

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama pelaksanaan kerja magang di PT Bayu Cahaya Gemilang, mahasiswa berada di bawah supervisi langsung Bapak Muhammad Guntur Wicaksono selaku Direktur Utama perusahaan. Kegiatan magang dilaksanakan pada divisi pengembangan teknologi informasi, khususnya dalam proyek pembuatan *website company profile* dan *CMS Admin*. Koordinasi dilakukan secara rutin baik secara daring maupun luring sesuai sistem kerja *hybrid* yang berlaku, serta melalui sesi diskusi dan presentasi progres proyek kepada supervisor dan tim terkait setiap satu hingga dua minggu sekali.

3.2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang, mahasiswa mendapatkan arahan langsung dari Bapak Muhammad Guntur Wicaksono selaku Direktur Utama PT Bayu Cahaya Gemilang. Mahasiswa terlibat dalam pengembangan dua proyek utama, yaitu *website company profile* dan sistem *CMS Admin*. Proyek ini dikembangkan menggunakan *ReactJS* dan *TailwindCSS* untuk sisi *frontend*, serta *NodeJS (Express)* dan *MySQL* untuk sisi *backend*. Untuk pengelolaan media, proyek *CMS Admin* juga terintegrasi dengan layanan *Cloudinary*.

Adapun tugas yang mahasiswa kerjakan selama magang terbagi dalam beberapa tahap sebagai berikut:

1. Menganalisis kebutuhan fitur dan struktur konten untuk kedua proyek *website*.
2. Mendesain antarmuka pengguna (*UI*) menggunakan prinsip *user-friendly* dan *responsive*, serta membuat prototipe awal sesuai permintaan perusahaan.
3. Mengembangkan tampilan antarmuka menggunakan *ReactJS* dan *TailwindCSS* berdasarkan desain yang telah disetujui.
4. Membangun dan menghubungkan *API* menggunakan *NodeJS (Express)* untuk pengelolaan data dinamis, terutama pada *CMS Admin*.

5. Melakukan integrasi dengan layanan *Cloudinary* untuk pengunggahan dan pengelolaan gambar secara daring.
6. Melaksanakan pengujian fitur secara manual untuk memastikan fungsionalitas berjalan dengan baik dan sesuai harapan pengguna.
7. Melakukan revisi berdasarkan hasil evaluasi dari supervisor dan tim pengembang, serta mendokumentasikan setiap tahapan pekerjaan.

3.3 Uraian Pelaksanaan Magang

Dalam bagian ini akan dijelaskan secara rinci mengenai tahapan pelaksanaan kegiatan magang yang dilaksanakan setiap minggunya. Penjabaran mencakup deskripsi pekerjaan yang telah diselesaikan, hambatan yang dihadapi selama proses pengerjaan, serta solusi yang diterapkan untuk mengatasi permasalahan tersebut. Penjelasan ini bertujuan untuk memberikan pemahaman yang lebih jelas terkait progres kegiatan magang dan pengalaman yang diperoleh. Dengan demikian, diharapkan dapat menggambarkan pengembangan keterampilan dan kompetensi yang diperoleh selama periode magang berlangsung.

3.3.1 Proses Pelaksanaan

Pelaksanaan kerja magang berlangsung selama 16 minggu atau 4 bulan, dengan kegiatan yang ditampilkan pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu

Minggu Ke -	Pekerjaan yang dilakukan
1 - 2	<ul style="list-style-type: none"> • <i>Brainstorming</i> kebutuhan dan konsep desain <i>website company profile</i> dan <i>CMS Admin</i>. • Membuat sketsa awal desain <i>UI</i> di <i>Figma</i> dan <i>pitching</i> ide ke <i>stakeholder</i>. • Mendapatkan <i>feedback</i> awal dan melakukan revisi desain.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu Ke -	Pekerjaan yang di lakukan
3 - 4	<ul style="list-style-type: none"> • Finalisasi desain <i>UI/UX</i> untuk semua halaman utama. • Mulai belajar <i>React.js</i>, <i>Tailwind CSS</i>, dan <i>React Router</i> untuk persiapan pengembangan <i>frontend</i>. • Implementasi dasar komponen <i>layout</i> dan <i>routing</i>.
5 - 6	<ul style="list-style-type: none"> • Setup proyek <i>frontend</i> dengan <i>React.js</i> dan <i>Vite</i>, serta konfigurasi dependensi. • Membuat komponen dasar seperti <i>Navbar</i> dan halaman beranda. • Menerapkan desain ke dalam halaman Beranda, Tentang Kami, Developer, Simulasi KPR, Kontak Kami dan Cluster.
7 - 8	<ul style="list-style-type: none"> • Menyempurnakan <i>UI/UX</i> dengan <i>Tailwind CSS</i> dan <i>AOS</i> sebagai penambahan animasi. • Mempelajari dasar-dasar <i>backend</i> dengan <i>Node.js (Express)</i> dan <i>REST API</i>. • Mulai integrasi <i>Clouinary</i> untuk pengelolaan media.
9 - 10	<ul style="list-style-type: none"> • Mengembangkan <i>API</i> untuk Beranda, Tentang Kami, Developer, Simulasi KPR, Kontak Kami dan Cluster. • <i>Testing API</i> dengan <i>Postman</i>, memperbaiki bug, dan integrasi dengan <i>frontend</i>. • Mempersiapkan <i>hosting</i> dan memisahkan folder <i>frontend</i> dan <i>backend</i>.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu Ke -	Pekerjaan yang di lakukan
11 - 12	<ul style="list-style-type: none"> • Menambahkan fitur <i>CRUD</i> pada <i>CMS Admin</i> untuk konten seperti <i>gallery</i>, spesifikasi, dan cluster. • Implementasi autentikasi dengan <i>JWT</i>, serta fitur <i>login</i> dan proteksi <i>endpoint</i>. • Finalisasi tampilan <i>frontend</i> dan integrasi <i>backend</i>.
13 - 14	<ul style="list-style-type: none"> • Testing menyeluruh <i>frontend</i> dan <i>backend</i>, <i>fixing bug</i>, dan revisi berdasarkan review supervisor. • Finalisasi fitur <i>CMS Admin</i>, seperti cluster <i>gallery</i>, spesifikasi, dan kontak. • Persiapan presentasi final dan dokumentasi teknis.
15 - 16	<ul style="list-style-type: none"> • Finalisasi integrasi data <i>frontend</i> dan <i>backend</i> dan revisi <i>UI</i>. • Melakukan meeting teknis dan diskusi tentang <i>hosting</i>, domain, dan revisi akhir. • Mengumpulkan aset tambahan untuk implementasi final.

Pengerjaan projek website ini menggunakan perangkat lunak dan perangkat keras. Berikut ini merupakan perangkat lunak yang digunakan:

1. *Framework React* versi 19.0.0
2. *Framework Vite* versi 6.1.0
3. *Node JS* versi 20.18.0
4. *Express* versi 4.21.2
5. *Cloudinary* versi 1.41.3
6. *Tailwind CSS* versi 4.1.7
7. *Visual Studio Code* versi 1.100.3

8. *Github Desktop* versi 3.4.20 (x64) 1
9. *Postman* versi 1.48.2
10. Sistem Operasi *Windows 11* versi 24H2

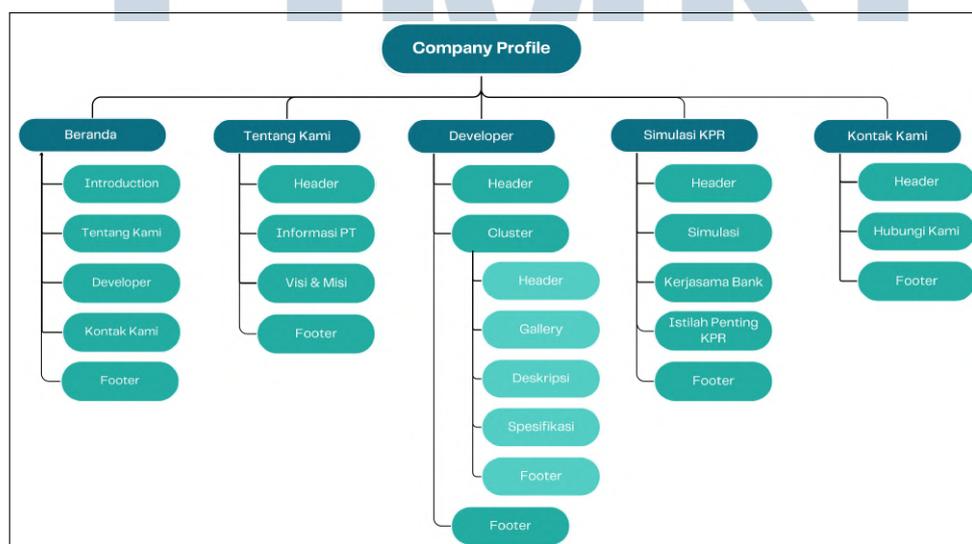
Berikut ini merupakan perangkat keras yang di gunakan selama proses kerja magang adalah laptop *Lenovo LOQ 15IRX9* dengan spesifikasi sebagai berikut:

1. *Processor Intel Core i5-13450HX*
2. *RAM 20 GB*
3. *Storage 512 GB*

3.3.2 Perancangan Site Map

Sitemap pada bagian ini berfungsi untuk memberikan gambaran menyeluruh mengenai struktur navigasi dari *website* yang dikembangkan, yaitu *Website Company Profile* dan *CMS Admin*. Perancangan ini menjadi dasar dalam membangun alur pengguna dan arsitektur konten yang efisien, sehingga memudahkan pengguna dalam mengakses informasi secara terstruktur.

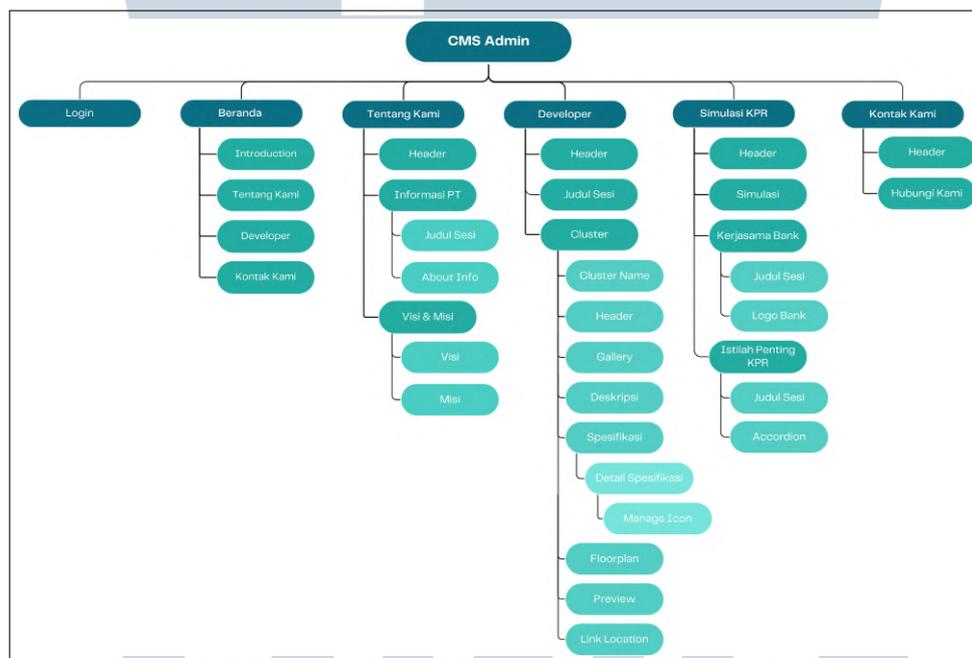
3.3.2.1 Sitemap Website Company Profile



Gambar 3.1. Company Profile

Gambar 3.1 menunjukkan struktur navigasi dari *Website Company Profile* PT Bayu Cahaya Gemilang. Terdapat lima halaman utama yaitu Beranda, Tentang Kami, Developer, Simulasi KPR, dan Kontak Kami. Masing-masing halaman terdiri dari beberapa sesi konten. Contohnya pada halaman Beranda terdiri dari *Introduction*, Tentang Kami, Developer, Kontak Kami, dan *Footer*. Sementara pada halaman Developer terdapat sesi *Cluster* yang memiliki subbagian penting seperti *Gallery*, Deskripsi, dan Spesifikasi. *Sitemap* ini disusun agar pengguna *website* dapat menjelajahi informasi secara logis dan hierarkis.

3.3.2.2 Sitemap Website CMS Admin



Gambar 3.2. CMS Admin

Gambar 3.2 memperlihatkan struktur *sitemap* dari *Website CMS Admin* yang digunakan untuk mengelola konten *website Company Profile*. *Sitemap* ini terdiri dari halaman *login*, lalu berlanjut ke halaman-halaman utama yang sama dengan *website* publik (Beranda, Tentang Kami, Developer, Simulasi KPR, dan Kontak Kami), namun dilengkapi dengan detail pengelolaan masing-masing sesi. Contohnya pada halaman Developer terdapat pengelolaan *cluster* seperti nama *cluster*, galeri, deskripsi, spesifikasi, hingga *link* lokasi. Di halaman Simulasi KPR, admin dapat mengelola judul sesi, logo bank, serta konten istilah penting melalui elemen *accordion*. Perancangan ini memungkinkan admin mengatur seluruh konten

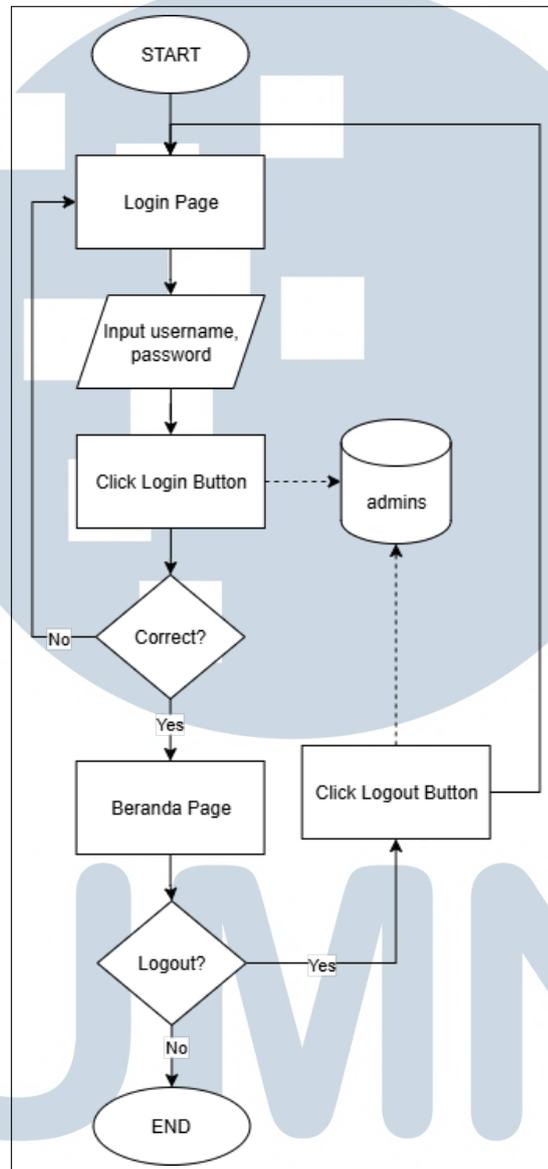
website secara dinamis dan modular.

3.3.3 Perancangan Flowchart

Flowchart pada bagian ini bertujuan untuk menggambarkan alur proses sistem *frontend CMS Admin* termasuk penambahan, pembaruan, dan penghapusan data yang ada. Dengan adanya *flowchart*, alur proses setiap page pada website yang berjalan dapat divisualisasikan secara lebih jelas dan sistematis. Setiap *flowchart* merepresentasikan langkah-langkah logis yang dilakukan oleh admin dalam mengelola konten *website* melalui antarmuka *Content Management System (CMS)*.



3.3.3.1 Flowchart Proses Login dan Logout

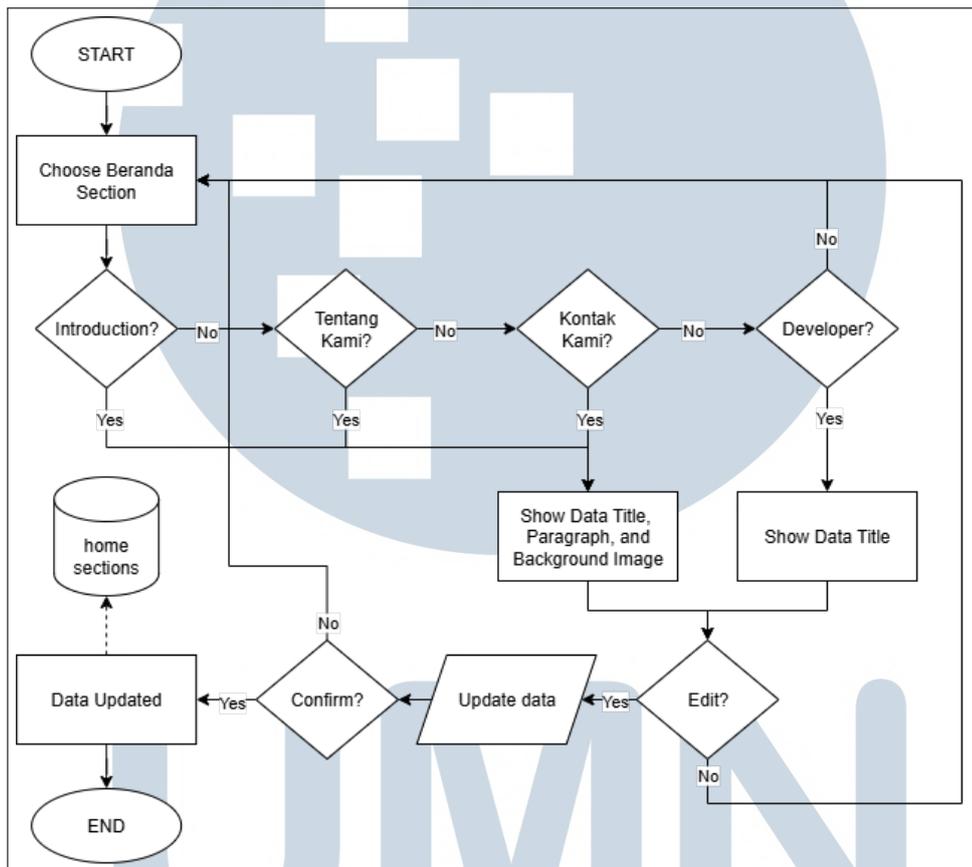


Gambar 3.3. Flowchart proses login dan logout

Gambar 3.3 ini menggambarkan alur proses *login* dan *logout* pada sistem *CMS Admin*. Proses dimulai dari halaman *login*, di mana pengguna memasukkan *username* dan *password*, lalu menekan tombol *login*. Sistem akan memvalidasi kredensial dengan mencocokkannya pada tabel *admins*. Jika data cocok, pengguna diarahkan ke halaman beranda dan dapat mengakses seluruh fitur *CMS Admin*; jika tidak, pengguna tetap berada di halaman login. Proses logout dilakukan dengan menekan tombol *logout*, yang akan mengakhiri sesi dan mengembalikan

pengguna ke halaman login. Flowchart ini dirancang untuk memastikan bahwa hanya pengguna yang telah terverifikasi yang dapat mengakses sistem, sehingga keamanan dan privasi konten tetap terjaga.

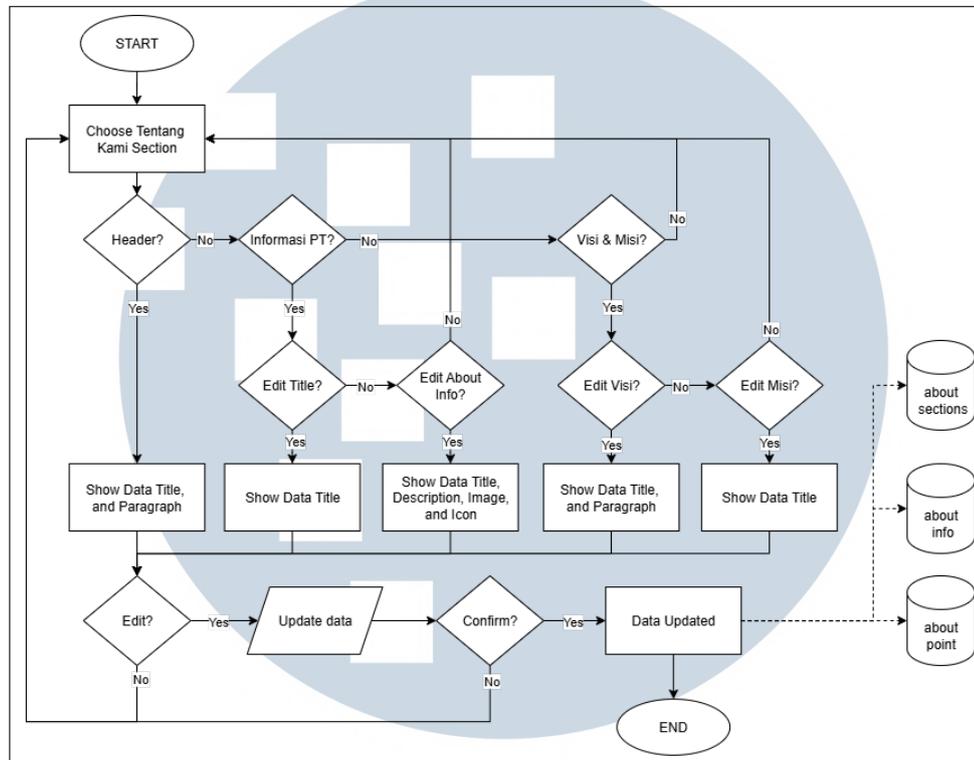
3.3.3.2 Flowchart Proses Edit Halaman Beranda



Gambar 3.4. Flowchart proses edit halaman beranda

Gambar 3.4 menjelaskan alur proses pengelolaan konten pada halaman Beranda (*Home*) melalui sistem *CMS Admin*. Pengguna dapat memilih salah satu dari empat *section* yang tersedia untuk diubah, kemudian sistem akan menampilkan data yang relevan, seperti *title*, *paragraph*, dan *image* (jika ada). Setelah melihat data yang ingin diperbarui, pengguna dapat memilih untuk mengedit konten, mengonfirmasi perubahan, lalu menekan tombol *update* untuk menyimpan perubahan ke *database home_sections*. Jika pengguna membatalkan proses atau tidak mengubah, maka sistem akan langsung diarahkan ke akhir proses. *Flowchart* ini dirancang untuk mempermudah pengelolaan konten statis pada halaman beranda secara terstruktur dan aman oleh admin yang berwenang.

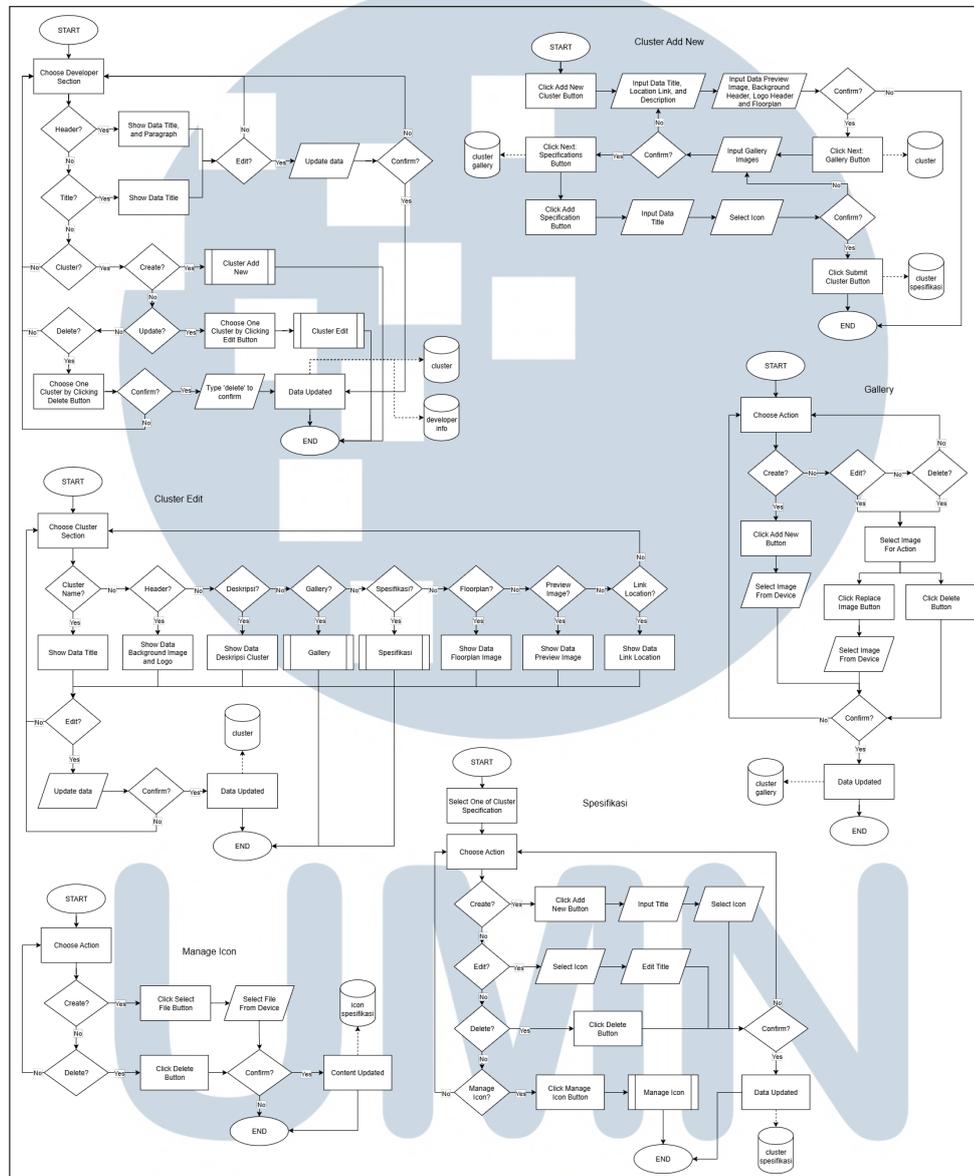
3.3.3.3 Flowchart Proses Edit Halaman Tentang Kami



Gambar 3.5. Flowchart proses edit halaman tentang kami

Gambar 3.5 menggambarkan alur proses pengelolaan konten pada halaman Tentang Kami melalui sistem *CMS Admin*. Pengguna dapat memilih salah satu dari tiga *section* yang tersedia, yaitu *header*, informasi umum perusahaan, dan visi & misi perusahaan. Setiap *section* memiliki jenis data yang berbeda, seperti *title*, *paragraph*, *image*, dan *icon*, yang diambil dari tabel *about_sections*, *about_info*, dan *about_point*. Setelah data ditampilkan, pengguna dapat mengubah konten, lalu menekan tombol *update* untuk menyimpan perubahan ke *database*. *Flowchart* ini dirancang untuk memberikan kemudahan dalam memisahkan dan mengelola konten berdasarkan struktur halaman, sehingga proses *edit* lebih terarah dan efisien.

3.3.3.4 Flowchart Proses Pengelolaan Halaman Developer dan Cluster



Gambar 3.6. Flowchart keseluruhan proses pengelolaan halaman Developer dan Cluster

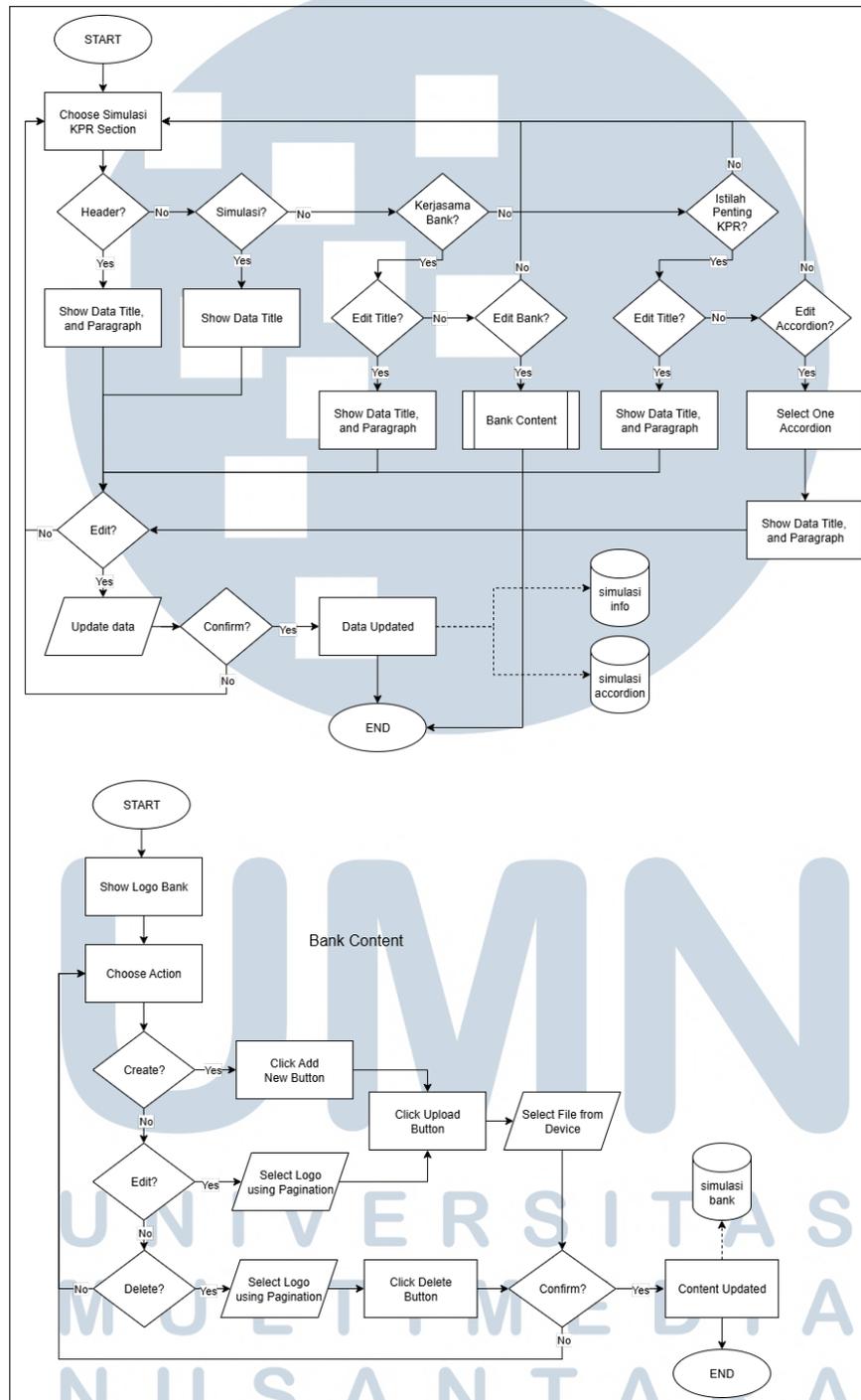
Gambar 3.6 menggambarkan alur proses utama dan subproses dalam pengelolaan konten halaman *Developer* dan *Cluster* melalui sistem *CMS Admin*. Proses dimulai dari pilihan awal untuk mengelola bagian informasi *Developer* atau bagian *Cluster*. Pada bagian *Developer*, admin dapat memperbarui data berupa judul dan paragraf yang disimpan ke dalam tabel *developer_info*. Sedangkan pada bagian *Cluster*, tersedia beberapa subproses, mulai dari penambahan cluster baru, pengeditan data cluster, pengelolaan galeri gambar, spesifikasi teknis, hingga ikon

spesifikasi yang digunakan pada halaman publik.

Masing-masing subproses mencakup tiga aksi utama yaitu *Create*, *Edit*, dan *Delete*, dengan validasi dan konfirmasi di setiap tahap penyimpanan data. Data yang dikelola akan disimpan ke tabel terkait seperti *cluster*, *cluster_gallery*, *cluster_spesifikasi*, dan *icon_spesifikasi*. Flowchart ini dirancang secara terstruktur dan modular untuk memastikan admin dapat mengelola seluruh komponen halaman developer secara fleksibel dan efisien, sekaligus menjaga konsistensi data dan integritas tampilan pada website publik.



3.3.3.5 Flowchart Proses Pengelolaan Halaman Simulasi KPR



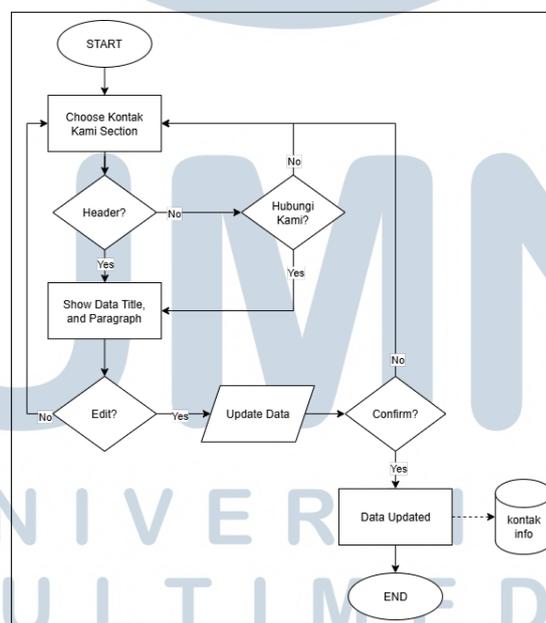
Gambar 3.7. Flowchart keseluruhan proses pengelolaan halaman Simulasi KPR

Gambar 3.7 menggambarkan alur utama dan subproses dalam pengelolaan konten halaman Simulasi KPR melalui sistem *CMS Admin*. Halaman Simulasi

KPR terdiri dari beberapa *section*, yaitu *Header*, *Judul Simulasi*, *Kerjasama Bank*, dan *Istilah Penting KPR (accordion content)*. Admin dapat mengelola tiap bagian secara terpisah untuk memastikan konten tetap relevan dan akurat. Pada bagian *Kerjasama Bank*, tersedia fitur pengelolaan logo bank dengan tiga aksi utama: *Create*, *Edit*, dan *Delete*. Setiap perubahan logo akan disimpan pada tabel *simulasi_bank*. Sementara itu, bagian *accordion* untuk istilah penting KPR dikelola melalui tabel *simulasi_accordion*, di mana admin dapat menambahkan, memperbarui, atau menghapus konten penjelasan istilah.

Seluruh proses dilengkapi dengan konfirmasi tindakan, validasi data, serta pemisahan modul pengelolaan agar admin dapat mengatur struktur informasi secara efisien dan terkontrol. Flowchart ini bertujuan memudahkan pengelolaan konten Simulasi KPR dengan pendekatan modular, fleksibel, dan aman, serta menjaga konsistensi tampilan dan informasi yang ditampilkan pada *website Company Profile*.

3.3.3.6 Flowchart Proses Edit Halaman Kontak Kami

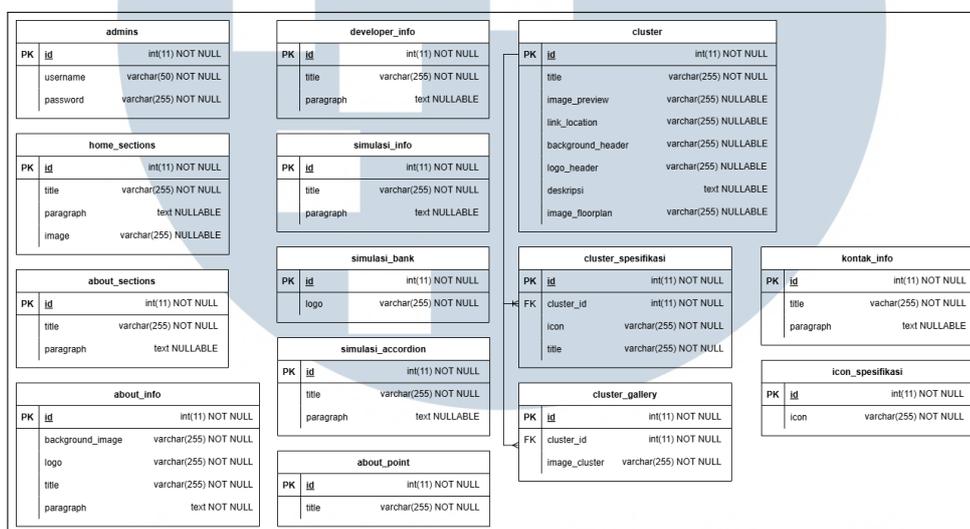


Gambar 3.8. Flowchart proses edit halaman kontak kami

Gambar 3.8 menggambarkan alur proses pengelolaan konten pada halaman Kontak Kami melalui sistem *CMS Admin*. Terdapat dua *section* yang dapat diedit oleh admin, yaitu bagian header dan konten hubungi kami, yang masing-masing menyimpan data berupa *title* dan *paragraph*. Setelah memilih salah

satu *section*, sistem akan menampilkan data yang ada, kemudian admin dapat melakukan perubahan, mengonfirmasi pembaruan, dan menyimpannya ke dalam tabel *kontak_info*. Jika tidak ada perubahan atau proses dibatalkan, maka sistem akan kembali ke akhir alur tanpa menyimpan data. *Flowchart* ini dirancang untuk memastikan bahwa setiap informasi kontak yang ditampilkan pada *website Company Profile* selalu akurat, mudah diperbarui, dan tersimpan secara terstruktur.

3.3.4 Perancangan Skema Basis Data



Gambar 3.9. Database schema company profile website

Gambar 3.9 menunjukkan *Database Schema* yang digunakan sebagai fondasi penyimpanan data untuk kedua *website* PT Bayu Cahaya Gemilang, yaitu *website Company Profile* dan *CMS Admin*. Setiap tabel memiliki fungsi spesifik dan struktur relasi antar tabel dibuat dengan mempertimbangkan efisiensi dan kemudahan integrasi ke dalam antarmuka pengguna. Berikut adalah penjelasan lengkap untuk tiap tabelnya:

1. Tabel *admins*: Menyimpan data autentikasi untuk pengguna *CMS Admin*, berupa *username* dan *password*. Kolom *password* berisi hasil *hash* yang dienkripsi menggunakan *bcrypt* untuk menjamin keamanan akses.
2. Tabel *home_sections*: Menyimpan konten untuk 4 *section* pada halaman Beranda, seperti *title*, *paragraph*, dan *image*. Data ini dikelola melalui *CMS Admin* dan ditampilkan di *website Company Profile*.

3. Tabel *about_sections*: Menyimpan judul dan deskripsi untuk seluruh *section*, termasuk header pada halaman Tentang Kami. Konten ini dikelola melalui *CMS Admin* dan ditampilkan secara publik.
4. Tabel *about_info*: Digunakan untuk mengelola konten pada *section* ke-2 halaman Tentang Kami, seperti *background_image*, *logo*, *title*, dan deskripsi.
5. Tabel *about_point*: Digunakan untuk menampilkan daftar poin-poin pada *section* ke-3 halaman Tentang Kami, seperti misi perusahaan. Tabel ini hanya menyimpan *title* setiap poin.
6. Tabel *developer_info*: Menyimpan konten header dan deskripsi pada halaman Developer, termasuk *title* dan *paragraph*. Dikelola melalui *CMS Admin* dan ditampilkan di halaman publik.
7. Tabel *simulasi_info*: Menyimpan informasi pada keempat *section* halaman Simulasi KPR, termasuk header, dengan format *title* dan *paragraph*. Data ini dikelola dari *CMS Admin*.
8. Tabel *simulasi_bank*: Menyimpan data logo bank untuk *section* ke-3 halaman Simulasi KPR. Digunakan untuk menampilkan daftar bank yang bekerja sama dengan perusahaan dalam bentuk logo.
9. Tabel *simulasi_accordion*: Menyimpan daftar istilah penting dalam bentuk *accordion* pada *section* ke-4 halaman Simulasi KPR. Setiap entri memiliki *title* dan *paragraph* sebagai penjelasan istilah.
10. Tabel *kontak_info*: Menyimpan judul dan deskripsi pada dua *section* halaman Kontak Kami, termasuk header. Dikelola melalui *CMS Admin* dan ditampilkan secara publik.
11. Tabel *cluster*: Merupakan tabel utama untuk menyimpan data masing-masing cluster properti, seperti *title*, *image_preview*, *link_location*, *background_header*, *logo_header*, *deskripsi*, dan *image_floorplan*. Konten ini digunakan pada halaman detail cluster.
12. Tabel *cluster_spesifikasi*: Menyimpan daftar spesifikasi teknis untuk masing-masing cluster, seperti *icon* dan *title*. Tabel ini memiliki *cluster_id* sebagai *foreign key* dari tabel *cluster*.

13. Tabel *cluster_gallery*: Menyimpan koleksi gambar untuk setiap cluster. Setiap gambar terkait dengan *cluster_id* sebagai *foreign key* dari tabel *cluster*.
14. Tabel *icon_spesifikasi*: Digunakan untuk manajemen ikon spesifikasi yang dapat dipilih secara dinamis saat menambahkan konten spesifikasi cluster. Setiap data *icon* berupa URL dari layanan *Cloudinary*.

3.3.5 Perancangan Antarmuka Pengguna

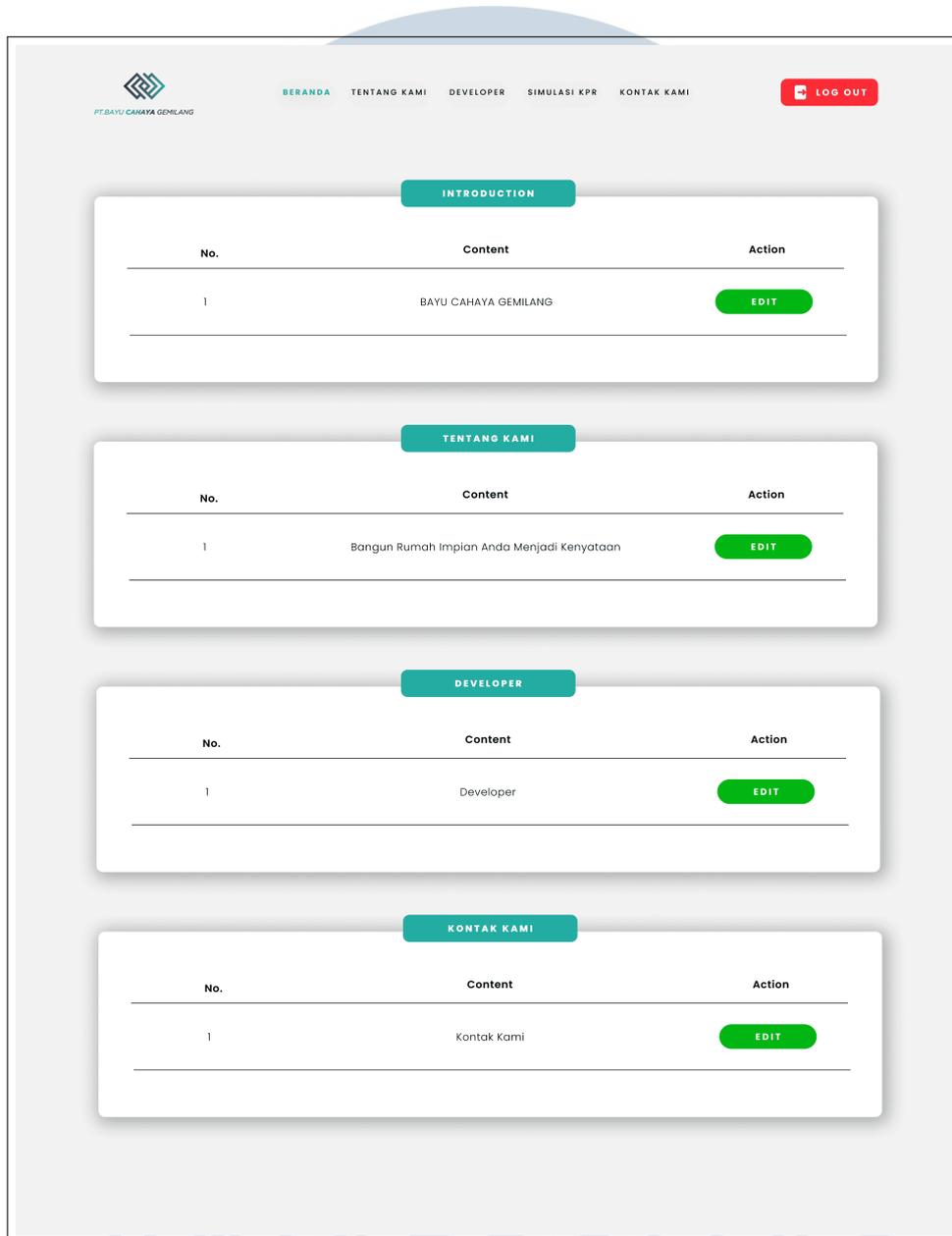
Pada bagian ini ditampilkan hasil rancangan antarmuka (*user interface*) yang dibuat untuk *website CMS Admin* maupun *Company Profile* menggunakan *Figma* sebagai acuan awal dalam proses pengembangan *website*. Desain ini disusun berdasarkan kebutuhan perusahaan dan prinsip desain yang mengacu pada teori *Eight Golden Rules of Interface Design* oleh Ben Shneiderman, seperti konsistensi, kontrol pengguna, dan kemudahan navigasi [4]. Setiap halaman dirancang agar memiliki struktur yang intuitif dan estetis, sehingga memudahkan pengguna (admin) dalam mengelola konten.

Perbandingan antara desain *Figma* dan hasil implementasi juga dilakukan untuk menunjukkan sejauh mana kesesuaian antara rancangan dan produk akhir. Dalam proses implementasi *website*, beberapa penyesuaian atau revisi mungkin terjadi, baik karena masukan dari supervisor, pertimbangan teknis, maupun peningkatan fungsionalitas. Meskipun demikian, setiap perubahan tetap mengacu pada prinsip desain yang baik dan fungsionalitas tetap berjalan dengan sesuai.

3.3.5.1 Tampilan Antarmuka CMS Admin

Pada bagian ini dijelaskan perancangan tampilan antarmuka untuk sistem *CMS Admin* yang digunakan untuk mengelola seluruh konten *website* secara dinamis. Penjelasan difokuskan pada tampilan-tampilan utama dan desain yang memiliki karakteristik berbeda atau unik dibandingkan komponen lainnya. Setiap elemen antarmuka dianalisis berdasarkan penerapan prinsip *Eight Golden Rules of Interface Design* oleh Ben Shneiderman, seperti konsistensi, keterbacaan, serta kemudahan navigasi, guna memastikan pengalaman pengguna yang optimal. Desain yang dihasilkan tidak hanya memperhatikan estetika, namun juga bertujuan untuk mendukung efektivitas dan efisiensi kerja admin dalam mengelola sistem.

3.3.5.1.1 Halaman Beranda

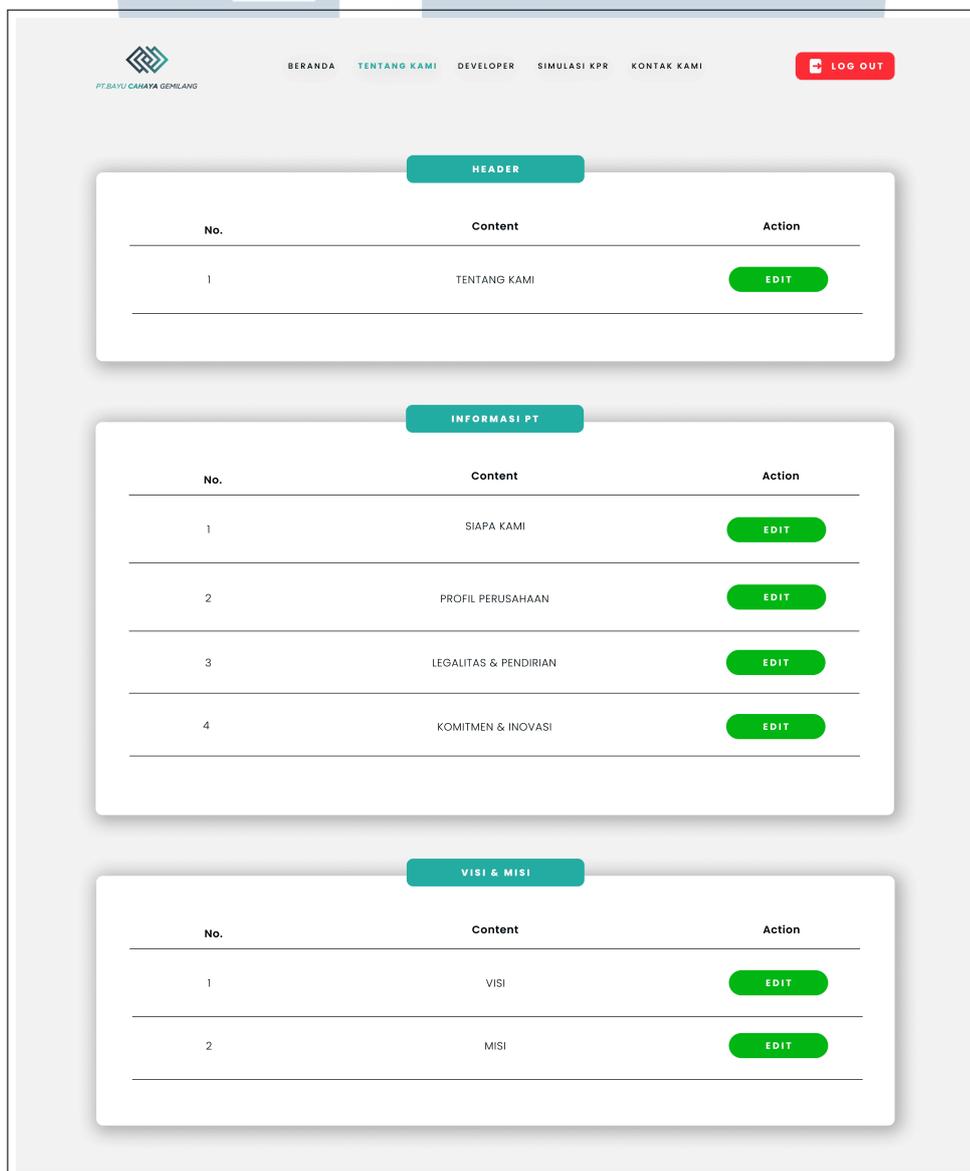


Gambar 3.10. Desain tampilan halaman Beranda pada CMS Admin

Gambar 3.10 memperlihatkan desain awal halaman Beranda untuk *CMS Admin*. Halaman ini dirancang agar admin dapat melihat seluruh daftar *section* yang tersedia pada halaman Beranda *website Company Profile*, seperti Introduction, Tentang Kami, Kontak Kami, dan Developer. Masing-masing baris dalam tabel dilengkapi dengan informasi singkat dan tombol aksi seperti *Edit* untuk mendukung proses pembaruan konten yang efisien.

Desain halaman ini menerapkan prinsip *strive for consistency* melalui penggunaan tabel, label, dan tombol yang memiliki struktur dan penempatan seragam dengan halaman lain. Selain itu, penggunaan tag judul pada masing-masing sesi memberikan *offer informative feedback* kepada pengguna mengenai perbedaan tiap sesi yang ada. Dengan susunan konten yang terstruktur, pengguna dapat fokus pada satu area kerja dalam satu waktu, mendukung prinsip *reduce short-term memory load*.

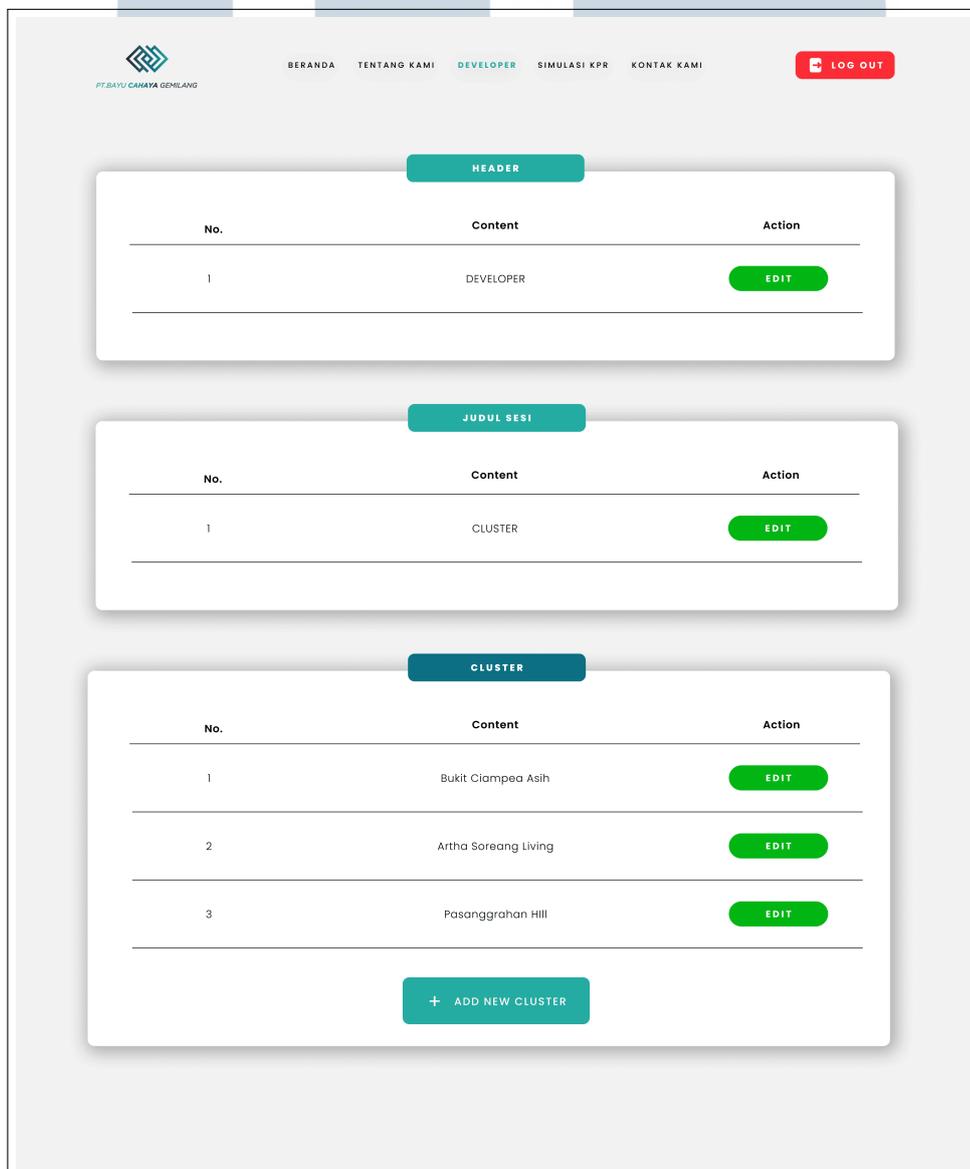
3.3.5.1.2 Halaman Tentang Kami



Gambar 3.11. Desain tampilan halaman Tentang Kami pada CMS Admin

Gambar 3.11 memperlihatkan rancangan visual halaman Tentang Kami yang disusun secara modular, di mana setiap sesi dipisahkan dalam tampilan kartu yang rapi. Penempatan elemen dilakukan dengan mempertimbangkan urutan logis dan hirarki informasi yang jelas untuk memudahkan pemindaian visual oleh pengguna. Warna aksen hijau digunakan secara konsisten sebagai penanda visual antar *section*, selaras dengan prinsip *strive for consistency* agar antarmuka terasa familiar.

3.3.5.1.3 Halaman Developer



Gambar 3.12. Desain tampilan halaman Developer pada CMS Admin

Gambar 3.12 menampilkan rancangan halaman Developer yang terdiri atas dua sesi utama, yaitu Header dan Cluster. Tampilan tabel pada sesi Cluster dibuat lebih menonjol secara visual dibandingkan sesi lain karena memiliki fungsionalitas tambahan, seperti navigasi ke halaman manajemen konten cluster. Perbedaan gaya ini membantu pengguna mengenali jenis interaksi yang akan dilakukan, sekaligus memberi kesan hierarki antar konten. Penggunaan tombol aksi berwarna hijau untuk *edit* dan *add new* memperkuat prinsip *offer informative feedback*, karena setiap aksi diberi tanda visual yang jelas. Selain itu, penempatan elemen yang simetris dan struktur antarmuka yang konsisten mendukung prinsip *strive for consistency*, sehingga pengguna dapat bernavigasi tanpa kebingungan.

3.3.5.1.3.1 Fitur Add New Cluster (CREATE)

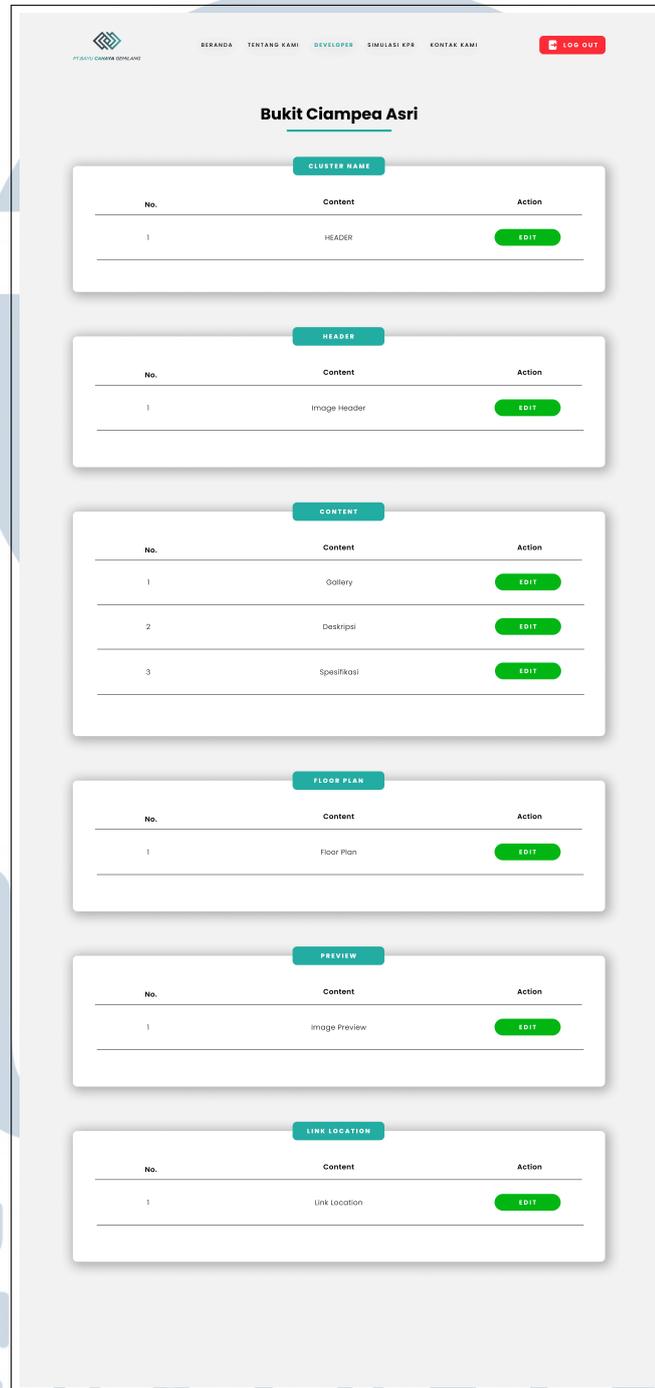
The screenshot shows a web interface for adding a new cluster. At the top, there is a navigation bar with links for BERANDA, TENTANG KAMI, DEVELOPER, SIMULASI KPR, and KONTAK KAMI, along with a LOG OUT button. The main content area is titled 'ADD NEW CLUSTER' and features a progress indicator with three steps: 1. Cluster Info, 2. Gallery, and 3. Specifications. The 'Cluster Info' step is currently active. The form includes several input fields: 'Title' (required), 'Location Link', 'Description', 'Preview Image' (with a 'Choose File' button and 'No file chosen' text), 'Background Header Image' (with a 'Choose File' button and 'No file chosen' text), 'Logo Header' (with a 'Choose File' button and 'No file chosen' text), and 'Floor Plan' (with a 'Choose File' button and 'No file chosen' text). At the bottom of the form, there are 'Cancel' and 'Next: Gallery' buttons.

Gambar 3.13. Desain fitur Add New Cluster step pertama

Desain fitur *Add New Cluster* terdiri dari tiga langkah yang ditampilkan sebagai halaman terpisah pada Gambar 3.13 hingga step *Specifications*. Setiap langkah difokuskan pada satu jenis *input* agar proses tidak membebani pengguna sekaligus, sesuai dengan prinsip *reduce short-term memory load*. Dengan memecah proses kompleks menjadi bagian kecil yang mudah dipahami, pengguna dapat menyelesaikan tugas tanpa harus mengingat terlalu banyak informasi dalam satu waktu. Label navigasi seperti "1", "2", dan "3" digunakan untuk memastikan *strive for consistency*, sehingga pengguna dapat memahami alur dengan pola yang seragam. Selain itu, setiap aksi utama seperti *submit* atau *next* memberikan respon visual atau pesan status, menerapkan prinsip *offer informative feedback* agar pengguna selalu mengetahui bahwa data mereka telah diterima atau bahwa langkah berikutnya telah berhasil dimuat. Untuk meningkatkan pengalaman pengguna, sistem juga dirancang agar pengguna dapat kembali ke langkah sebelumnya jika diperlukan, sesuai dengan prinsip *permit easy reversal of actions*. Dengan demikian, pengguna memiliki kontrol penuh atas proses yang mereka jalani, sejalan dengan prinsip *support internal locus of control*.



3.3.5.1.4 Halaman Cluster

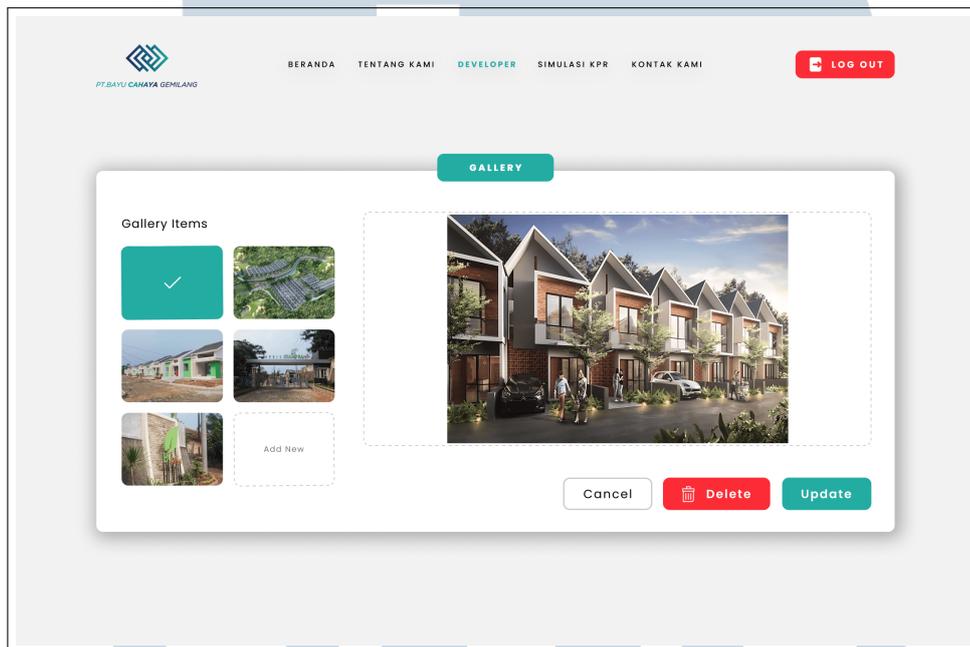


Gambar 3.14. Desain tampilan halaman Cluster pada CMS Admin

Gambar 3.14 menampilkan struktur halaman Cluster yang terdiri dari beberapa sesi, seperti Nama Cluster, Header, Gallery, dan lainnya. Desain ini menerapkan prinsip *strive for consistency*, di mana setiap sesi ditampilkan dalam

bentuk tabel dengan urutan dan gaya visual yang konsisten. Pengelompokan konten ke dalam sesi-sesi berbeda bertujuan untuk *reduce short-term memory load*, agar pengguna tidak perlu mengingat struktur keseluruhan sekaligus dan dapat fokus pada satu sesi dalam satu waktu.

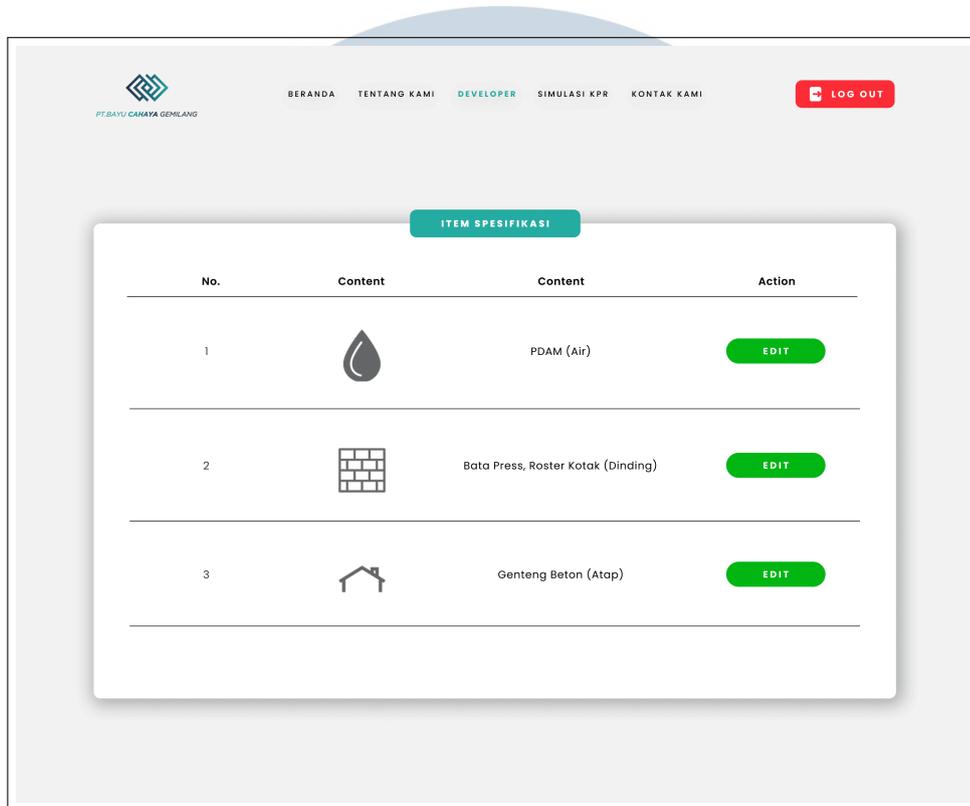
3.3.5.1.4.1 Sesi Gallery



Gambar 3.15. Desain sesi Gallery pada halaman Cluster

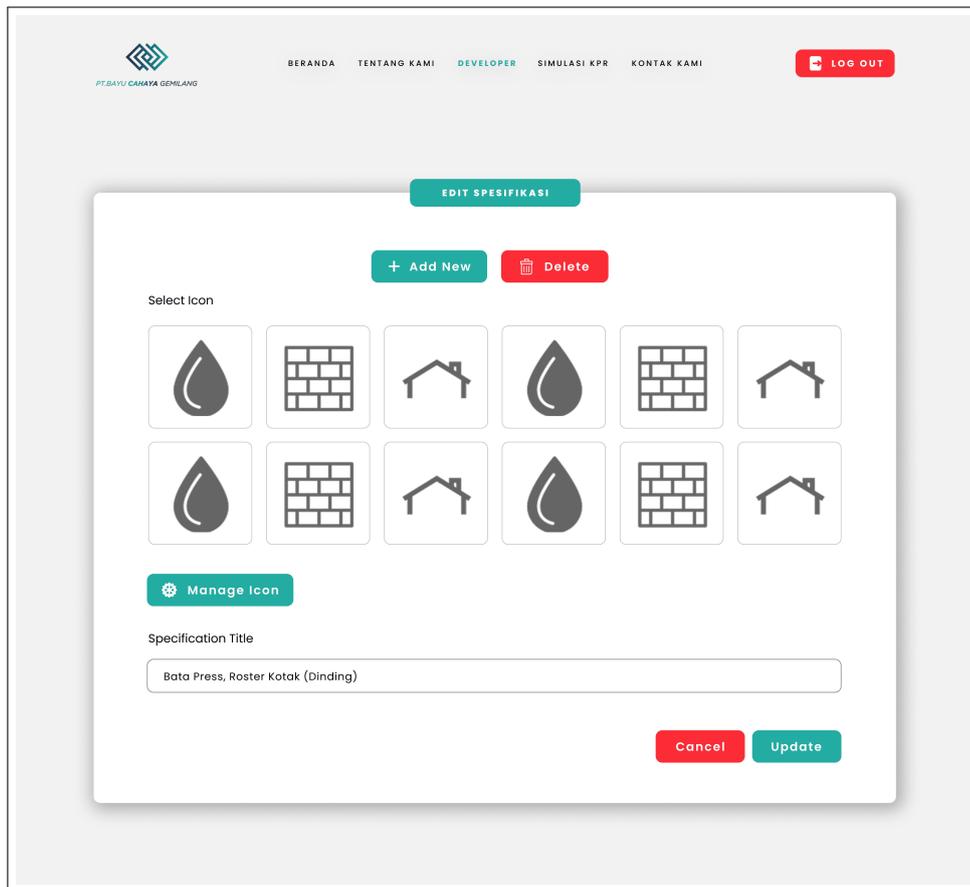
Gambar 3.15 menunjukkan tampilan galeri gambar dalam bentuk *thumbnail*, yang memudahkan pengguna dalam memilih dan melihat gambar. Pemisahan gambar dalam bentuk visual ini membantu *reduce short-term memory load*, karena pengguna tidak perlu mengingat nama *file* atau urutan upload. Selain itu, struktur tombol dan *layout* konsisten dengan sesi lain sebagai bentuk penerapan *strive for consistency*.

3.3.5.1.4.2 Sesi Spesifikasi



Gambar 3.16. Desain sesi Spesifikasi pada halaman Cluster

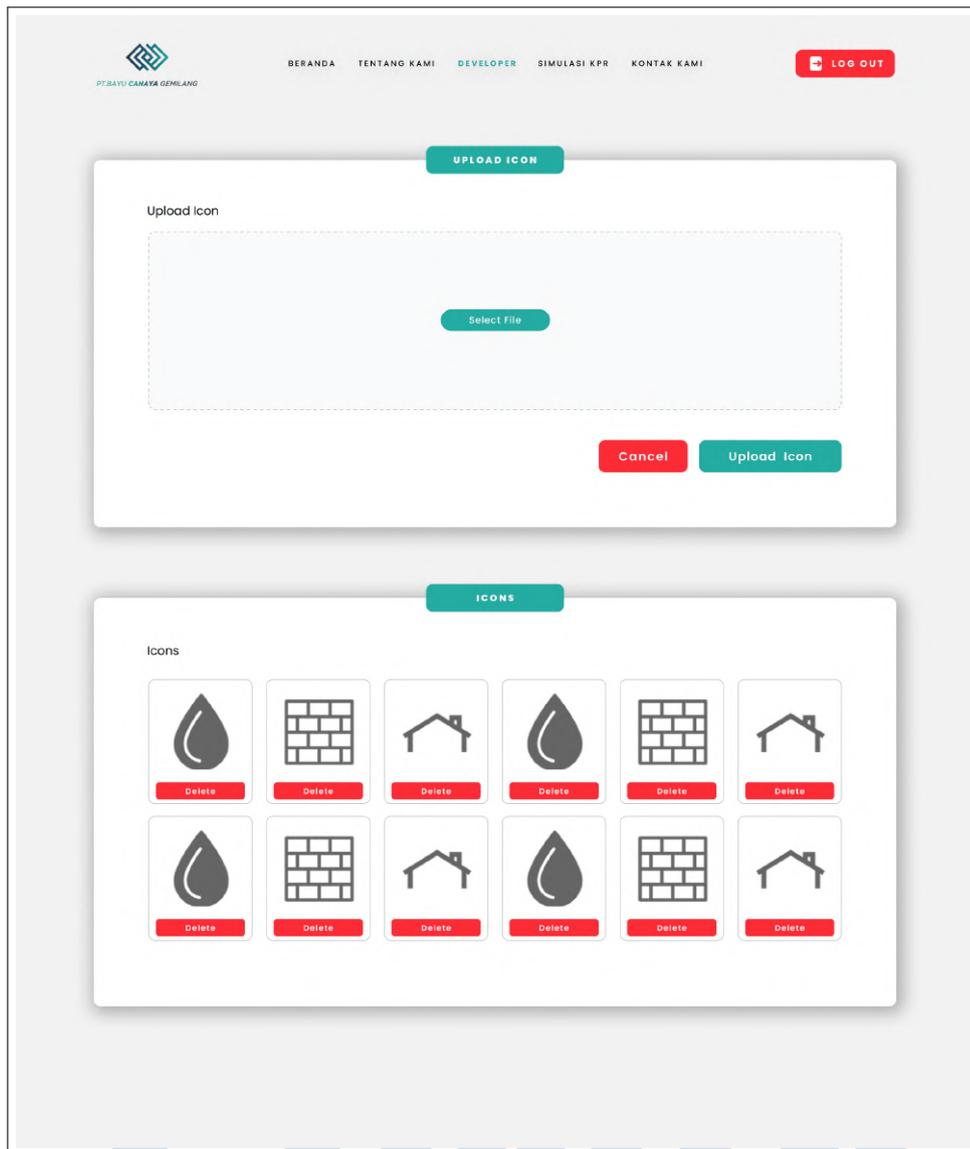
UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.17. Desain sesi Spesifikasi Detail pada halaman Cluster

Desain sesi spesifikasi ditunjukkan pada Gambar 3.16, dan 3.17. Pengguna dapat melihat daftar item spesifikasi yang terdiri dari ikon dan deskripsi singkat. Tiap item dilengkapi tombol aksi untuk melakukan *edit* atau *delete*, serta terdapat tombol *Add New* yang mengarahkan ke halaman tambah item baru. Tampilan detail spesifikasi disusun agar memisahkan dengan jelas antara area input judul dan pemilihan ikon, sesuai prinsip *reduce short-term memory load*. Pengguna juga memiliki kendali untuk langsung menambahkan atau menghapus item, serta mengakses tombol *Manage Icon* yang mengarah ke halaman khusus manajemen ikon.

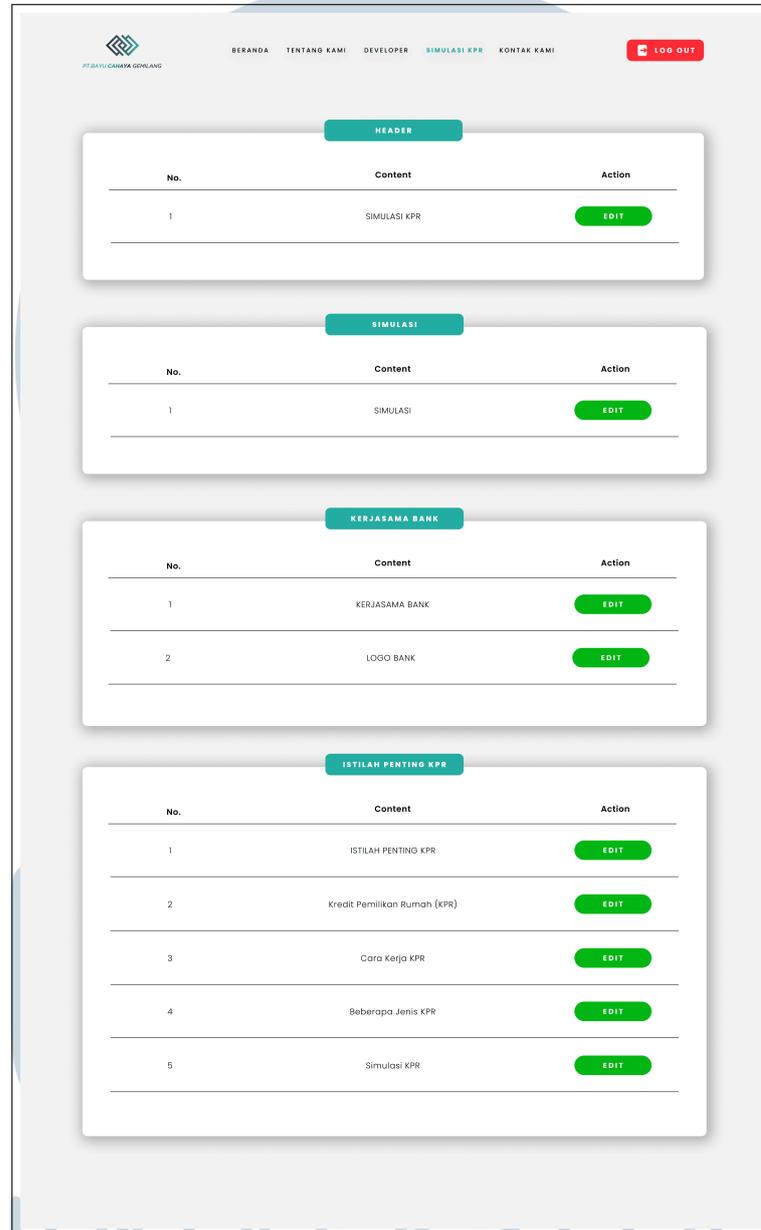
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.18. Desain sesi Manajemen Ikon pada halaman Cluster

Gambar 3.18 memperlihatkan tampilan galeri ikon dengan visualisasi langsung dan tombol *Delete* pada tiap ikon. Hal ini memperkuat prinsip *support internal locus of control* karena pengguna dapat mengelola sendiri variasi ikon yang tersedia. Keseluruhan sesi mempertahankan konsistensi tata letak dan warna tombol aksi, mendukung prinsip *strive for consistency* dan *offer informative feedback* melalui umpan balik visual secara langsung saat pengguna memilih ikon atau melakukan aksi tertentu.

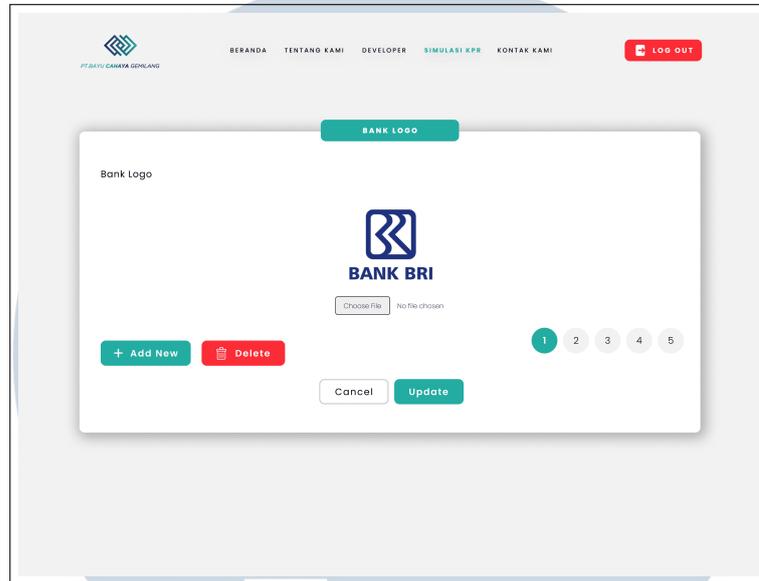
3.3.5.1.5 Halaman Simulasi KPR



Gambar 3.19. Desain tampilan halaman Simulasi KPR pada CMS Admin

Desain halaman Simulasi KPR ditampilkan pada Gambar 3.19, yang terdiri dari empat sesi utama yang disusun dalam format tabel. Struktur antar sesi dibuat konsisten dengan halaman lainnya, mendukung prinsip *strive for consistency*. Penggunaan warna tombol dan jarak antar elemen dibuat seragam untuk memberikan pengalaman yang stabil di seluruh halaman *CMS Admin*.

3.3.5.1.5.1 Sesi Kerjasama Bank

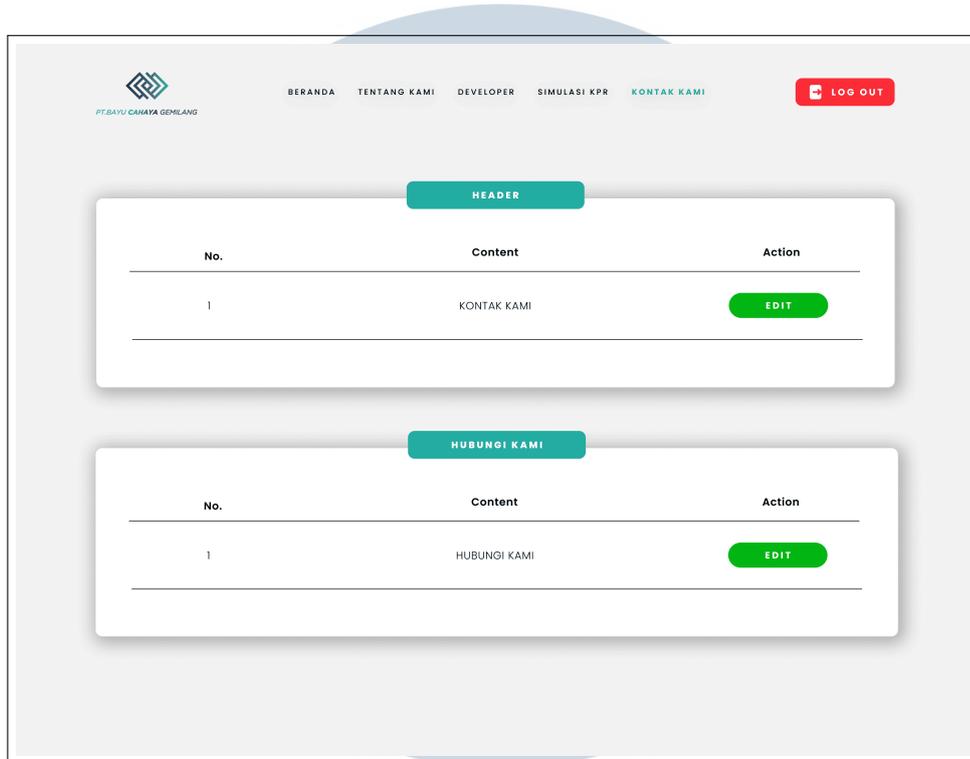


Gambar 3.20. Desain sesi Kerjasama Bank pada halaman Simulasi KPR

Pada Gambar 3.20, sesi ini didesain untuk menampilkan logo-logo bank secara individual menggunakan sistem *pagination*. Penempatan tombol *Add New* dan *Delete* di sisi kiri, sementara tombol navigasi di sisi kanan, memberikan kejelasan distribusi fungsi sesuai prinsip *support internal locus of control*. Tombol aksi utama seperti *Cancel* dan *Update* diposisikan di tengah bawah untuk mengarahkan pengguna ke penyelesaian tugas, sesuai prinsip *design dialogs to yield closure*.

U M 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

3.3.5.1.6 Halaman Kontak Kami



Gambar 3.21. Desain tampilan halaman Kontak Kami pada CMS Admin

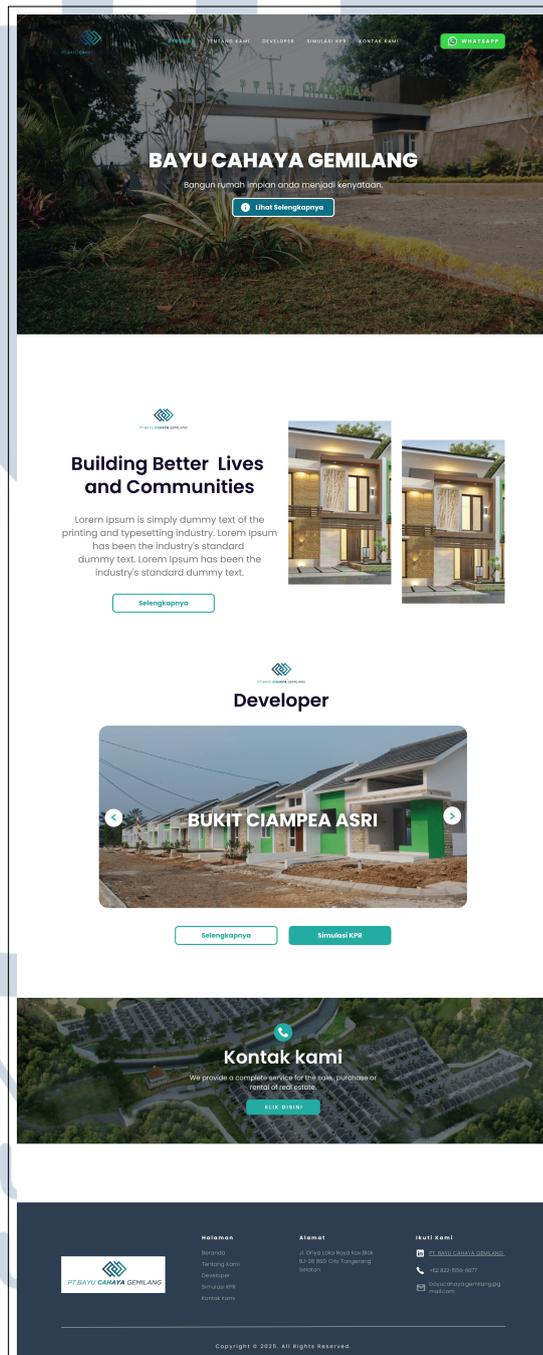
Gambar 3.21 menampilkan tampilan utama halaman Kontak Kami pada *CMS Admin*, yang terdiri dari dua tabel *section* dengan desain yang konsisten dengan halaman lain. Struktur halaman ini dirancang agar mudah dikenali dan dioperasikan oleh pengguna, dengan penggunaan layout dan tipografi yang seragam. Konsistensi desain ini membantu mengurangi beban kognitif pengguna dalam menavigasi antar sesi.

3.3.5.2 Tampilan Antarmuka Company Profile

Bagian ini membahas hasil perancangan tampilan antarmuka untuk *website Company Profile* PT Bayu Cahaya Gemilang yang ditujukan bagi publik atau calon pelanggan. Desain disusun secara menyeluruh dengan mempertimbangkan prinsip *Eight Golden Rules of Interface Design*, seperti konsistensi visual, struktur informasi yang jelas, dan kemudahan dalam menjelajahi halaman. Penjelasan difokuskan pada setiap halaman utama dan sesi penting yang memiliki peran strategis dalam menyampaikan informasi perusahaan. Desain antarmuka ini

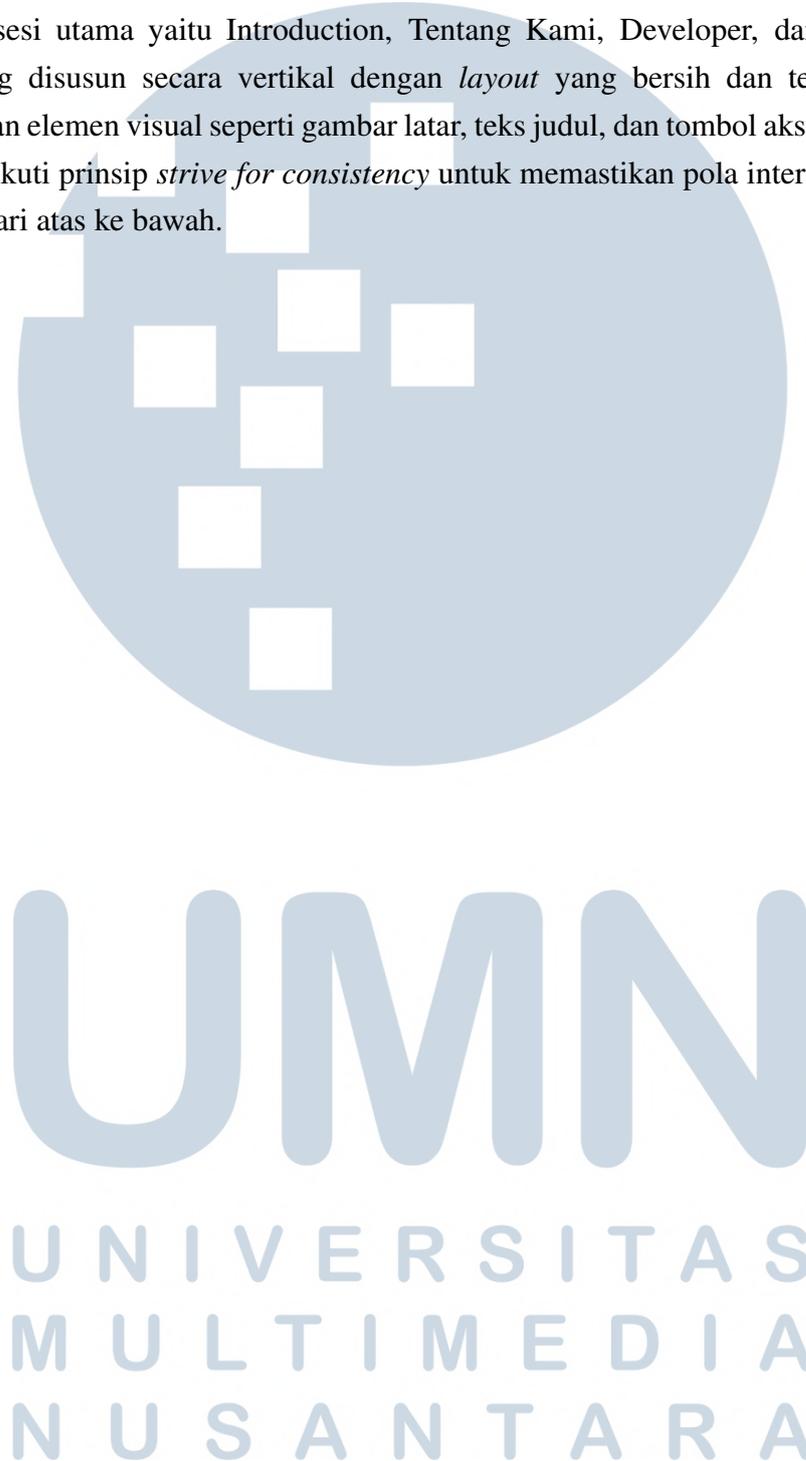
diharapkan mampu memperkuat citra profesional perusahaan serta meningkatkan keterlibatan pengguna saat mengakses informasi secara digital.

3.3.5.2.1 Halaman Beranda



Gambar 3.22. Desain halaman Beranda pada website Company Profile

Gambar 3.22 menampilkan keseluruhan desain halaman Beranda pada *website Company Profile* PT Bayu Cahaya Gemilang. Halaman ini terdiri dari beberapa sesi utama yaitu Introduction, Tentang Kami, Developer, dan Kontak Kami yang disusun secara vertikal dengan *layout* yang bersih dan terstruktur. Penempatan elemen visual seperti gambar latar, teks judul, dan tombol aksi di setiap sesi mengikuti prinsip *strive for consistency* untuk memastikan pola interaksi yang seragam dari atas ke bawah.



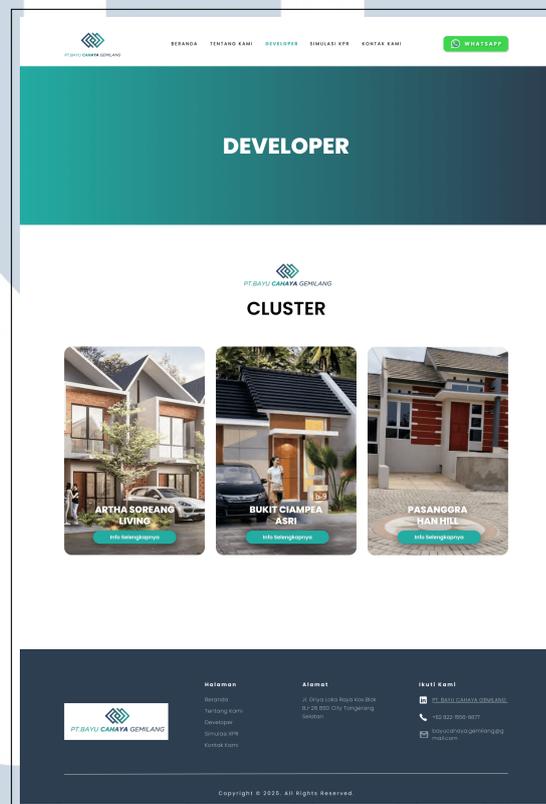
3.3.5.2.2 Halaman Tentang Kami



Gambar 3.23. Desain halaman Tentang Kami pada website Company Profile

Gambar 3.23 memperlihatkan tampilan lengkap dari halaman Tentang Kami pada website *Company Profile*. Halaman ini dibagi ke dalam dua bagian utama yaitu sesi Informasi Perusahaan dan sesi Visi & Misi. Masing-masing sesi didesain menggunakan elemen visual yang mendukung keterbacaan serta penyampaian pesan perusahaan secara profesional. Penempatan ikon, teks, dan gambar dibuat seragam antar sesi sehingga menjaga prinsip *strive for consistency*.

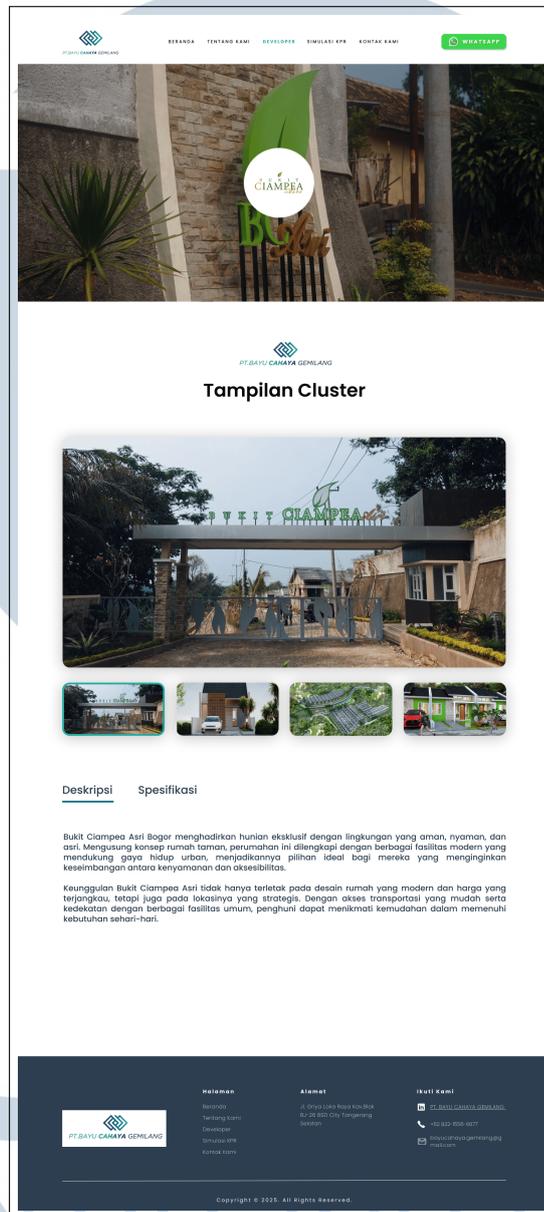
3.3.5.2.3 Halaman Developer



Gambar 3.24. Desain halaman Developer pada website Company Profile

Gambar 3.24 menampilkan halaman Developer pada *website Company Profile* yang hanya memuat satu sesi, yaitu sesi Cluster. Secara keseluruhan, struktur halaman ini tetap menyertakan komponen *navbar*, *header*, dan *footer* yang konsisten dengan halaman lainnya, menjaga prinsip *strive for consistency* dalam antarmuka pengguna. Penempatan elemen antar bagian juga dijaga agar tetap seimbang dan rapi, sehingga mendukung keterbacaan serta estetika desain.

3.3.5.2.4 Halaman Cluster

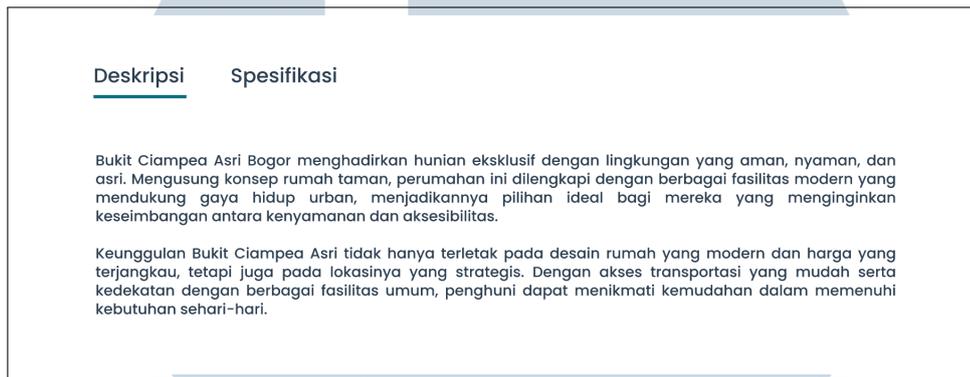


Gambar 3.25. Desain halaman Cluster pada website Company Profile

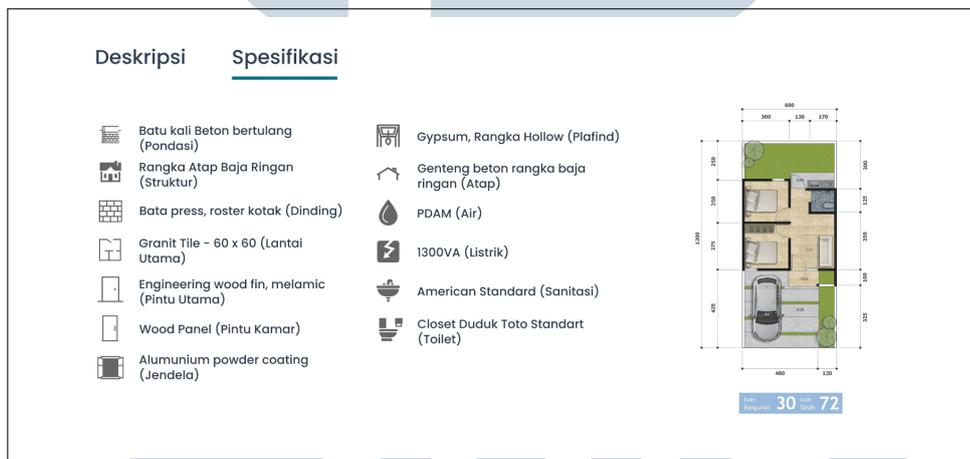
Gambar 3.25 menampilkan keseluruhan halaman Cluster pada *website Company Profile* yang menampilkan informasi lengkap mengenai salah satu proyek hunian, yaitu Bukit Ciampea Asri. Halaman ini terdiri dari beberapa sesi utama seperti *Header*, *Gallery*, serta *Deskripsi* dan *Spesifikasi*. Struktur kontennya disusun secara hierarkis dengan margin antar elemen yang proporsional sehingga pengguna dapat membaca informasi secara bertahap tanpa merasa terbebani.

Desain halaman ini juga memanfaatkan prinsip *reduce short-term memory load* dengan membagi konten ke dalam tab terpisah (Deskripsi dan Spesifikasi) untuk menghindari kelebihan informasi dalam satu waktu.

3.3.5.2.4.1 Sesi Deskripsi & Spesifikasi



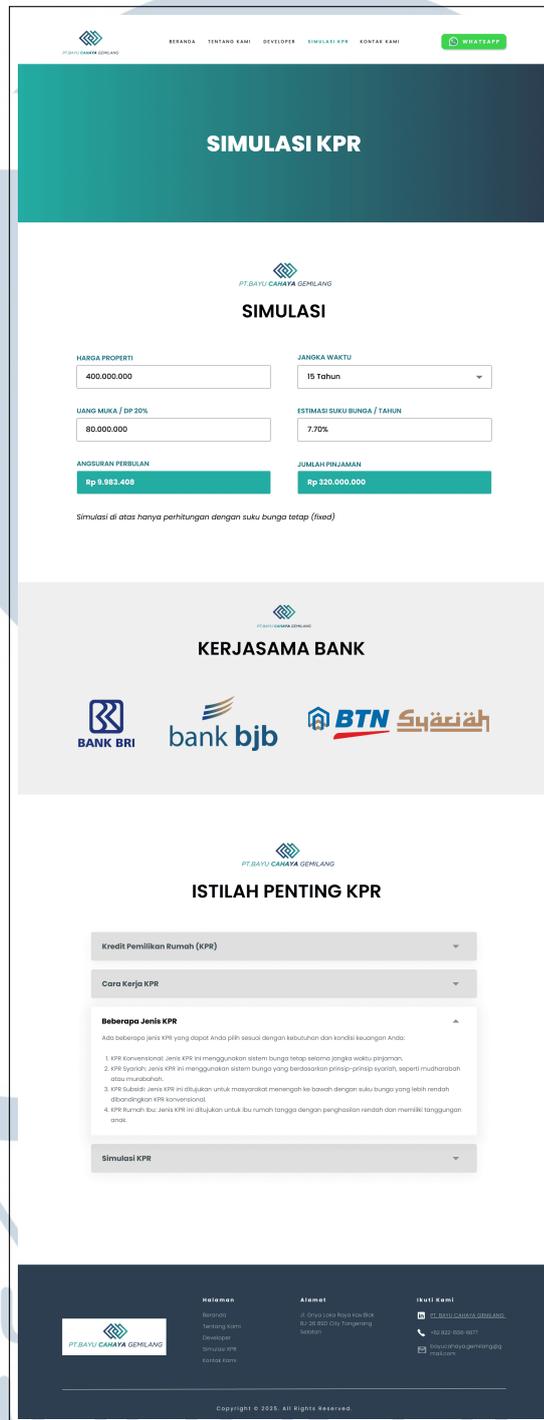
Gambar 3.26. Desain sesi Deskripsi pada halaman Cluster



Gambar 3.27. Desain sesi Spesifikasi pada halaman Cluster

Gambar 3.26 dan Gambar 3.27 menampilkan sesi konten tekstual dan teknis yang dibagi dalam dua bagian utama. Bagian "Deskripsi" memuat narasi tentang lingkungan dan keunggulan proyek hunian, sementara bagian "Spesifikasi" merinci fitur tampilan *floor plan*, serta teknis bangunan seperti pondasi, atap, dinding, dan fasilitas lainnya yang ditampilkan secara visual menggunakan ikon. Pendekatan ini mendukung prinsip *strive for consistency* karena penggunaan *layout*, ikon, dan tipografi dibuat seragam dengan halaman lain. Penggunaan tab juga memperkuat prinsip *reduce short-term memory load* karena pengguna hanya melihat satu jenis informasi pada satu waktu.

3.3.5.2.5 Halaman Simulasi KPR

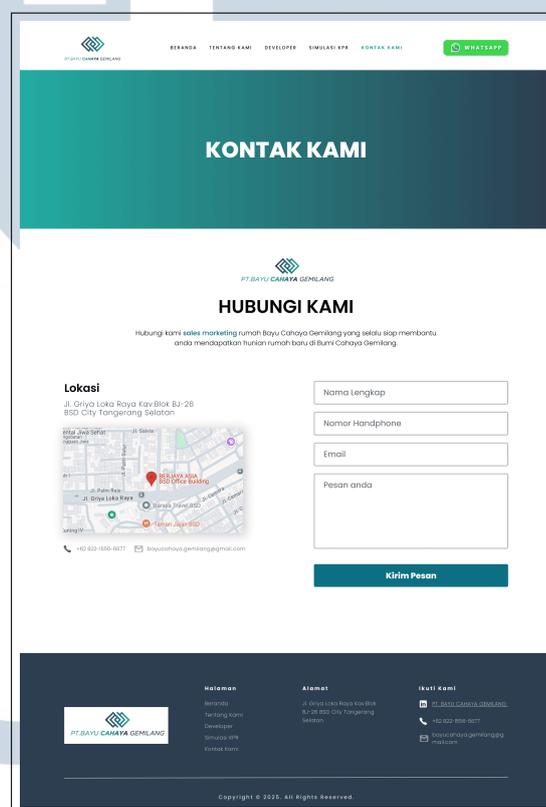


Gambar 3.28. Desain halaman Simulasi KPR pada website Company Profile

Gambar 3.28 memperlihatkan desain lengkap dari halaman Simulasi KPR yang terdiri dari tiga sesi utama: Simulasi, Kerjasama Bank, dan Istilah

Penting KPR. Halaman ini dirancang untuk memberikan informasi interaktif sekaligus edukatif bagi pengunjung *website* terkait perhitungan cicilan, informasi bank rekanan, serta pemahaman istilah KPR. Penataan visual setiap sesi dibuat terpisah dengan *padding* yang cukup agar konten mudah dibaca dan tidak saling bertabrakan. Penggunaan warna *teal* untuk elemen penting seperti hasil simulasi dan tombol memperkuat prinsip *offer informative feedback*, di mana pengguna langsung mengetahui hasil dari interaksi mereka.

3.3.5.2.6 Halaman Kontak Kami



Gambar 3.29. Desain halaman Kontak Kami pada website Company Profile

Gambar 3.29 menunjukkan desain halaman Kontak Kami yang terdiri dari satu sesi utama, yaitu Hubungi Kami. Desain halaman ini menempatkan bagian peta lokasi-lokasi cluster yang ada, informasi kontak, dan formulir input secara berdampingan dalam dua kolom yang seimbang. Tujuannya adalah agar pengunjung dapat langsung menemukan informasi penting sekaligus melakukan interaksi dalam satu tampilan layar. Warna tombol yang kontras serta *form* yang bersih mendukung keterbacaan dan aksesibilitas pengguna.

3.3.6 Implementasi Sistem

Bagian ini menjelaskan proses implementasi sistem yang telah dirancang pada tahap sebelumnya. Implementasi dilakukan berdasarkan desain antarmuka yang telah disusun menggunakan *Figma*, serta spesifikasi sistem yang dirancang selama kegiatan magang. Proses ini mencakup integrasi antar komponen *frontend* dan *backend*, pengujian fitur utama, serta penyesuaian teknis agar seluruh sistem berjalan sesuai kebutuhan. Fokus utama dari implementasi adalah memastikan sistem dapat berfungsi secara *end-to-end*, mendukung manajemen konten dinamis melalui *CMS Admin*, serta menampilkan informasi secara optimal melalui *website Company Profile*.

3.3.6.1 Implementasi Backend

Implementasi *backend* dilakukan untuk membangun sistem yang dapat menangani autentikasi, pengelolaan data, dan komunikasi antara *frontend* dengan *database* secara aman dan efisien. *Backend* dikembangkan menggunakan *Node.js* dengan *framework Express*, serta terhubung ke layanan *Supabase* sebagai database utama. Selain itu, sistem juga mengintegrasikan layanan pihak ketiga seperti *Cloudinary* untuk penyimpanan gambar. Pada sub-bagian ini akan dibahas dua fitur utama dari implementasi *backend*, yaitu sistem autentikasi berbasis *JWT* dan integrasi *Cloudinary* sebagai solusi penyimpanan media.

3.3.6.1.1 Authentication System

Untuk mengamankan akses pada sistem *CMS Admin*, digunakan metode autentikasi berbasis *JSON Web Token (JWT)*. *JWT* adalah standar terbuka (RFC 7519) yang memungkinkan pertukaran informasi secara aman antara dua pihak sebagai objek *JSON*. Dalam konteks ini, *JWT* digunakan untuk mengidentifikasi admin yang telah berhasil *login*, dan memberikan hak akses terhadap berbagai *endpoint* yang bersifat sensitif. *JWT* dipilih karena bersifat *stateless*, ringan, serta cocok untuk digunakan dalam arsitektur modern berbasis *REST API*.

3.3.6.1.1.1 Autentikasi Login dan Pembuatan Token

Proses autentikasi dimulai dari pembuatan *API login* pada *server.js*, seperti terlihat pada potongan kode 3.1. Ketika pengguna mengirimkan *username* dan

password, sistem akan mencocokkan data dengan yang tersimpan di tabel *admins* pada Supabase. Apabila valid, sistem akan membangkitkan token menggunakan *jwt.sign()* dan mengembalikannya ke sisi *client*. Token ini menyimpan data *id* dari admin dan memiliki waktu kedaluwarsa selama 3 jam.

```
1 app.post("/api/login", async (req, res) => {
2   const { username, password } = req.body;
3
4   const { data: admin, error } = await supabase
5     .from("admins")
6     .select("*")
7     .eq("username", username)
8     .single();
9
10  if (error || !admin) {
11    return res.status(401).json({ error: "Invalid credentials" });
12  }
13
14  const isMatch = await bcrypt.compare(password, admin.password);
15  if (!isMatch) return res.status(401).json({ error: "Invalid
16    credentials" });
17
18  const token = jwt.sign({ id: admin.id }, process.env.JWT_SECRET,
19    {
20      expiresIn: "3h",
21    });
22  res.json({ token });
23 }
```

Kode 3.1: Kode API untuk autentikasi login admin dan pembuatan JWT token

3.3.6.1.1.2 Struktur Endpoint dan Penerapan Middleware

Setelah proses *login* berhasil dan token diterima oleh *client*, token tersebut digunakan sebagai otorisasi untuk mengakses berbagai *endpoint*. Semua *route* pada *backend* dikelompokkan dan disusun secara modular, seperti */api/home*, */api/about*, */api/cluster*, dan sebagainya. Untuk memastikan bahwa hanya pengguna yang sudah *login* yang dapat mengakses *endpoint* tertentu, *middleware verifyToken* diterapkan pada setiap *route* yang bersifat *protected*.

Contoh kode yang telah diterapkan dapat dilihat pada Kode 3.2.

```

1 app.use("/api/home", homeRoutes);
2 app.use("/api/about", aboutRoutes);
3 app.use("/api/about-info", aboutInfoRoutes);
4 ...

```

Kode 3.2: Struktur endpoint backend untuk setiap resource API pada sistem

3.3.6.1.1.3 Middleware Verifikasi Token

Middleware otorisasi dituliskan dalam berkas *auth.js*. Fungsi *verifyToken* mengambil token dari header *Authorization*, memverifikasi keasliannya menggunakan *JWT_SECRET*, dan menyisipkan *adminId* ke objek *request*. Dengan pendekatan ini, *backend* dapat mengetahui identitas admin yang sedang mengakses dan membatasi akses apabila token tidak valid atau tidak tersedia.

Contoh kode yang telah diterapkan dapat dilihat pada Kode 3.3.

```

1 import jwt from "jsonwebtoken";
2
3 export const verifyToken = (req, res, next) => {
4   const authHeader = req.headers["authorization"];
5   const token =
6     authHeader && authHeader.startsWith("Bearer ")
7     ? authHeader.split(" ")[1]
8     : authHeader;
9
10  if (!token) return res.status(403).json({ error: "No token
11    provided" });
12
13  jwt.verify(token, process.env.JWT_SECRET, (err, decoded) => {
14    if (err) return res.status(401).json({ error: "Unauthorized"
15    });
16    req.adminId = decoded.id;
17    next();
18  });
19 }

```

Kode 3.3: Kode middleware *verifyToken* untuk memverifikasi JWT pada setiap permintaan

3.3.6.1.1.4 Penerapan Middleware pada Endpoint Backend

Contoh penerapan *verifyToken* dapat dilihat pada salah satu *endpoint* untuk menambahkan konten baru. Di sini, token diverifikasi terlebih dahulu

sebelum admin diizinkan mengunggah gambar dan menambahkan data ke tabel *home_sections*. Ini menjadi salah satu langkah untuk mencegah manipulasi data oleh pengguna yang tidak memiliki akses sah.

Contoh kode yang telah diterapkan dapat dilihat pada Kode 3.4.

```
1 router.post("/", verifyToken, upload.single("image"), async (req,
  res) => {
2   try {
3     const { title, paragraph } = req.body;
4     const imageUrl = req.file ? req.file.path : null;
5
6     const { data, error } = await supabase.from("home_sections").
insert([
7     {
8       title,
9       paragraph: paragraph || null,
10      image: imageUrl,
11    },
12  ]);
13
14   if (error) return res.status(500).json({ error: error.message
});
15   res.json({ message: "Content added successfully!", data });
16 } catch (err) {
17   res.status(500).json({ error: err.message });
18 }
19 });
```

Kode 3.4: Contoh penerapan *verifyToken* pada route *POST* untuk menambah konten halaman Beranda

3.3.6.1.1.5 Validasi Token di Sisi Klien

Di sisi klien, token JWT yang diterima saat login disimpan di *localStorage* dan digunakan untuk mengatur akses halaman. Fungsi *isTokenValid()* melakukan pengecekan apakah token masih berlaku berdasarkan nilai *exp*. Apabila token valid, pengguna diizinkan mengakses halaman *ProtectedRoute*, jika tidak, mereka akan diarahkan kembali ke halaman login. Pendekatan ini menjamin bahwa pengguna yang belum terautentikasi tidak dapat melihat atau mengubah data di dalam *CMS Admin*.

Contoh kode yang telah diterapkan dapat dilihat pada Kode 3.5.

```
1 import { jwtDecode } from "jwt-decode";
```

```

2
3 function isValidToken(token) {
4   if (!token) return false;
5   try {
6     const { exp } = jwtDecode(token);
7     return exp * 1000 > Date.now();
8   } catch {
9     return false;
10  }
11 }
12
13 function ProtectedRoute({ children }) {
14   const token = localStorage.getItem("token");
15   return isValidToken(token) ? children : <Navigate to="/" />;
16 }

```

Kode 3.5: Pengecekan validitas token JWT pada sisi klien untuk melindungi route privat

Dengan penggunaan JWT sebagai sistem autentikasi, keamanan akses terhadap sistem *CMS Admin* menjadi lebih terjaga karena hanya pengguna yang memiliki token valid yang dapat mengakses dan memodifikasi konten. Selain itu, implementasi ini juga mendukung arsitektur *frontend-backend* terpisah yang efisien dan mudah untuk dikembangkan lebih lanjut.

3.3.6.1.2 Cloudinary (Cloud Image Storage)

Pada pengembangan *CMS Admin*, gambar-gambar seperti *preview cluster*, logo bank, ikon spesifikasi, serta konten halaman lainnya perlu disimpan secara terpusat dan efisien. Untuk itu, digunakan layanan *Cloudinary* sebagai solusi penyimpanan gambar berbasis *cloud*. *Cloudinary* memungkinkan pengelolaan gambar dengan performa tinggi, dukungan *CDN*, dan *API* fleksibel untuk proses unggah, penghapusan, serta manajemen *metadata*. Penggunaan penyimpanan berbasis *cloud* juga bertujuan untuk menghindari penyimpanan langsung pada *server* utama demi efisiensi dan keamanan.

3.3.6.1.2.1 Konfigurasi Autentikasi Cloudinary

Integrasi *Cloudinary* dilakukan melalui *cloudinary* dan *multer-storage-cloudinary* yang dikonfigurasi pada sisi *backend*. Gambar yang diunggah oleh admin melalui antarmuka *CMS* akan otomatis dikirim ke folder *cms-images* di akun

Cloudinary, dan *URL* dari gambar tersebut akan disimpan di *database Supabase*. Seluruh proses ini dilakukan melalui alur yang aman dan terstandarisasi. Berikut adalah konfigurasi dasar autentikasi *Cloudinary*:

```
1 import cloudinary from "cloudinary";
2
3 cloudinary.v2.config({
4   cloud_name: process.env.CLOUD_NAME,
5   api_key: process.env.CLOUD_API_KEY,
6   api_secret: process.env.CLOUD_API_SECRET,
7 });
```

Kode 3.6: Konfigurasi kredensial Cloudinary menggunakan *dotenv*

3.3.6.1.2.2 Konfigurasi Penyimpanan Gambar

Selanjutnya, sistem menggunakan konfigurasi *CloudinaryStorage* dari *multer-storage-cloudinary* agar gambar secara otomatis tersimpan di folder yang ditentukan. *Public ID* dari gambar akan diset berdasarkan waktu unggah agar unik.

```
1 import { CloudinaryStorage } from "multer-storage-cloudinary";
2
3 const storage = new CloudinaryStorage({
4   cloudinary: cloudinary.v2,
5   params: {
6     folder: "cms-images",
7     format: async () => "png",
8     public_id: () => Date.now(),
9   },
10 });
```

Kode 3.7: Konfigurasi penyimpanan gambar Cloudinary menggunakan *multer-storage-cloudinary*

3.3.6.1.2.3 Pengunggahan Gambar saat Menambah Konten

Contoh implementasi pengunggahan gambar terdapat pada endpoint *POST* untuk menambah konten halaman Beranda. Gambar yang diterima dari klien langsung diunggah ke *Cloudinary*, dan *URL*-nya disimpan ke *database*.

```
1 router.post("/", verifyToken, upload.single("image"), async (req,
2   res) => {
3   const { title, paragraph } = req.body;
```

```

3  const imageUrl = req.file ? req.file.path : null;
4
5  const { data, error } = await supabase.from("home_sections").
    insert([
6    { title, paragraph, image: imageUrl },
7  ]);
8
9  ...
10 });

```

Kode 3.8: Contoh *POST* untuk menambah konten halaman Beranda

3.3.6.1.2.4 Penghapusan Gambar Lama saat Pembaruan Konten

Saat admin memperbarui konten dengan gambar baru, gambar lama yang tersimpan di *Cloudinary* akan dihapus terlebih dahulu untuk menjaga efisiensi ruang penyimpanan. Proses ini dilakukan dengan mengekstrak *public_id* dari *URL* gambar lama.

```

1  router.put("/:id", verifyToken, upload.single("image"), async (req
    , res) => {
2    try {
3      ...
4
5      if (req.file && oldImageUrl) {
6        let publicId = oldImageUrl.split("/").slice(-2).join("/").
split(".")[0];
7        cloudinary.v2.uploader.destroy(publicId, (error) => {
8          if (error)
9            console.log("Gagal menghapus gambar lama dari Cloudinary
: ", error);
10         else console.log("Gambar lama berhasil dihapus dari
Cloudinary");
11       });
12     }
13
14     ...
15   }
16 });

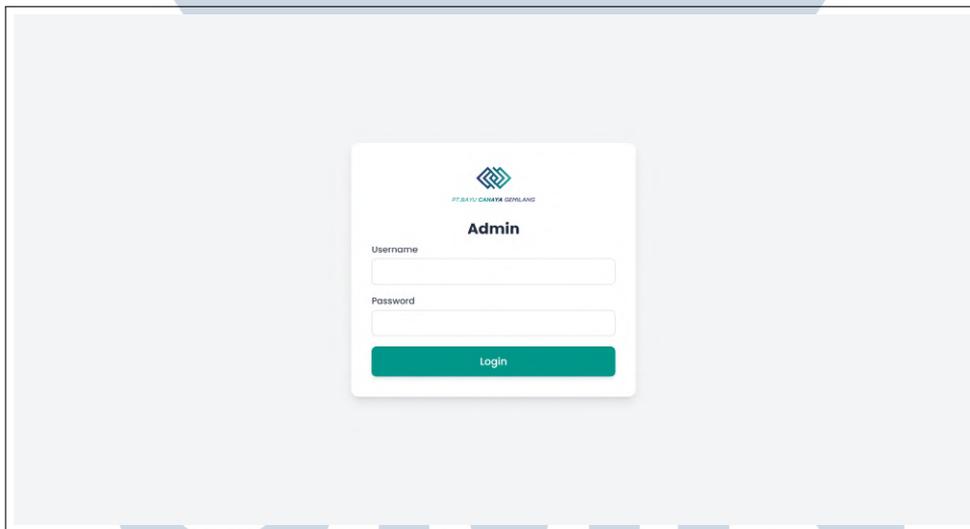
```

Kode 3.9: Kode untuk menghapus gambar lama saat memperbarui konten

3.3.6.2 Implementasi Frontend CMS Admin

Frontend CMS Admin dirancang untuk memungkinkan admin mengelola konten *website* secara mudah, dinamis, dan aman. Implementasi dilakukan menggunakan *React.js* dan *Tailwind CSS* untuk menghasilkan antarmuka yang konsisten, responsif, dan mudah dipahami. Sistem ini mendukung fungsi *CRUD* terhadap berbagai data seperti informasi perusahaan, *cluster*, dan simulasi KPR. Sub-bagian berikut akan menampilkan hasil implementasi *website* dari desain antarmuka *CMS Admin* yang sudah dibuat sebelumnya. Untuk tampilan *website* tidak semuanya ditampilkan namun fitur-fitur yang penting dan memiliki karakteristik tersendiri yang akan ditampilkan di sub-bagian ini.

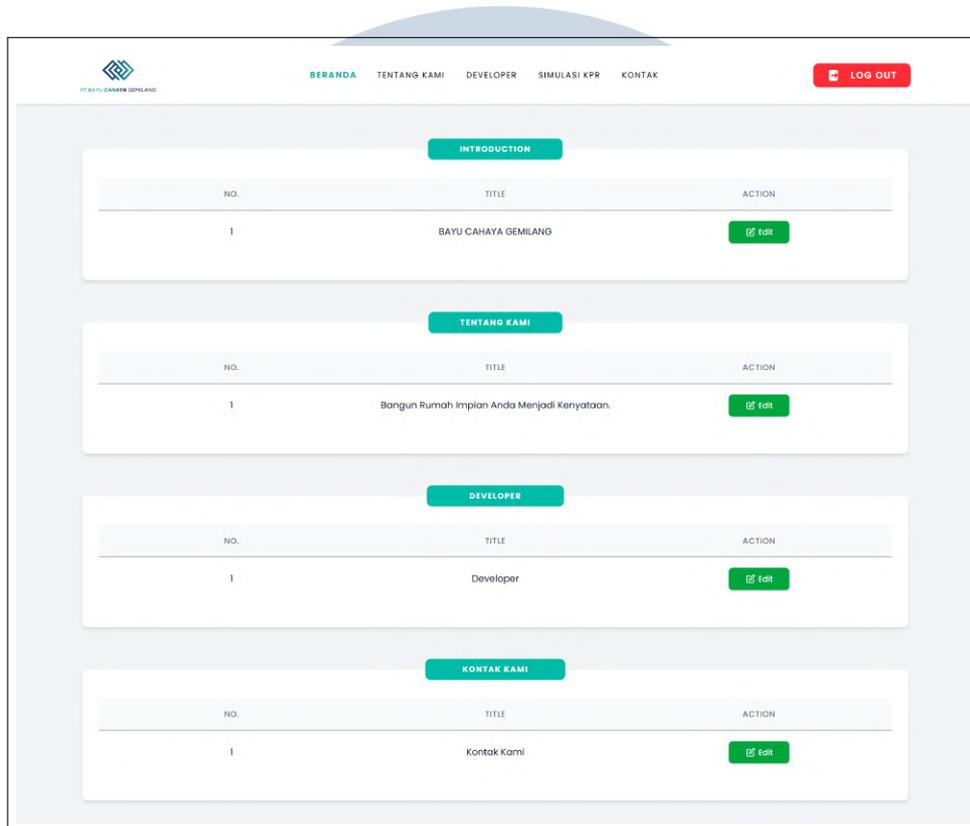
3.3.6.2.1 Halaman Login



Gambar 3.30. Tampilan login pada website CMS Admin

Gambar 3.30 merupakan tampilan awal halaman login pada website *CMS Admin*. Halaman ini terdiri dari dua komponen utama yang harus diisi oleh pengguna, yaitu *username* dan *password*. Setelah memasukkan data yang valid, pengguna akan diarahkan menuju halaman beranda. Antarmuka ini dirancang sederhana mungkin untuk mempermudah proses autentikasi pengguna.

3.3.6.2.2 Halaman Beranda

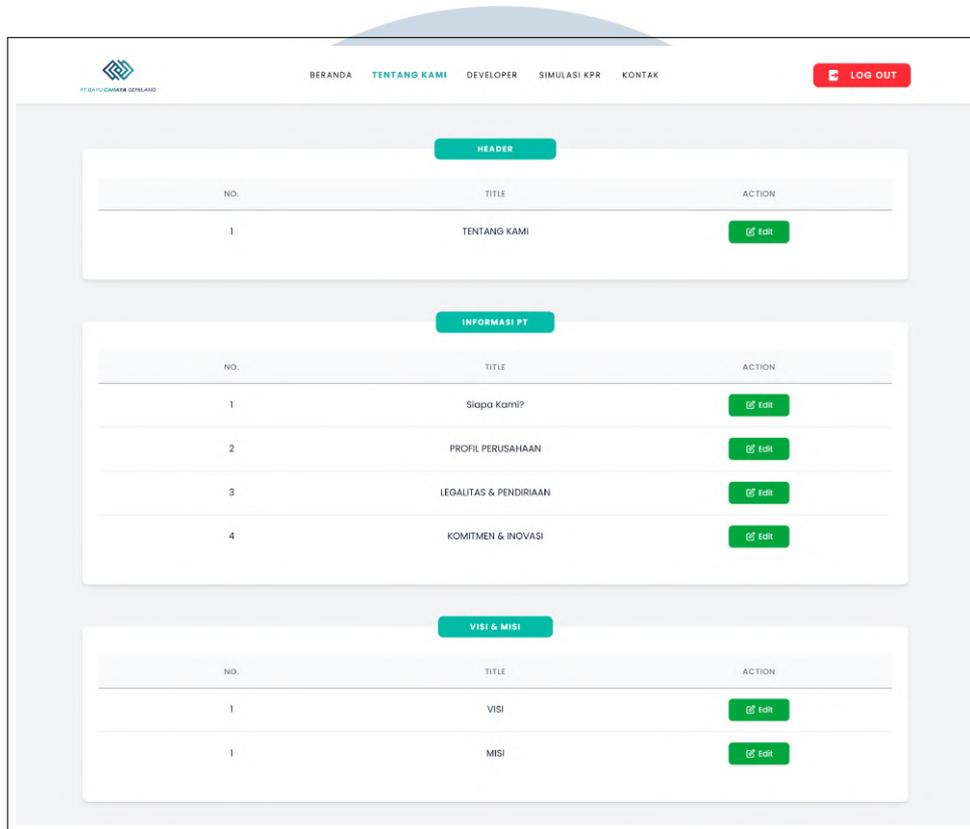


Gambar 3.31. Tampilan Beranda pada website CMS Admin

Gambar 3.31 merupakan halaman Beranda pada *CMS Admin* yang menampilkan daftar empat *section* utama yang ditampilkan di *website Company Profile*, yaitu Introduction, Tentang Kami, Kontak Kami, serta Project dan Simulasi KPR. Masing-masing *section* ditampilkan dalam bentuk tabel yang dilengkapi informasi konten dan tombol *Edit* untuk memudahkan admin memperbarui *title*, *paragraph*, dan *image*. Antarmuka yang sederhana dan terstruktur memungkinkan admin untuk langsung menavigasi dan mengelola konten dengan cepat dan efisien.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.3.6.2.3 Halaman Tentang Kami

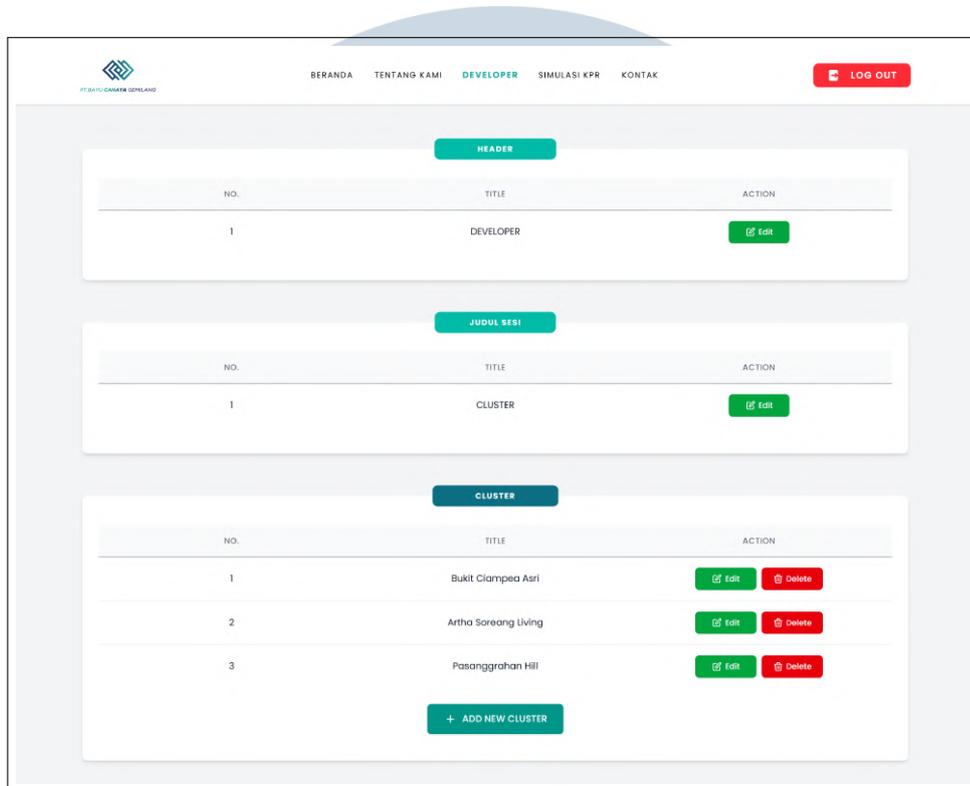


Gambar 3.32. Tampilan halaman Tentang Kami pada website CMS Admin

Halaman Tentang Kami terdiri dari tiga sesi utama, yaitu Header, Informasi Perusahaan, serta Visi dan Misi, yang seluruhnya dapat dikelola melalui *CMS Admin*. Gambar 3.32 menampilkan tabel daftar konten dari masing-masing sesi, lengkap dengan informasi jumlah konten, judul, dan tombol *Edit* pada tiap barisnya. Struktur ini memudahkan admin untuk memilih bagian yang ingin diperbarui tanpa harus membuka keseluruhan halaman secara langsung.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.3.6.2.4 Halaman Developer

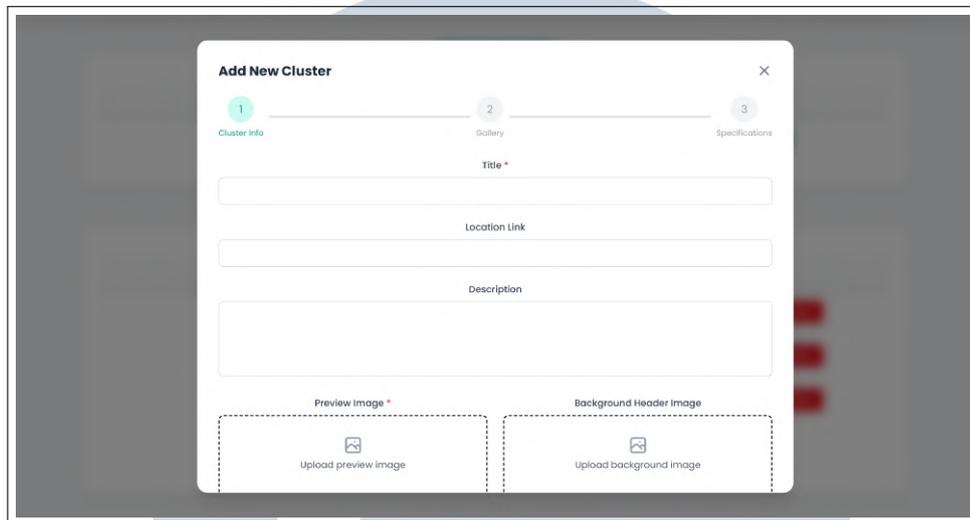


Gambar 3.33. Tampilan halaman Developer pada website CMS Admin

Gambar 3.33 menampilkan halaman Developer pada *CMS Admin* yang terdiri dari dua *section* utama, yaitu Header dan Cluster. Struktur halaman ini mirip dengan halaman lainnya, menampilkan tabel dengan informasi konten, judul, serta tombol *Edit*. Khusus untuk *section* Cluster, tersedia tambahan fitur untuk menambahkan dan menghapus cluster menggunakan tombol *Add New Cluster* dan *Delete*. Setiap baris cluster yang ditampilkan dapat diakses lebih lanjut untuk pengelolaan detail konten masing-masing cluster pada halaman terpisah.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.3.6.2.4.1 Fitur Add New Cluster (CREATE)

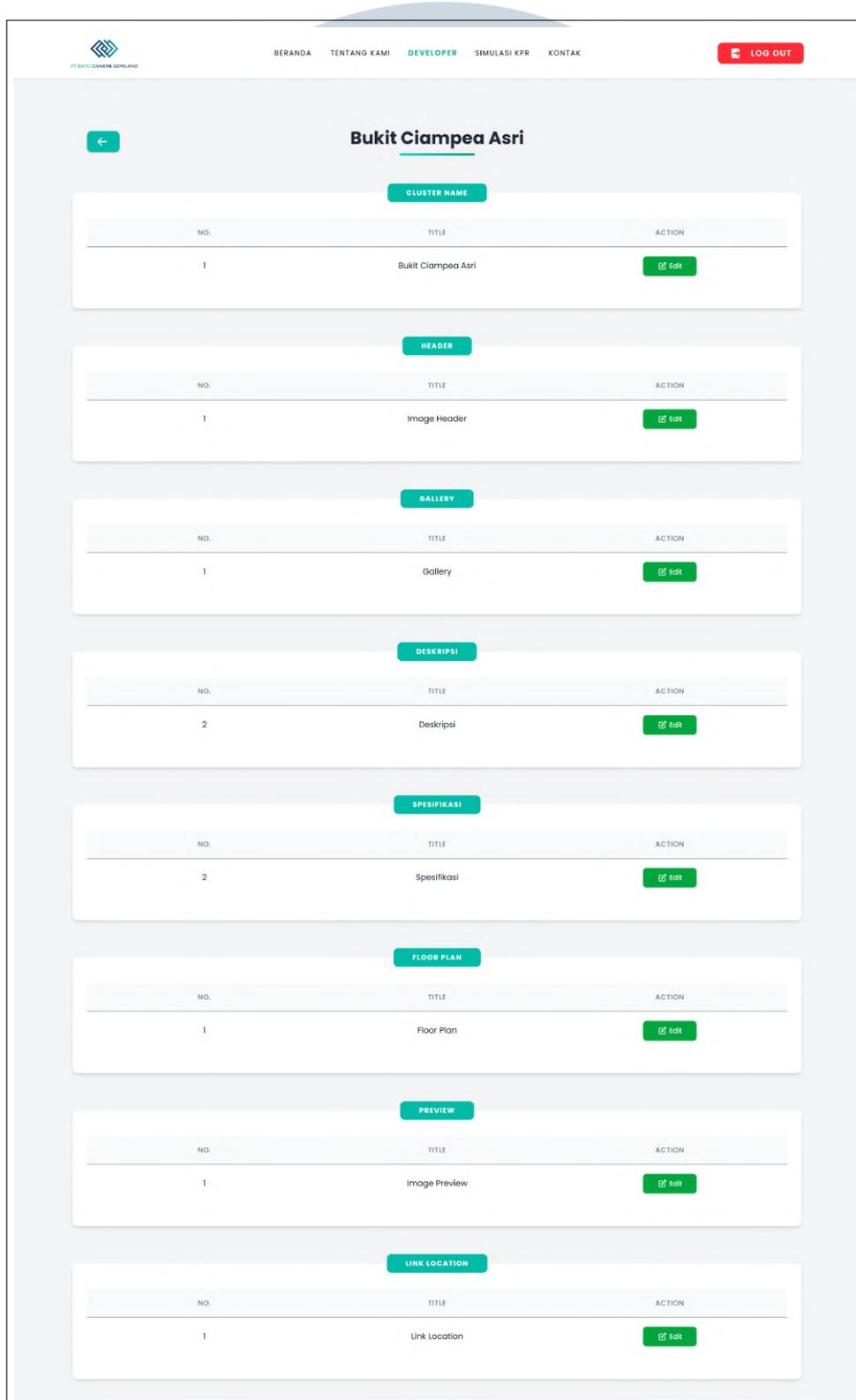
The image shows a web form titled "Add New Cluster" with a close button (X) in the top right corner. The form is divided into three steps: 1. Cluster Info (highlighted in green), 2. Gallery, and 3. Specifications. Under step 1, there are three text input fields labeled "Title", "Location Link", and "Description". Below these are two image upload sections. The first is labeled "Preview Image" and the second is "Background Header Image". Each section contains a dashed box with a camera icon and the text "Upload [type] image".

Gambar 3.34. Tampilan fitur Add New Cluster step pertama

Fitur *Add New Cluster* pada halaman Developer memungkinkan admin menambahkan data cluster baru ke dalam sistem. Langkah pertama (Gambar 3.34) meminta admin untuk mengisi informasi utama seperti *title*, *location link*, *description*, serta mengunggah beberapa gambar, yaitu *preview image*, *background header*, *logo header* dan *floor plan image*. Setelah data berhasil disimpan, sistem akan mengupdate tabel *cluster*. Langkah kedua memungkinkan admin untuk menambahkan beberapa gambar galeri untuk cluster tersebut. Gambar yang diunggah akan langsung disimpan ke dalam tabel *cluster_gallery*. Langkah ketiga adalah input item spesifikasi, yang terdiri dari *title* dan pilihan *icon*. Admin dapat menambah satu atau lebih item spesifikasi. Setelah semua langkah diselesaikan dan tombol *Submit* ditekan, sistem akan menyimpan data ke dalam tabel *cluster_spesifikasi* dan proses pembuatan cluster baru dinyatakan selesai.

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

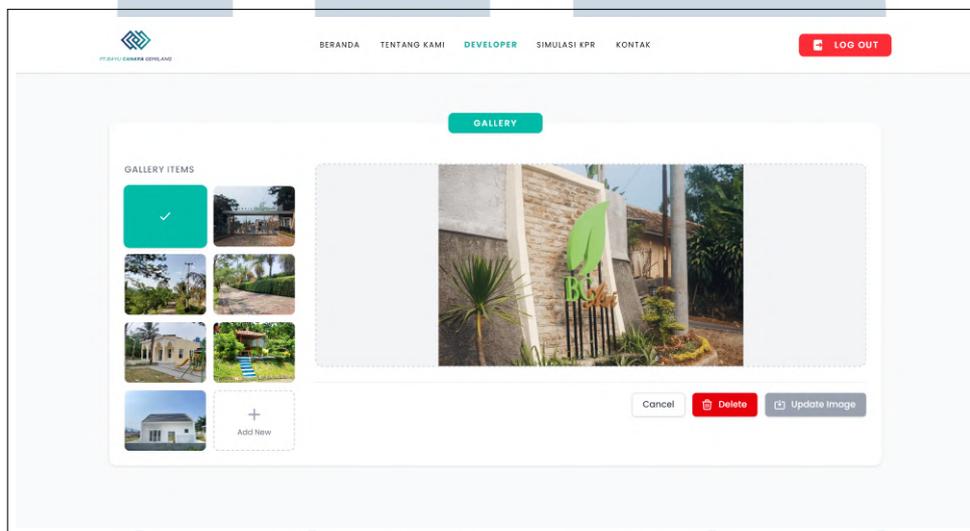
3.3.6.2.5 Halaman Cluster



Gambar 3.35. Tampilan halaman Cluster pada website CMS Admin

Gambar 3.35 menampilkan tampilan utama halaman Cluster pada *CMS Admin* yang terdiri dari delapan sesi manajemen data. Setiap sesi dikhususkan untuk pengelolaan konten terkait informasi cluster, mulai dari nama cluster, gambar, deskripsi, spesifikasi, hingga lokasi. Struktur ini mempermudah admin dalam memisahkan pengelolaan konten berdasarkan jenisnya, serta memastikan setiap bagian dapat diperbarui secara spesifik dan fleksibel.

3.3.6.2.5.1 Sesi Gallery

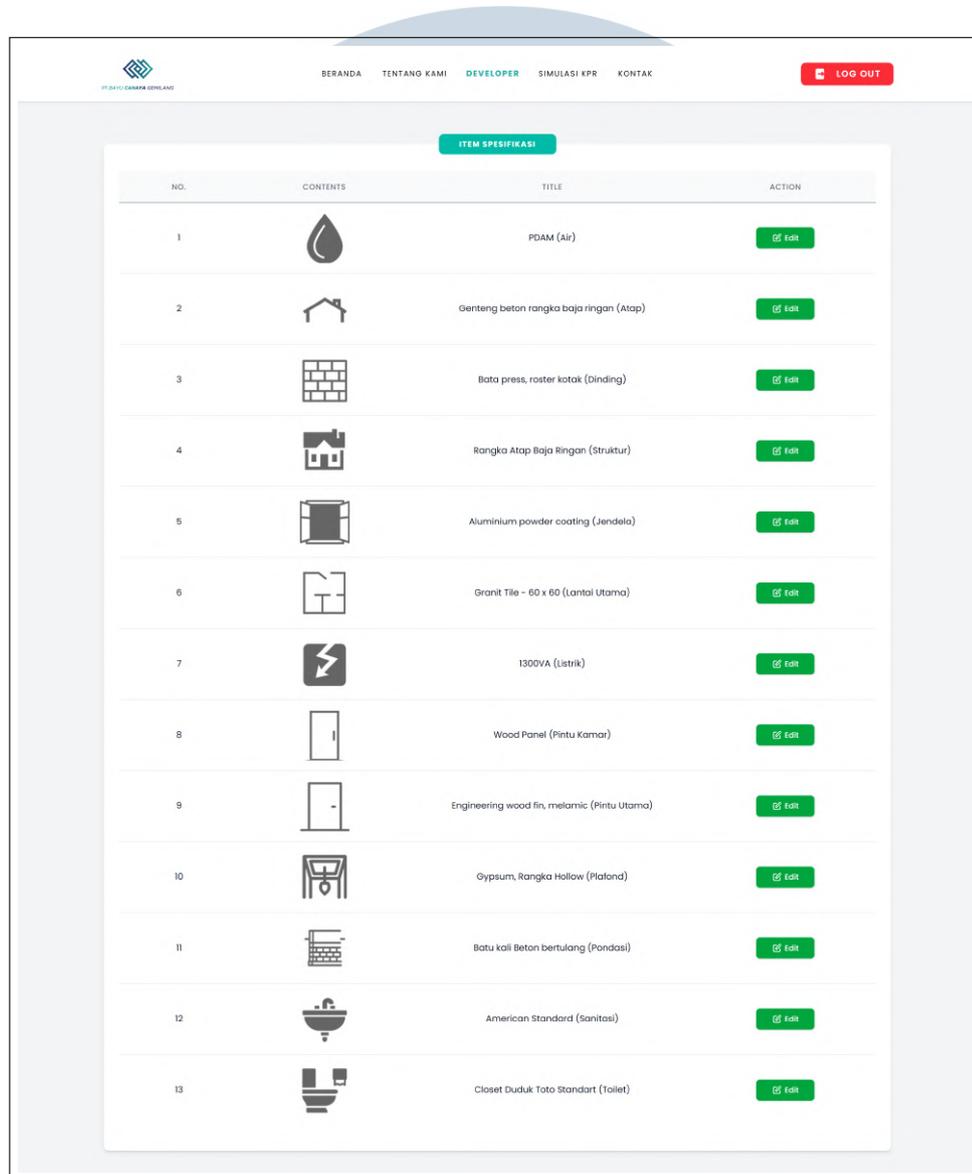


Gambar 3.36. Tampilan sesi Gallery pada halaman Cluster

Gambar 3.36 menampilkan fitur CRUD (Create, Read, Update, Delete) untuk galeri gambar cluster. Admin dapat menambahkan beberapa gambar untuk ditampilkan pada galeri setiap cluster, serta menghapus gambar yang tidak diperlukan. Sistem juga telah dilengkapi validasi yang mencegah penghapusan gambar terakhir agar galeri tidak kosong.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

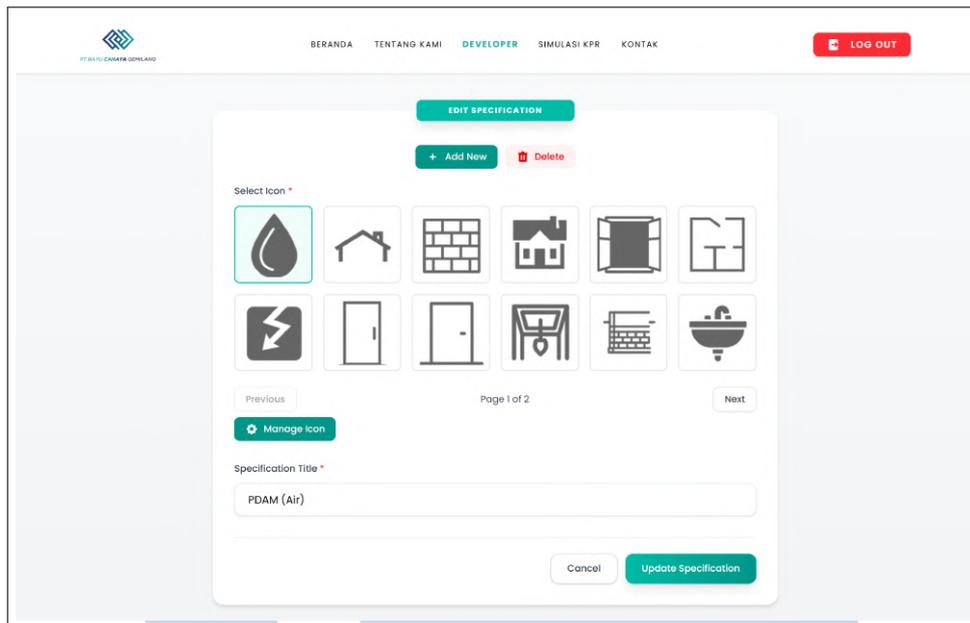
3.3.6.2.5.2 Sesi Spesifikasi



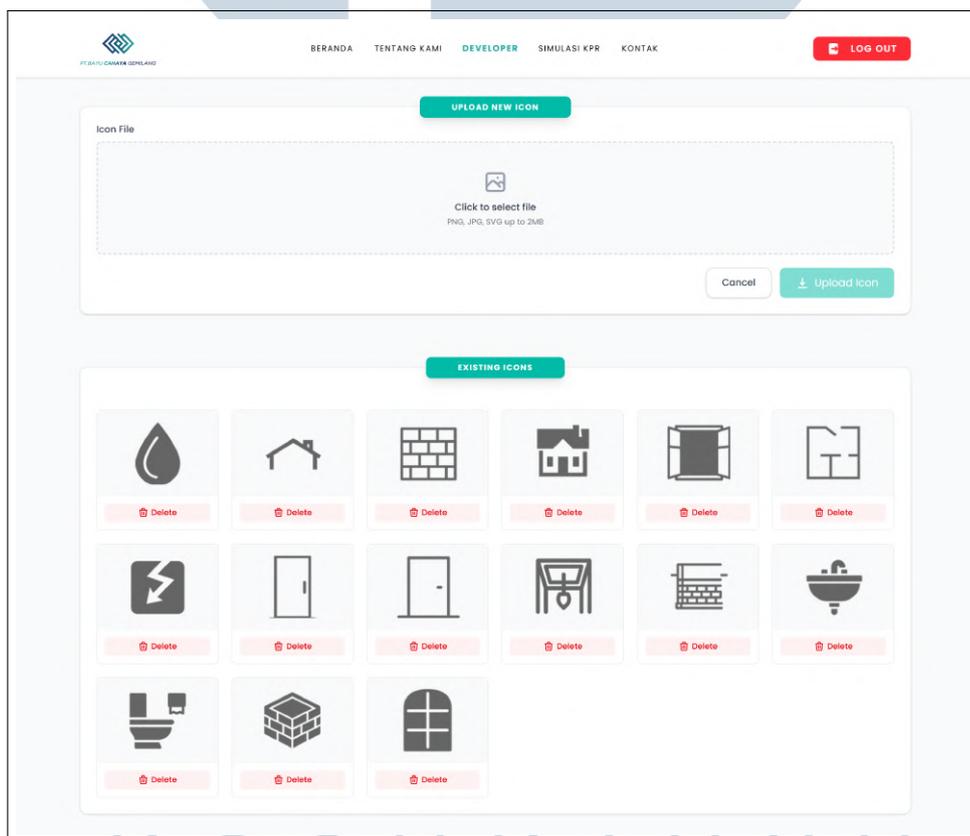
NO.	CONTENTS	TITLE	ACTION
1		PDAM (Air)	Edit
2		Genteng beton rangka baja ringan (Atap)	Edit
3		Bata press, roster kotak (Dinding)	Edit
4		Rangka Atap Baja Ringan (Struktur)	Edit
5		Aluminium powder coating (Jendela)	Edit
6		Granit Tile - 60 x 60 (Lantai Utama)	Edit
7		1300VA (Listrik)	Edit
8		Wood Panel (Pintu Kamar)	Edit
9		Engineering wood fin, melamic (Pintu Utama)	Edit
10		Gypsum, Rangka Hollow (Plafond)	Edit
11		Batu kali Beton bertulang (Pondasi)	Edit
12		American Standard (Sanitasi)	Edit
13		Closet Duduk Toto Standart (Toilet)	Edit

Gambar 3.37. Tampilan sesi Spesifikasi pada halaman Cluster

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.38. Tampilan sesi Spesifikasi Detail pada halaman Cluster



Gambar 3.39. Tampilan sesi Manajemen Ikon pada halaman Cluster

Gambar 3.37 menampilkan daftar item spesifikasi dari cluster tertentu. Admin dapat menambah, memperbarui, dan menghapus item spesifikasi seperti *title*

dan *icon* secara dinamis. Gambar 3.38 menunjukkan form detail untuk pengelolaan spesifikasi dengan fitur tambahan seperti *manage icon*, yang memungkinkan admin memilih ikon dari daftar atau menambahkan ikon baru jika belum tersedia. Gambar 3.39 memperlihatkan halaman manajemen ikon, tempat admin dapat mengunggah ikon baru atau menghapus ikon yang sudah tidak digunakan. Data spesifikasi ini penting untuk menyampaikan keunggulan fitur atau fasilitas dari tiap cluster kepada calon pembeli.

3.3.6.2.6 Halaman Simulasi KPR

HEADER		
NO.	TITLE	ACTION
1	SIMULASI KPR	Edit

SIMULASI		
NO.	TITLE	ACTION
1	SIMULASI	Edit

KERJASAMA BANK		
NO.	TITLE	ACTION
1	KERJASAMA BANK	Edit
2	LOGO BANK	Edit

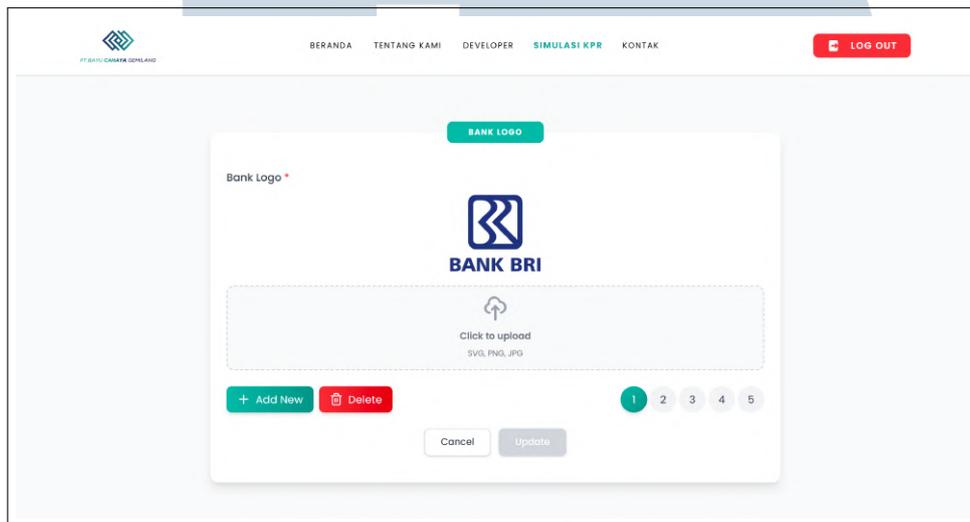
ISTILAH PENTING KPR		
NO.	TITLE	ACTION
1	ISTILAH PENTING KPR	Edit
2	Kredit Pemilikan Rumah (KPR)	Edit
3	Cara Kerja KPR	Edit
4	Beberapa Jenis KPR	Edit
5	Simulasi KPR	Edit

Gambar 3.40. Tampilan halaman Simulasi KPR pada website CMS Admin

Gambar 3.40 menunjukkan tampilan awal halaman Simulasi KPR pada *CMS Admin* dalam bentuk tabel yang dibagi menjadi empat sesi utama. Setiap sesi

memiliki tombol aksi untuk melakukan pengelolaan konten sesuai kebutuhan. Sesi-sesi tersebut meliputi Header, Simulasi, Kerjasama Bank, dan Istilah Penting KPR. Dengan struktur yang jelas dan terpisah, admin dapat mengakses dan mengelola setiap bagian dengan mudah dan efisien.

3.3.6.2.6.1 Sesi Kerjasama Bank

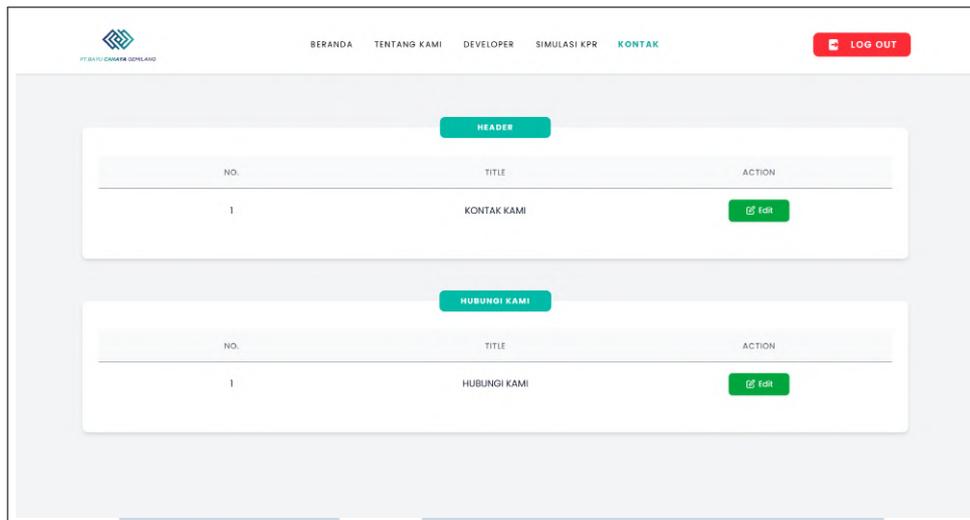


Gambar 3.41. Tampilan sesi Kerjasama Bank pada halaman Simulasi KPR

Gambar 3.41 menunjukkan sesi Kerjasama Bank yang berisi daftar logo bank yang telah bekerja sama dengan perusahaan. Setiap logo ditampilkan menggunakan sistem *pagination* untuk menghindari tampilan yang terlalu panjang. Admin dapat menambahkan, memperbarui, atau menghapus logo melalui tombol aksi yang tersedia. Fitur ini memudahkan pengelolaan mitra perbankan yang ditampilkan pada halaman simulasi.

3.3.6.2.7 Halaman Kontak Kami

Gambar 3.42 memperlihatkan tampilan utama halaman Kontak Kami pada *CMS Admin*, yang terdiri dari dua sesi utama: Header dan Hubungi Kami. Seperti halaman lainnya, sesi ditampilkan dalam bentuk tabel yang menyajikan informasi konten dan tombol *Edit* untuk memudahkan pengelolaan. Setiap bagian dapat diakses dan diperbarui secara terpisah untuk menjaga kejelasan struktur dan efisiensi manajemen konten.



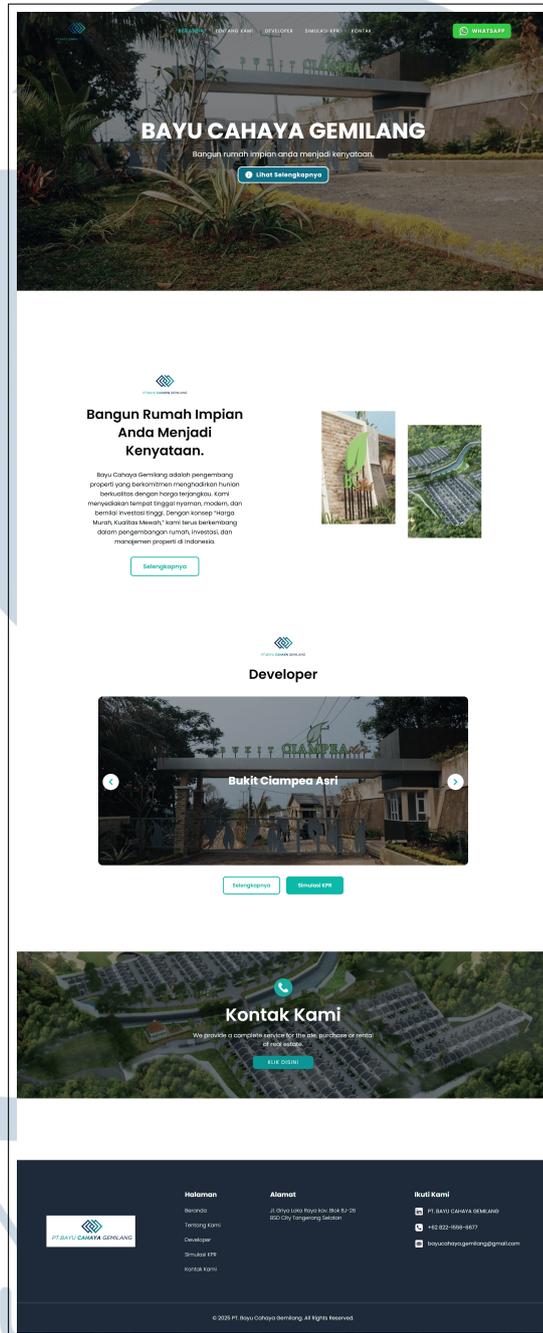
Gambar 3.42. Tampilan halaman Kontak Kami pada website CMS Admin

3.3.6.3 Implementasi Frontend Company Profile

Implementasi *frontend website Company Profile* berfokus pada penyajian informasi kepada publik dengan antarmuka yang modern dan informatif. Sistem ini dikembangkan dengan memperhatikan desain yang telah disusun sebelumnya, serta memastikan bahwa tampilan tetap optimal di berbagai perangkat. Komponen *frontend* terhubung dengan *backend* melalui proses *fetching* data agar konten yang ditampilkan selalu diperbarui secara real-time. Sub-bagian berikut akan menampilkan hasil implementasi *website* dari desain antarmuka *Company Profile* yang sudah dibuat sebelumnya dan juga hasil tampilan responsif di berbagai perangkat.

U M 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

3.3.6.3.1 Halaman Beranda



Gambar 3.43. Tampilan halaman Beranda pada website Company Profile

Gambar 3.43 menampilkan halaman utama *website Company Profile* PT Bayu Cahaya Gemilang yang terdiri dari beberapa sesi, yaitu *Introduction*, *Tentang Kami*, *Developer*, dan *Kontak Kami*. Seluruh data pada halaman ini bersifat dinamis dan diambil dari *backend* menggunakan metode *fetch* dengan *request GET*. Desain

telah diimplementasikan agar tampil optimal di berbagai ukuran layar dan perangkat secara responsif. Fungsionalitas pada setiap tombol navigasi juga diaktifkan untuk mengarahkan pengguna ke halaman atau sesi yang relevan guna meningkatkan alur interaksi pengguna.

3.3.6.3.2 Halaman Tentang Kami

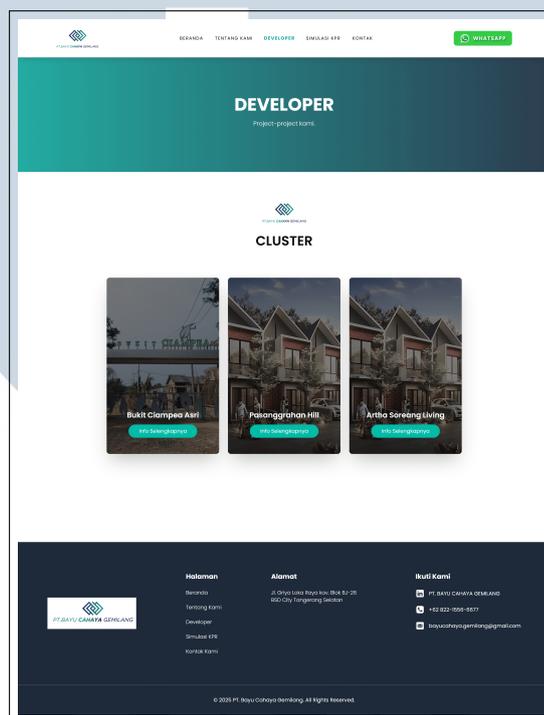


Gambar 3.44. Tampilan halaman Tentang Kami pada website Company Profile

Gambar 3.44 menampilkan halaman Tentang Kami yang disusun untuk menyampaikan konten profil perusahaan seperti informasi perusahaan, dan visi &

misi perusahaan kepada pengunjung secara lebih rinci. Seluruh konten ditampilkan berdasarkan data yang ditarik dari sistem *backend* dan bersifat *read-only*, sehingga pengguna hanya dapat melihat informasi tanpa mengubahnya. Halaman ini juga sudah dioptimalkan untuk tampil secara responsif di berbagai perangkat.

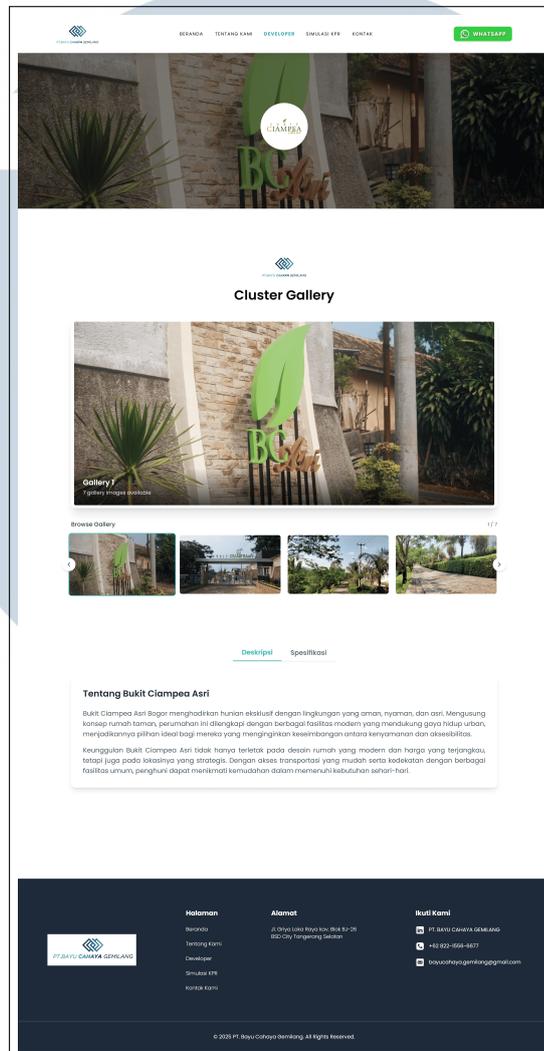
3.3.6.3.3 Halaman Developer



Gambar 3.45. Tampilan halaman Developer pada website Company Profile

Gambar 3.45 menampilkan halaman *Developer* yang dirancang untuk menampilkan daftar proyek properti yang dikembangkan oleh perusahaan. Seluruh data proyek ditampilkan secara dinamis berdasarkan data dari *backend* dan bersifat *read-only*. Halaman ini juga telah sepenuhnya responsif, sehingga konten dapat ditampilkan dengan baik di berbagai ukuran layar, mulai dari *handphone* hingga *desktop*.

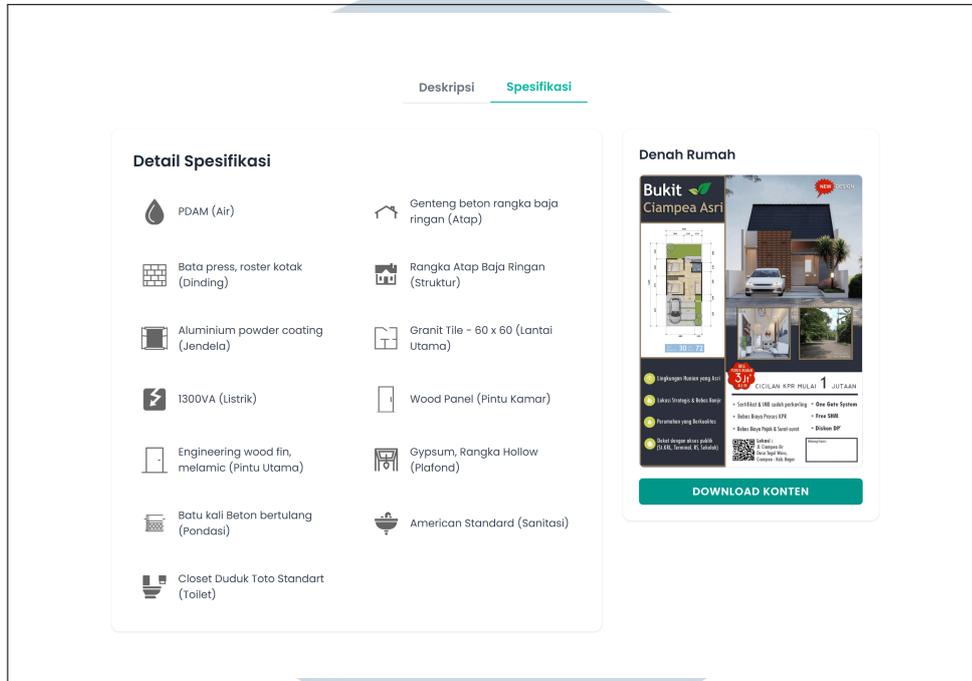
3.3.6.3.4 Halaman Cluster



Gambar 3.46. Tampilan halaman Cluster pada website Company Profile

Gambar 3.46 menampilkan halaman Cluster yang menyajikan informasi terperinci mengenai salah satu proyek perumahan milik PT Bayu Cahaya Gemilang. Secara visual, struktur halaman disusun dalam empat bagian utama yaitu *Header*, *Gallery*, *Deskripsi*, dan *Spesifikasi*. Halaman ini bersifat *read-only*, diambil melalui pemanggilan data dari *backend* dan ditampilkan secara responsif pada berbagai perangkat. Semua tampilan dan setiap sesi pada halaman ini juga sudah responsif secara keseluruhan sehingga semua pengguna dapat mengakses *website* ini dengan tampilan yang nyaman.

3.3.6.3.4.1 Sesi Spesifikasi

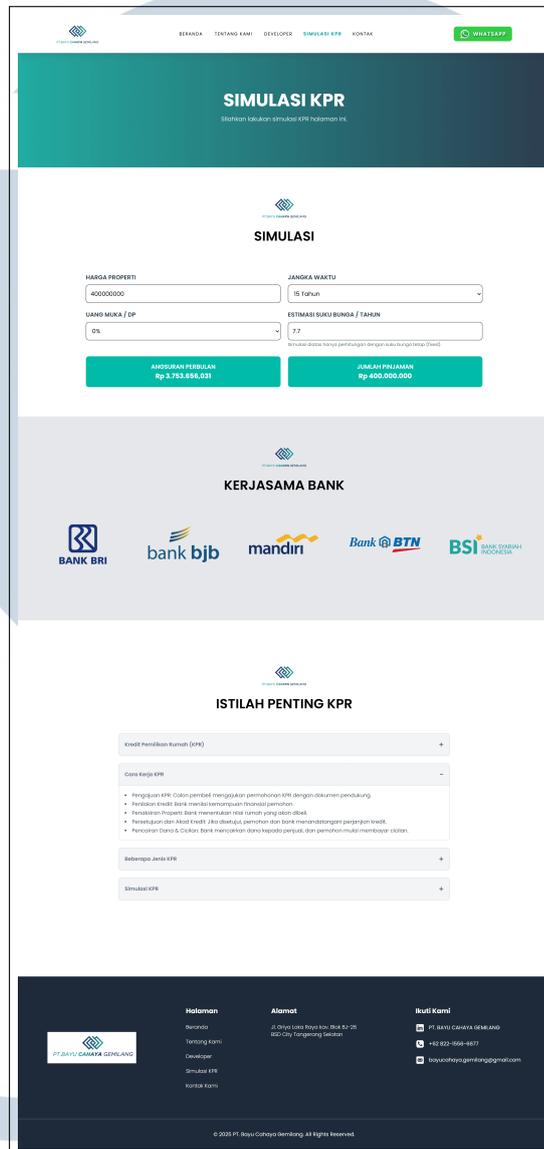


Gambar 3.47. Tampilan sesi Spesifikasi pada halaman Cluster (Company Profile)

Gambar 3.47 menunjukkan detail teknis dari rumah, seperti material yang digunakan (pondasi, struktur atap, dinding, lantai, dll), sistem kelistrikan, air, hingga sanitasi. Selain itu, pada sisi kanan tersedia gambar denah rumah (*floor plan*) beserta tombol “*Download Konten*” yang memungkinkan pengunjung untuk mengunduh brosur dalam format visual. Informasi ini ditampilkan dalam bentuk ikon dan teks untuk meningkatkan keterbacaan dan daya tarik visual. Tujuannya adalah memberikan informasi teknis yang akurat namun tetap mudah diakses oleh pengguna umum tanpa latar belakang teknis.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

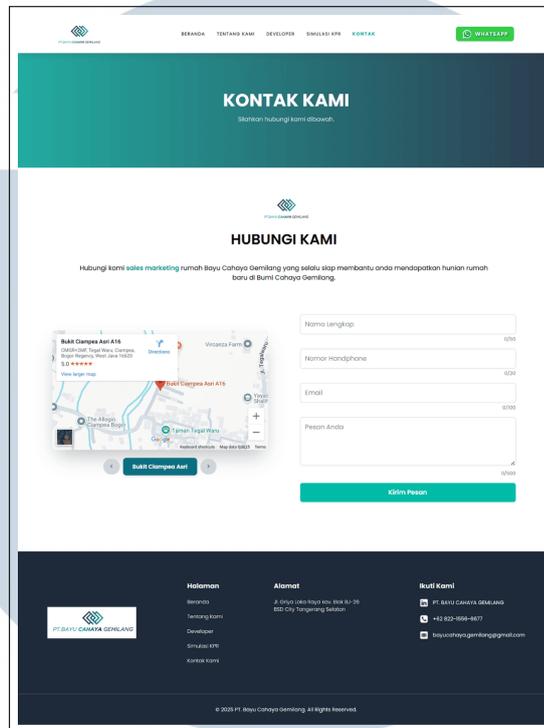
3.3.6.3.5 Halaman Simulasi KPR



Gambar 3.48. Tampilan halaman Simulasi KPR pada website Company Profile

Gambar 3.48 menampilkan tampilan akhir halaman Simulasi KPR yang berfungsi untuk membantu calon pembeli rumah dalam memperkirakan angsuran bulanan berdasarkan harga rumah, uang muka, suku bunga, dan jangka waktu kredit. Halaman ini hanya menampilkan hasil berdasarkan data yang diambil dari masukan pengguna, tanpa menyimpan data ke *backend* (*read-only*). Seluruh fitur dan tampilan telah dirancang agar responsif dan dapat diakses dengan baik melalui berbagai perangkat seperti *handphone*, *tablet*, dan *desktop*.

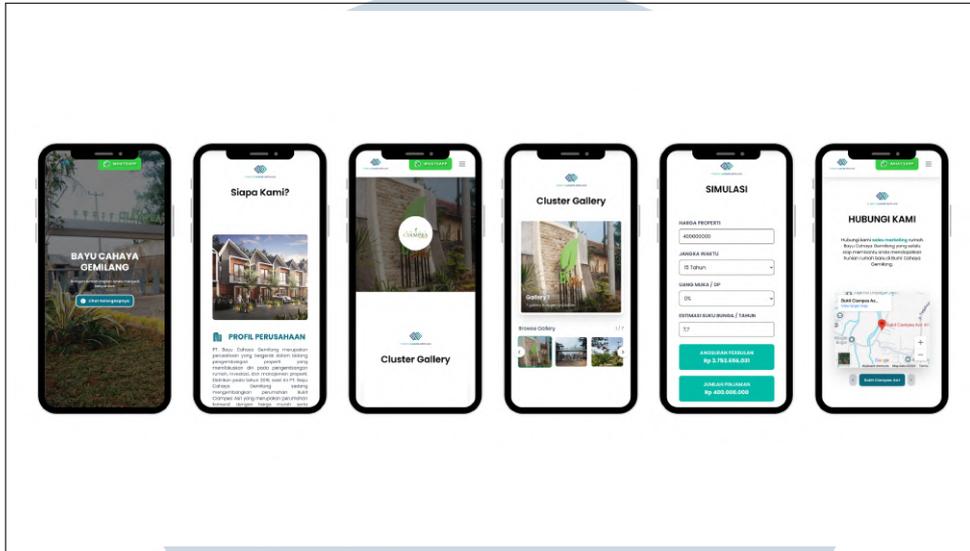
3.3.6.3.6 Halaman Kontak Kami



Gambar 3.49. Tampilan halaman Kontak Kami pada website Company Profile

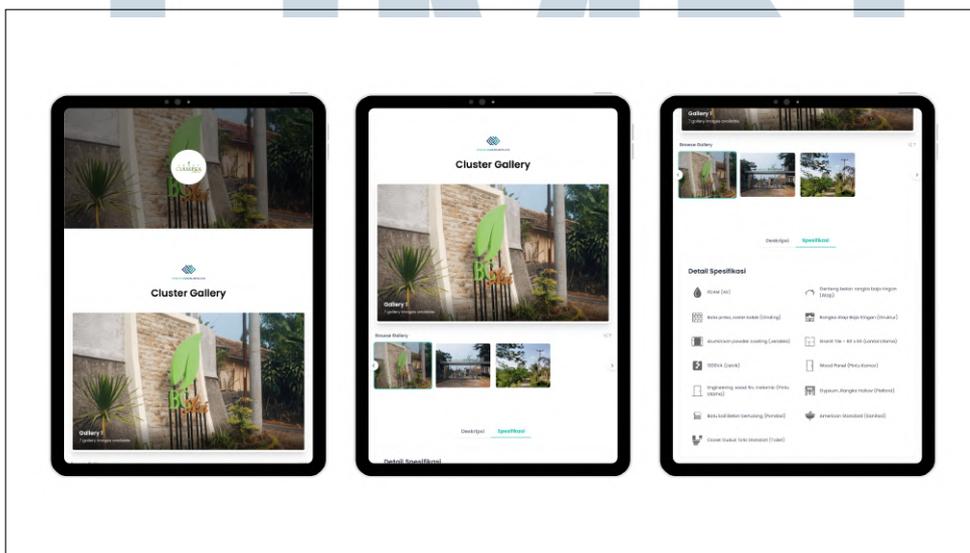
Gambar 3.49 menampilkan halaman akhir dari *website Company Profile* yang bertujuan untuk mempermudah pengunjung dalam menghubungi pihak *marketing* dari PT Bayu Cahaya Gemilang. Tampilan halaman ini di sisi kiri terdapat *embed Google Maps* yang menampilkan lokasi proyek perumahan Bukit Ciampea Asri, untuk memberikan konteks visual dan kemudahan navigasi bagi pengunjung. Di bawahnya juga tersedia *carousel* lokasi untuk melihat lokasi proyek perumahan lainnya. Di sisi kanan ditampilkan sebuah formulir kontak yang terdiri dari empat kolom input, yaitu Nama Lengkap, Nomor *Handphone*, *Email*, dan Pesan. *Form* ini terhubung langsung dengan *backend* menggunakan *library Nodemailer*, yang memungkinkan pesan dikirimkan ke *email* resmi perusahaan. Saat pengguna mengisi *form* dan menekan tombol "Kirim Pesan", server akan memproses data tersebut dan mengirimkan email berisi detail pesan ke alamat email tujuan, tanpa perlu menyimpan data ke dalam *database*. Dengan adanya integrasi *Nodemailer*, halaman ini tidak hanya menampilkan informasi, tetapi juga memberikan jalur komunikasi dua arah yang langsung dan praktis antara pengunjung *website* dan tim *marketing* PT Bayu Cahaya Gemilang.

3.3.6.3.7 Tampilan Responsif Website



Gambar 3.50. Tampilan responsif handphone pada website Company Profile

Gambar 3.50 menampilkan antarmuka *website* pada perangkat *handphone*. Tampilan ini menunjukkan bahwa seluruh komponen seperti *header*, konten, galeri, *form*, dan *footer* telah diatur ulang secara proporsional agar nyaman digunakan dalam layar berukuran kecil. Komponen seperti tombol dan teks memiliki jarak yang cukup agar tetap mudah diakses menggunakan sentuhan jari, serta *layout* konten ditata secara vertikal agar sesuai dengan tinggi layar.



Gambar 3.51. Tampilan responsif tablet pada website Company Profile

Gambar 3.51 menampilkan tampilan pada perangkat tablet. *Layout*

responsif disesuaikan agar tetap menampilkan informasi dengan jelas dalam format yang lebih lebar dari *handphone*, namun belum seluas *desktop*. Komponen seperti galeri, *tab* deskripsi dan spesifikasi, serta elemen visual lainnya tetap tersusun rapi dan adaptif terhadap perubahan orientasi layar. Hal ini menunjukkan bahwa desain yang diterapkan telah dibuat responsif secara menyeluruh untuk mendukung pengalaman pengguna lintas perangkat.

3.4 Kendala dan Solusi yang Ditemukan

Pada bagian ini akan dijelaskan berbagai kendala yang dijumpai selama pelaksanaan kerja magang, serta langkah-langkah yang diambil untuk mengatasinya. Penjelasan ini bertujuan untuk menyajikan pandangan yang jelas mengenai kendala yang ada di lapangan serta langkah-langkah yang diambil untuk mengatasi masalah tersebut.

3.4.1 Kendala yang Ditemukan

Berikut ini merupakan beberapa kendala yang dihadapi selama pelaksanaan kerja magang, yaitu:

1. Kesulitan dalam menanggung beban tanggung jawab proyek secara menyeluruh, mulai dari tahap perancangan desain, pengembangan antarmuka pengguna, hingga implementasi sistem *backend*. Sebagai mahasiswa magang yang belum memiliki pengalaman mendalam, terlebih dalam pengembangan *backend*, mahasiswa dituntut untuk menyelesaikan proyek *website Company Profile* dan *CMS Admin* seorang diri (meskipun terdapat rekan magang lain, proyek yang dikerjakan berbeda dan pengerjaannya dilakukan secara mandiri). Proyek ini ditargetkan selesai dalam waktu 4 bulan, yang mencakup seluruh proses desain, *frontend*, dan *backend*, sehingga menjadi tantangan besar dalam hal manajemen waktu dan penguasaan teknologi.
2. Kesulitan dalam mendapatkan arahan fitur dan aset visual dari pihak perusahaan. Pada awal pengembangan, mahasiswa dan rekan hanya diberikan gambaran umum bahwa *website* harus memiliki halaman Beranda, Tentang Kami, Developer, Simulasi KPR, dan Kontak Kami, tanpa penjelasan lebih lanjut terkait konten atau sesi yang harus dihadirkan pada masing-masing halaman. Hanya halaman Simulasi KPR yang dijelaskan memuat

fitur simulasi dan daftar bank rekanan. Akibatnya, kami harus menyusun sendiri ide konten dan struktur halaman melalui eksplorasi mandiri. Selain itu, aset gambar untuk beberapa *cluster* seperti Artha Soreang Living dan Pasanggrahan Hill juga tidak tersedia lengkap, sehingga menyebabkan keterlambatan dalam proses pembuatan tampilan.

3. Terdapat kendala komunikasi dengan pihak perusahaan yang bukan berasal dari latar belakang teknologi informasi. PT Bayu Cahaya Gemilang belum memiliki divisi TI secara khusus, sehingga proses komunikasi saat diskusi fitur dan pengujian fungsi *website* cukup sulit dilakukan. Beberapa penjelasan mengenai alur penggunaan, fitur, dan implementasi teknis perlu disederhanakan agar dapat dipahami oleh pihak non-teknis.
4. Terdapat permasalahan teknis dalam proses penghapusan *file* gambar pada integrasi *Cloudinary*. Saat melakukan proses *update* atau *delete*, *file* gambar lama yang tidak lagi digunakan tidak terhapus secara otomatis dari penyimpanan. Hal ini berpotensi menyebabkan penumpukan data dan pemborosan kapasitas penyimpanan jika tidak ditangani dengan tepat.

3.4.2 Solusi yang Ditemukan

Solusi dari kendala yang dihadapi selama pelaksanaan kerja magang adalah sebagai berikut:

1. Untuk mengatasi kendala dalam menangani proyek secara menyeluruh, mahasiswa menyusun strategi pembelajaran mandiri secara bertahap mengenai pengembangan antarmuka dengan *Tailwind CSS* yang responsif, serta pemahaman mengenai *Node.js* dan *Express* dalam membangun layanan *REST API*. Referensi belajar mahasiswa mayoritas ada di platform *youtube* milik "Web Programming UNPAS". Selain itu, mahasiswa menerapkan manajemen waktu yang disiplin dengan menyusun *timeline* pribadi, yaitu bulan Januari untuk desain *website Company Profile*, Februari untuk implementasi *frontend*, Maret untuk desain *CMS Admin*, dan April untuk implementasi sistem *CMS Admin*. Dengan pendekatan ini, mahasiswa dapat menyelesaikan sebagian besar pekerjaan lebih cepat dari rencana awal, sehingga memiliki waktu lebih untuk fokus memperdalam penguasaan teknis *backend*.

2. Dalam menghadapi kurangnya arahan fitur dan keterbatasan aset visual, mahasiswa mengambil inisiatif untuk melakukan riset terhadap berbagai referensi desain *website* perusahaan properti, guna mengidentifikasi sesi atau konten penting yang umum digunakan. Langkah ini membantu mahasiswa dalam menentukan komposisi konten yang relevan untuk *website* PT Bayu Cahaya Gemilang. Untuk mengatasi keterbatasan aset gambar, mahasiswa tetap melanjutkan pembangunan struktur halaman menggunakan *template* dan gambar sementara, sehingga waktu pengembangan tidak terbuang dan tetap mengikuti jadwal yang telah mahasiswa tetapkan.
3. Untuk mengatasi kendala komunikasi dengan pihak non-teknis, mahasiswa menyesuaikan cara penyampaian informasi dengan menggunakan bahasa yang lebih umum dan mudah dipahami. Mahasiswa juga memanfaatkan media presentasi interaktif seperti demonstrasi langsung melalui tampilan *website*, serta mendokumentasikan fitur secara visual agar proses diskusi menjadi lebih lancar dan mengurangi kesalahpahaman. Setiap progres juga secara rutin mahasiswa klarifikasi kepada supervisor untuk memastikan bahwa hasil kerja tetap selaras dengan harapan perusahaan.
4. Permasalahan teknis dalam proses penghapusan gambar di *Cloudinary* berhasil diatasi setelah mahasiswa melakukan proses *debugging* secara bertahap. Mahasiswa menemukan bahwa kesalahan terdapat pada pendefinisian *path file* gambar yang tidak sesuai, sehingga gambar yang dimaksud tidak ditemukan pada direktori saat proses penghapusan dijalankan. Setelah memperbaiki struktur *path*, fungsi penghapusan dapat berjalan dengan baik dan tidak menimbulkan penumpukan *file* yang tidak terpakai.

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