARAN_JSON) || !voucher.PEMBAYARAN_JSON[index]) {
65       return res.status(400).json({ message: "Data pembayaran tidak valid" });
66     }
67
68     voucher.PEMBAYARAN_JSON[index].status = "lunas";
69     await voucher.save();
70
71     res.json({ message: `Pembayaran ke-${index + 1} ditandai lunas` });
72   } catch (err) {
73     console.error("Gagal update lunas:", err.message);
74     res.status(500).json({ message: "Gagal update status pembayaran" });
75   }
76 };
77

```

Gambar 3.38. Potongan kode *controller backend* voucher

```

1 // Pengajuan oleh anggota
2 export const ajukanVoucher = async (req, res) => {
3   const { user_id, nominal, tenor } = req.body;
4   try {
5     await TR_VOUCHER_USER.create({
6       UUID_MS_USER: user_id,
7       NOMINAL: nominal,
8       TENOR: tenor,
9       STATUS: "menunggu",
10      CREATED_AT: new Date(),
11    });
12    res.json({ message: "Pengajuan voucher berhasil" });
13  } catch (err) {
14    res.status(500).json({ message: "Gagal mengajukan voucher" });
15  }
16 };
17
18 // Semua pengajuan untuk admin
19
20 export const getAllPengajuan = async (req, res) => {
21   try {
22     const data = await TR_VOUCHER_USER.findAll({
23       include: [{
24         model: MS_USER,
25         attributes: ["NAMA LENGKAP"],
26       }],
27       order: [["CREATED_AT", "DESC"]],
28     });
29
30     const formatted = data.map((d) => ({
31       id: d.ID_VOUCHER_USER,
32       user_id: d.UUID_MS_USER,
33       nama_anggota: d.MS_USER?.NAMA LENGKAP || "Tanpa Nama",
34       nominal: d.NOMINAL,
35       tenor: d.TENOR,
36       status: d.STATUS,
37       created_at: d.CREATED_AT,
38     }));
39
40     res.json(formatted);
41   } catch (err) {
42     console.error("Gagal ambil semua pengajuan:", err.message);
43     res.status(500).json({ message: "Gagal mengambil data pengajuan" });
44   }
45 };
46
47
48 // Pengajuan per user
49 export const getPengajuanByUser = async (req, res) => {
50   const { user_id } = req.params;
51   try {
52     const data = await TR_VOUCHER_USER.findAll({
53       where: { UUID_MS_USER: user_id },
54       order: [["CREATED_AT", "DESC"]],
55     });
56
57     const formatted = data.map((d) => ({
58       id: d.ID_VOUCHER_USER,
59       nominal: d.NOMINAL,
60       tenor: d.TENOR,
61       status: d.STATUS,
62       created_at: d.CREATED_AT,
63     }));
64
65     res.json(formatted);
66   } catch (err) {
67     res.status(500).json({ message: "Gagal mengambil pengajuan user" });
68   }
69 };
70
71 // Setujui/Tolak
72 export const updateStatusPengajuan = async (req, res) => {
73   const { id } = req.params;
74   const { status } = req.body;
75   try {
76     await TR_VOUCHER_USER.update({ STATUS: status }, { where: { ID_VOUCHER_USER: id } });
77     res.json({ message: "Status diperbarui" });
78   } catch (err) {
79     res.status(500).json({ message: "Gagal memperbarui status" });
80   }
81 };

```

Gambar 3.39. Potongan kode *controller backend* voucher

3.6.3 Struktur Basis Data

Basis data menggunakan *PostgreSQL* dengan model utama TR_VOUCHER_USER. Struktur tabel meliputi:

- ID_VOUCHER_USER – primary key
- UUID_MS_USER – foreign key ke MS_USER
- NOMINAL, TENOR, STATUS, CREATED_AT
- PEMBAYARAN_JSON – array JSON berisi jadwal dan status cicilan



```

1 import { DataTypes } from "sequelize";
2 import db from "../config/database.js";
3 import MS_USER from "../MS_USER.js";
4
5 const TR_VOUCHER_USER = db.define("TR_VOUCHER_USER", {
6   ID_VOUCHER_USER: {
7     type: DataTypes.INTEGER,
8     primaryKey: true,
9     autoIncrement: true,
10  },
11  UUID_MS_USER: {
12    type: DataTypes.BIGINT,
13    allowNull: false,
14  },
15  NOMINAL: {
16    type: DataTypes.INTEGER,
17    allowNull: false,
18  },
19  TENOR: {
20    type: DataTypes.INTEGER,
21    allowNull: false,
22  },
23  STATUS: {
24    type: DataTypes.STRING,
25    allowNull: false,
26    defaultValue: "menunggu",
27  },
28  CREATED_AT: {
29    type: DataTypes.DATE,
30    allowNull: false,
31    defaultValue: DataTypes.NOW,
32  },
33  PEMBAYARAN_JSON: {
34    type: DataTypes.JSONB,
35    allowNull: true,
36    defaultValue: [],
37  },
38 }, {
39   freezeTableName: true,
40   timestamps: false,
41 });
42
43 TR_VOUCHER_USER.belongsTo(MS_USER, {
44   foreignKey: "UUID_MS_USER",
45 });
46
47 MS_USER.hasMany(TR_VOUCHER_USER, {
48   foreignKey: "UUID_MS_USER",
49 });
50
51 export default TR_VOUCHER_USER;
52

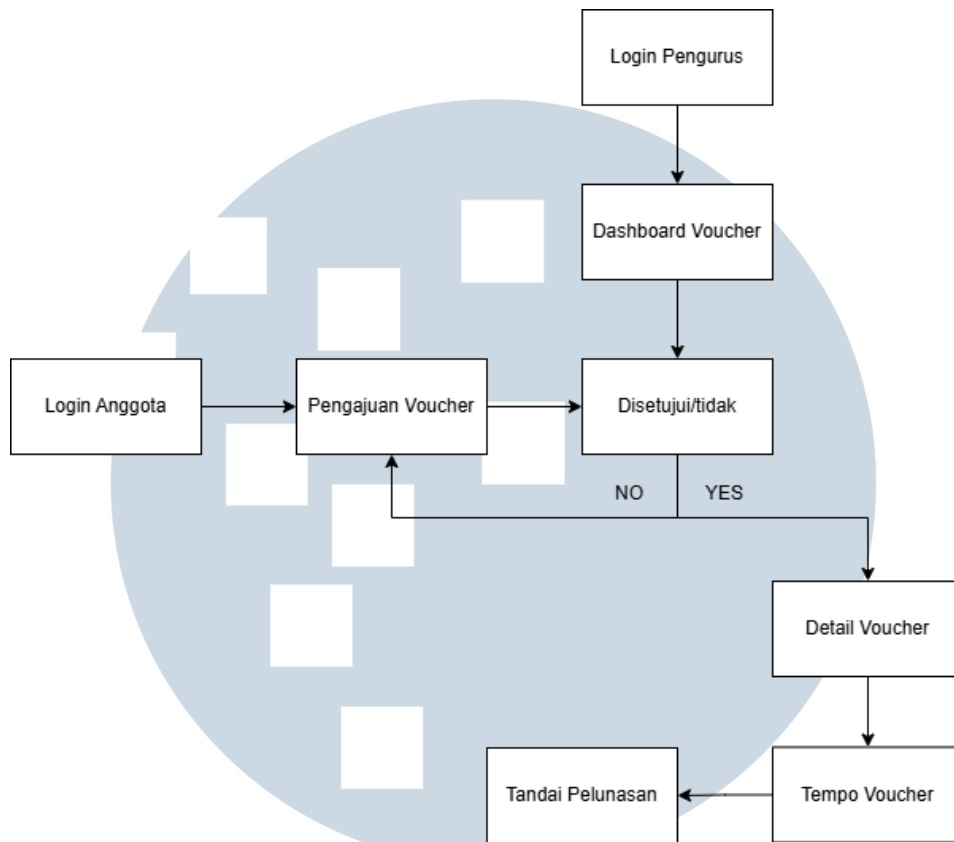
```

Gambar 3.40. Model Sequelize untuk tabel TR_VOUCHER_USER

3.6.4 Flowchart Fitur Voucher

Flowchart pada Gambar 3.41 menjelaskan alur sistem pengelolaan voucher yang dibagi menjadi dua peran utama, yaitu **Anggota** dan **Pengurus Koperasi**. Proses dimulai dari masing-masing pengguna melakukan *login* ke sistem, kemudian dilanjutkan sesuai dengan hak aksesnya.

- **Login Anggota:** Halaman *Login* untuk Anggota di redirect ke halaman Voucher Anggota .
- **Pengajuan Voucher:** Setelah *Login*, anggota dapat mengajukan permohonan voucher berdasarkan perhitungan pengurus koperasi terkait saldo tabungan anggota.
- **Login Pengurus:** Pengurus koperasi juga dapat masuk ke dalam sistem untuk mengakses *dashboard* pengelolaan voucher.
- **Dashboard Voucher:** Halaman ini menampilkan daftar pengajuan voucher dari anggota yang ditinjau oleh pengurus.
- **Disetujui/Tidak:** Pengurus dapat menyetujui atau menolak permohonan voucher. Jika disetujui, maka voucher akan di berikan.
- **Detail Voucher:** Pengurus maupun anggota dapat melihat detail dari voucher yang telah diajukan.
- **Tempo Voucher:** Setiap voucher memiliki tempo waktu pembayaran yang sudah di tentukan yang harus di bayar sesuai tempo.
- **Tandai Pelunasan:** Setelah voucher sudah di bayar sesuai tempo oleh anggota, pengurus dapat menandai voucher sebagai telah dilunasi.



Gambar 3.41. Flowchart Proses Pengelolaan Voucher

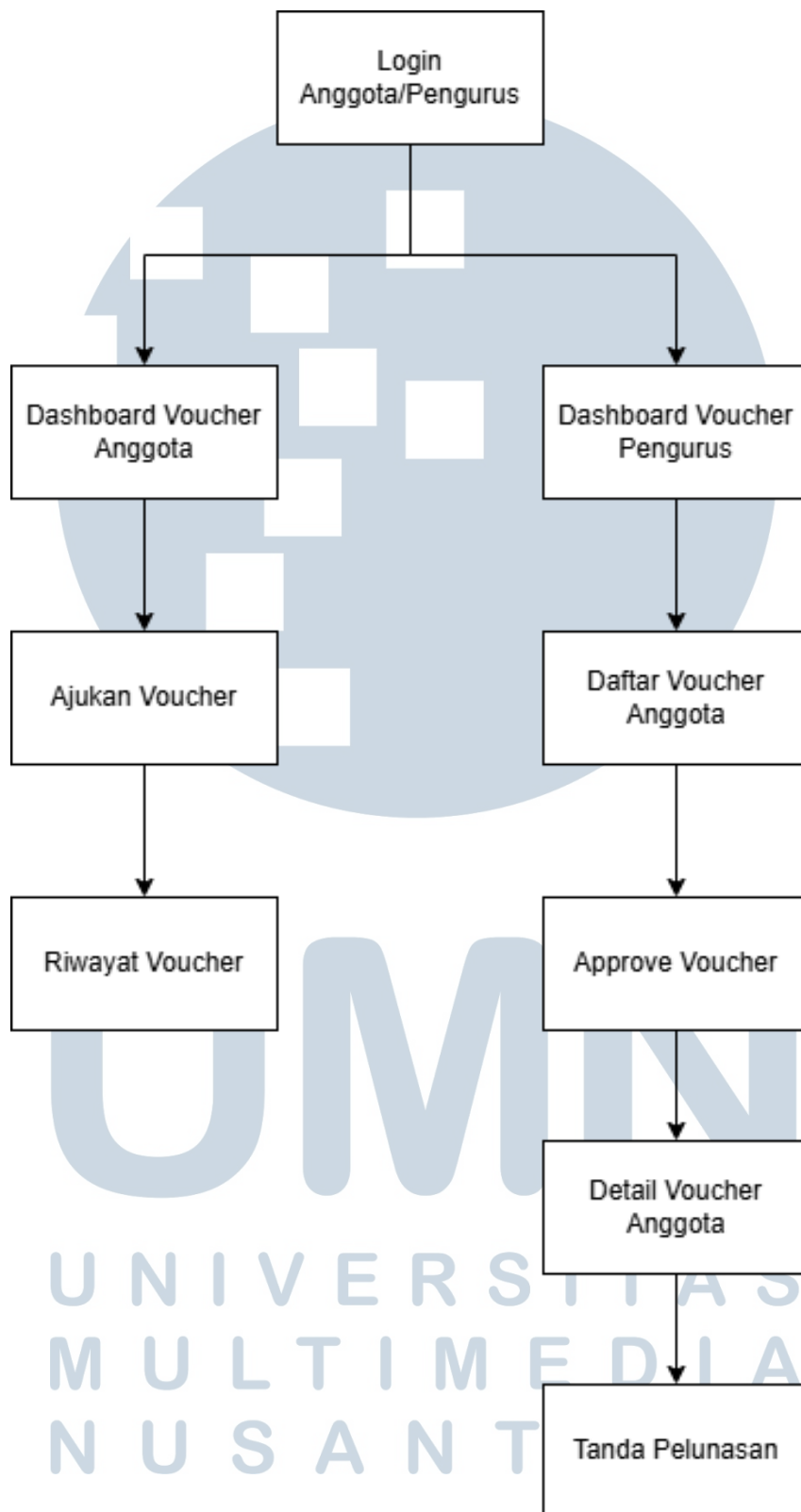
3.6.5 Sitemap Fitur Voucher

Sitemap pada Gambar 3.42 menunjukkan alur navigasi dan akses dari fitur voucher dalam sistem koperasi. Terdapat dua tipe pengguna utama yang menggunakan fitur ini, yaitu **anggota** dan **pengurus**.

- **Login Anggota:** Anggota mengakses halaman *login* dan setelah berhasil masuk, dapat mengajukan permohonan voucher melalui form pengajuan.
- **Pengajuan Voucher:** Permohonan voucher dikirim ke *backend* dan masuk ke *dashboard* pengurus dengan status *menunggu*.
- **Login Pengurus:** Pengurus masuk ke sistem melalui *login* khusus, lalu mengakses halaman *dashboard voucher* yang menampilkan semua permohonan.
- **Dashboard Voucher:** Pada halaman ini, pengurus dapat menyetujui atau menolak permohonan.

- **Detail Voucher:** Jika pengurus memilih untuk melihat detail, maka akan muncul informasi *tempo cicilan* berdasarkan tenor yang dipilih saat pengajuan.
- **Tempo Voucher dan Pelunasan:** Setiap tempo (misal 3x) memiliki jadwal cicilan bulanan. Pengurus dapat menandai setiap cicilan sebagai *lunas* setelah diverifikasi.





Gambar 3.42. Sitemap fitur voucher

3.7 Kendala dan Solusi Pembuatan Fitur Voucher

Pengembangan fitur voucher melibatkan integrasi antara *Frontend (React.js)* dan *backend (Express.js)* serta pengelolaan data dengan basis data *PostgreSQL*. Dalam proses pengembangan ini, terdapat beberapa kendala yang muncul dan telah diselesaikan dengan pendekatan solusi yang efektif. Rincian kendala dan solusi dijelaskan sebagai berikut:

3.7.1 Kendala pada *Frontend*

1. Permasalahan validasi input kosong

- *Kendala:* Saat pengguna mengajukan voucher tanpa mengisi nominal atau tenor, permintaan tetap dikirim ke server.
- *Solusi:* Ditambahkan validasi input di sisi *Frontend* menggunakan `if (!nominal || !tenor)` untuk memblokir tombol submit jika ada input kosong.

2. Penanganan error *Axios*

- *Kendala:* Pesan error dari *backend* tidak muncul secara jelas di *UI* saat gagal mengambil data.
- *Solusi:* Error handling *Axios* diperbaiki dengan menambahkan blok `try-catch` dan menampilkan notifikasi atau alert pada tampilan.

3. Kesalahan penggunaan `toLocaleString()`

- *Kendala:* Method `toLocaleString()` digunakan pada data yang belum terdefinisi menyebabkan error runtime.
- *Solusi:* Ditambahkan validasi `voucher.nominal || 0` sebelum memanggil `toLocaleString()` untuk menghindari akses nilai `undefined`.

4. Footer tidak berada di bawah halaman

- *Kendala:* Footer tidak berada di bagian paling bawah halaman ketika data terlalu sedikit.
- *Solusi:* Menambahkan styling *Tailwind CSS* `min-h-screen flex flex-col justify-between` untuk menjaga struktur *layout*.

3.7.2 Kendala pada *Backend*

1. Duplikasi *controller* dan kerancuan *routing*

- *Kendala:* Awalnya *backend* menggunakan dua file *controller* berbeda (*VoucherController.js* dan *VoucherPengajuanController.js*) yang membingungkan.
- *Solusi:* Semua logika digabungkan ke dalam satu file *VoucherController.js* untuk penyederhanaan dan efisiensi.

2. Model tidak ditemukan saat runtime

- *Kendala:* Error Cannot access 'TR.VOUCHER.USER' before initialization.
- *Solusi:* Struktur impor model diatur ulang agar tidak terjadi *circular dependency*. Pastikan tidak ada deklarasi *model* di bawah bagian yang menggunakannya.

3. *Endpoint* tidak dikenali (*Error 404*)

- *Kendala:* Request *Frontend* ke */pengurus-voucher/detail/:id* gagal karena route tidak terdaftar.
- *Solusi:* Menambahkan *endpoint* baru dan *handler* di *VoucherRoutes.js* serta fungsinya di *VoucherController.js*.

4. Penanganan struktur JSON cicilan tempo

- *Kendala:* Data *PEMBAYARAN_JSON* belum otomatis terisi jika belum pernah diakses sebelumnya.
- *Solusi:* Di *backend*, jika data cicilan kosong, maka sistem akan secara otomatis menghasilkan *array* jadwal cicilan berdasarkan tenor dan menyimpannya ke database.

3.7.3 Hasil dari Pemecahan Kendala

Dengan diterapkannya solusi-solusi di atas, fitur voucher kini dapat berjalan secara stabil. Proses pengajuan, validasi, pengelolaan status oleh pengurus, hingga pelunasan cicilan voucher dapat diakses secara lancar dan responsif baik oleh anggota maupun pengurus koperasi.