

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Pelaksanaan magang dilakukan di Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Multimedia Nusantara (UMN) dalam divisi Pengabdian Masyarakat, yang fokus pada pengembangan aplikasi web untuk mendukung kegiatan pemberdayaan masyarakat. Posisi magang yang diemban adalah sebagai *Full Stack Developer* pada proyek pengembangan website profil suku Baduy, menggunakan framework *Laravel* sebagai teknologi utama dalam pembangunan sistem.

Tugas utama meliputi perancangan dan pengembangan sistem secara keseluruhan, yang mencakup baik bagian frontend maupun backend website. Pada sisi backend, pengembangan dilakukan menggunakan *Laravel 11.x* sebagai framework *PHP* modern dengan implementasi pola *MVC (Model-View-Controller)*. Fokus pengembangan backend mencakup manajemen data, autentikasi pengguna, serta pengelolaan konten artikel, produk, dan gambar yang ditawarkan oleh masyarakat Baduy. Sistem *Role-based Access Control (RBAC)* diterapkan untuk membedakan hak akses pengguna, baik untuk admin maupun pengguna biasa. Selain itu, *Laravel Sanctum* digunakan untuk mengelola token *API* dan memastikan keamanan sistem melalui autentikasi berbasis session maupun *API*.

Koordinasi tim dilakukan dengan pembagian tugas berdasarkan fitur dan diskusi teknis rutin setiap dua hari sekali melalui platform daring seperti *Zoom Meeting*. Untuk pengembangan kode, digunakan *GitHub* dengan sistem branching untuk memastikan keteraturan dalam pengelolaan kode dan menghindari konflik. Komunikasi harian dilakukan melalui grup *WhatsApp* untuk mempermudah pertukaran informasi antar anggota tim, sedangkan klarifikasi terkait alur bisnis dilakukan melalui pertemuan langsung dengan pihak-pihak terkait. Pertemuan berkala juga diadakan dengan supervisor LPPM untuk melakukan evaluasi progres, membahas kendala, serta merencanakan pengembangan fitur tambahan.

#### 3.2 Tugas yang Dilakukan

Selama pelaksanaan magang, tugas utama berfokus pada perancangan dan pengembangan sistem aplikasi website, dengan perhatian khusus pada

pengembangan modul admin. Tugas awal yang dilakukan meliputi perancangan dan pembangunan modul admin, yang mencakup pengelolaan database produk, carousel, serta pembuatan role (*guest*, *user*, *admin*, dan *superadmin*). Pengembangan tampilan antarmuka pengguna untuk halaman admin juga menjadi bagian integral dari tugas yang dikerjakan.

Seiring berjalannya waktu, tanggung jawab diperluas untuk mencakup pengintegrasian fitur *CRUD* yang awalnya hanya tersedia di halaman admin agar dapat ditampilkan di halaman yang relevan bagi pengguna. Fitur produk dan carousel, yang sebelumnya hanya dapat ditambahkan dan dihapus melalui halaman admin, kini dapat ditampilkan pada homepage dan halaman produk yang ditujukan untuk pengguna. Untuk mewujudkan hal tersebut, dilakukan pembuatan routing yang memungkinkan pengambilan data produk dari database dan menampilkannya di halaman yang sesuai. Selain itu, penyesuaian frontend dilakukan untuk memastikan tampilan yang konsisten dan bebas dari error.

Pada bagian autentikasi dan otorisasi, tugas yang dilakukan mencakup pembangunan fitur login, register, dan forgot password, dengan penyesuaian pada database pengguna. Sistem autentikasi ini dibangun untuk memberikan pengalaman pengguna yang aman dan mudah. Selama pengembangan, ditemukan beberapa bug, salah satunya terkait dengan pengunggahan gambar profil pengguna. Masalah ini diatasi dengan merancang sistem penyimpanan gambar menggunakan metode *BLOB* dan menyatukan semua gambar dalam satu tabel *images* di database, yang memastikan pengelolaan gambar dapat dilakukan dengan lebih efisien.

Selain itu, pada bagian *superadmin* page, ditambahkan fitur *CRUD* admin yang memungkinkan hanya *superadmin* yang dapat menambah atau menghapus admin melalui pengisian email pengguna. Pada bagian artikel, perbaikan juga dilakukan pada proses approval artikel, yang sebelumnya mengalami kesalahan. Seluruh proses evaluasi bug dan perbaikan dilakukan secara rutin untuk memastikan aplikasi berjalan dengan baik dan memenuhi standar operasional yang ditetapkan.

### **3.3 Uraian Pelaksanaan Magang**

Selama pelaksanaan magang, pengembangan website profil suku Baduy menggunakan *Laravel 11.x* sebagai framework backend, dengan *Eloquent ORM* untuk interaksi dengan *MySQL 8.0+* sebagai basis data. Sistem autentikasi dibangun menggunakan *Laravel Sanctum* untuk *API token* dan *Laravel Authentication* untuk login berbasis sesi, dilengkapi dengan *Role-based Access*

*Control.* Pada sisi frontend, *Laravel Blade* digunakan untuk templating, sementara *Tailwind CSS* memastikan desain yang responsif dan modern. *Alpine.js* memberikan elemen interaktif tanpa memerlukan proses build kompleks. Fitur utama termasuk pengelolaan gambar dengan *BLOB storage*, sistem persetujuan artikel, dan fungsi *marketplace* untuk produk Baduy.

Selain itu, sistem dibangun dengan berbagai fitur tambahan, seperti *polymorphic relationships* untuk pengelolaan gambar fleksibel, serta *Laravel Vite* untuk optimasi dan pengelolaan aset. Pengelolaan gambar dilakukan menggunakan *Intervention Image library* dan *GD Extension* untuk pemrosesan gambar, serta *WebP Conversion* untuk optimasi. Semua pengembangan bertujuan untuk memberikan aplikasi yang aman, efisien, dan mudah digunakan, dengan menggunakan teknologi modern yang mendukung performa dan pengalaman pengguna optimal.

Rincian aktivitas selama pelaksanaan diuraikan pada tabel 3.2.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan rapat perdana dengan <i>supervisor</i> dan tim Project Baduy untuk membahas penentuan <i>jobdesk</i> dan perencanaan awal proyek. Mencari referensi terkait pembuatan website dan framework <i>Laravel</i> terbaru, serta mulai merancang mockup desain awal untuk halaman login dan register.
2	Membuat repositori <i>GitHub</i> untuk project Baduy dan melanjutkan pembuatan mockup untuk halaman login dan register. Mencari referensi untuk desain admin page dan menentukan warna utama untuk website.
3	Menyelesaikan mockup halaman login dan register, serta mengirimkan assets kepada tim terkait. Melakukan koordinasi dengan tim mengenai pengembangan desain halaman login dan admin.
4	Melanjutkan pengerjaan mockup halaman login dan register, serta merapikan desain mockup Admin Page. Mengirimkan assets dan berkordinasi dengan tim mengenai pengembangan lebih lanjut.
Lanjut pada halaman berikutnya	

Minggu Ke -	Pekerjaan yang dilakukan
5	Membangun fitur autentikasi untuk login, register, dan forgot password pada <i>Laravel</i> , serta melakukan implementasi dan pengujian awal untuk fitur forgot password.
6	Menyelesaikan pengembangan fitur forgot password dan mengatasi masalah bug yang muncul. Melakukan rapat rutin dengan tim untuk membahas perkembangan dan kendala yang dihadapi.
7	Mengembangkan dan menguji fitur <i>CRUD</i> untuk admin page, termasuk pengelolaan artikel, produk, dan carousel. Memperbaiki bug yang muncul pada sistem <i>CRUD</i> admin.
8	Melanjutkan pengembangan dan pengujian fitur <i>CRUD</i> pada admin page, serta berkoordinasi dengan tim mengenai progres pengerjaan website.
9	Mempelajari dan mengimplementasikan struktur database serta pembuatan tabel untuk produk, artikel, dan carousel. Menyusun skema relasi antar tabel untuk mendukung pengelolaan data.
10	Menyelesaikan fitur <i>CRUD</i> untuk produk dan carousel serta melakukan pengujian untuk memastikan semua data terkelola dengan baik dan tidak ada error.
11	Membangun dan menguji sistem penyimpanan gambar menggunakan metode <i>BLOB</i> untuk foto profil, produk, artikel, dan carousel. Menyelesaikan pengujian backend dan memperbaiki bug yang ditemukan.
12	Mengembangkan dan mengimplementasikan fitur forgot password, serta menguji sistem dengan <i>Mailtrap</i> untuk memastikan fitur berfungsi dengan benar dan token kedaluwarsa setelah 60 menit.
13	Menyelesaikan fitur forgot password dan melakukan pengujian lebih lanjut. Memperbaiki masalah yang terjadi dengan sistem penyimpanan gambar menggunakan <i>BLOB</i> .
Lanjut pada halaman berikutnya	

Minggu Ke -	Pekerjaan yang dilakukan
14	Melanjutkan perbaikan dan pengujian sistem gambar <i>BLOB</i> , serta menyelesaikan masalah dengan foto profil pengguna yang tidak terunggah dengan benar.
15	Menyelesaikan dan menguji seluruh fitur gambar menggunakan <i>BLOB</i> . Memastikan tidak ada error saat menginput atau menampilkan gambar. Melakukan persiapan untuk mengintegrasikan semua fitur yang telah selesai ke dalam branch utama proyek.
16	Melakukan finalisasi dan pengujian seluruh fitur yang telah selesai, memastikan integrasi antar fitur berjalan lancar tanpa kendala. Menyusun dokumentasi terkait pengembangan website dan membuat laporan kemajuan terakhir dari proyek magang.
17	Melakukan hosting website pada server, memastikan semua fitur berjalan dengan baik di lingkungan produksi. Melakukan uji coba aksesibilitas dan performa website untuk memastikan website dapat diakses dengan lancar oleh pengguna. Menyelesaikan laporan akhir mengenai seluruh kegiatan yang telah dilakukan selama magang.

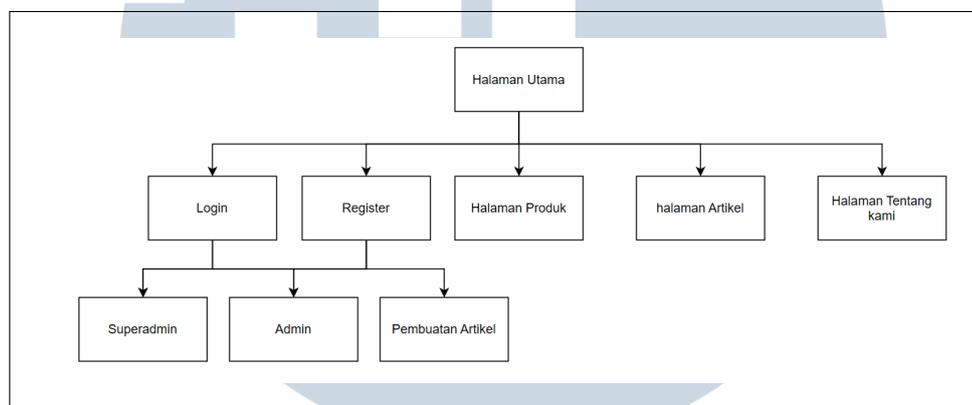
### 3.3.1 Sitemap

Website ini dirancang dengan alur navigasi yang jelas dan sederhana untuk memudahkan pengguna dalam menelusuri konten yang tersedia. Sitemap dimulai dengan halaman utama atau *Homepage*, yang berfungsi sebagai titik awal bagi pengunjung untuk mengakses fitur-fitur lainnya. Dari halaman *homepage*, pengguna dapat mengakses halaman-halaman lain seperti Halaman *Produk*, yang menampilkan berbagai produk yang dijual oleh masyarakat Baduy, serta Halaman *Artikel* yang berisi informasi dan artikel tentang kebudayaan dan kegiatan masyarakat Baduy.

Pengguna juga dapat mengunjungi Halaman *Tentang Kami*, yang menjelaskan latar belakang dan tujuan website ini. Halaman *Autentikasi* memungkinkan pengguna untuk melakukan login atau registrasi. Setelah login, pengguna dengan peran *Superadmin* atau *Admin* dapat mengakses fitur tambahan untuk mengelola produk, artikel, dan carousel di Halaman *Admin*. Pada halaman

admin, admin dapat menambah atau menghapus produk dan artikel yang terdaftar, serta mengelola konten yang tampil di website.

Salah satu fitur penting lainnya adalah Halaman *Pembuatan Artikel*, di mana user yang telah terdaftar dan berhasil login dapat membuat artikel baru yang akan dipublikasikan di halaman artikel website. Dengan desain yang intuitif dan mudah digunakan, website ini memastikan pengelolaan dan aksesibilitas yang baik bagi pengunjung dan pengguna terdaftar, baik untuk melihat konten maupun untuk mengelola data yang tersedia di dalamnya.



Gambar 3.1. Sitemap website Baduy Profile

### 3.3.2 Flowchart

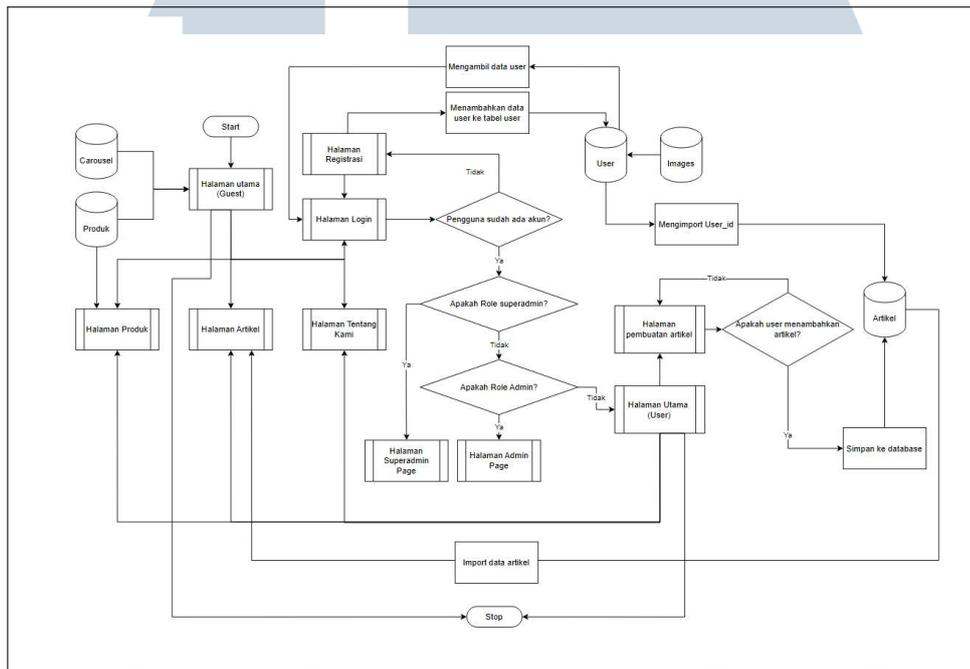
Proses penggunaan website dimulai dengan pengunjung yang mengakses *Homepage*. Halaman utama ini menampilkan fitur *carousel* dan daftar produk yang dapat dilihat oleh pengguna. Dari *homepage*, pengguna dapat menavigasi ke beberapa halaman lainnya, seperti Halaman *Produk*, yang menampilkan detail produk yang dijual oleh masyarakat Baduy, dan Halaman *Artikel*, yang berisi informasi terkait kebudayaan Baduy.

Setelah itu, pengguna dapat mengakses Halaman *Tentang Kami*, yang memberikan informasi tentang latar belakang website dan tujuan pengembangannya. Jika pengguna ingin mengakses fitur tambahan seperti mengelola artikel atau produk, mereka perlu melakukan login melalui Halaman *Autentikasi*. Pengguna yang sudah terdaftar dapat masuk dengan akun mereka, sementara pengguna baru dapat melakukan registrasi.

Setelah berhasil login, pengguna yang memiliki peran *Superadmin* atau *Admin* dapat mengakses Halaman *Admin*, yang memungkinkan mereka untuk mengelola produk, artikel, dan *carousel*. Pengguna *Admin* dapat menambah,

mengedit, atau menghapus produk dan artikel yang ada. Selain itu, *Superadmin* memiliki wewenang penuh untuk mengelola seluruh sistem, termasuk menambah atau menghapus *Admin*.

Terakhir, pengguna dengan peran *User* dapat membuat artikel baru setelah mereka login. Artikel yang dibuat akan melalui proses persetujuan dari *Admin* atau *Superadmin* sebelum dipublikasikan di website.



Gambar 3.2. Flowchart general Baduy Profile

### A Flowchart Halaman Superadmin

Diagram alur pada gambar 3.3 dimulai ketika sistem mendeteksi pengguna yang mengakses website. Sistem akan secara otomatis memeriksa data pengguna yang ada di database, memastikan apakah pengguna memiliki role *Superadmin* dan mencocokkan *user\_id* yang terdaftar. Jika verifikasi berhasil dan peran pengguna adalah *Superadmin*, sistem kemudian akan langsung mengarahkan pengguna ke *Superadmin Page*.

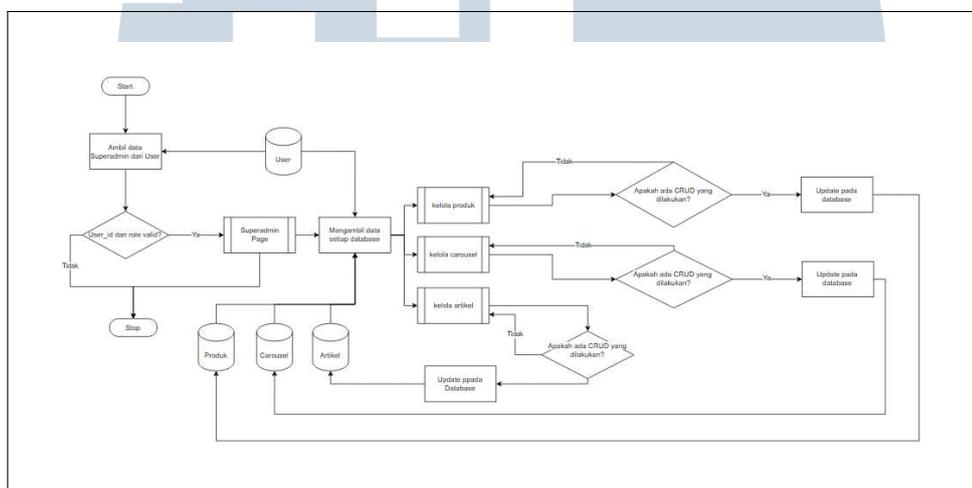
Di *Superadmin Page*, terdapat empat fitur utama yang dapat dikelola oleh *Superadmin*, yaitu *Kelola Admin*, *Kelola Produk*, *Kelola Carousel*, dan *Kelola Artikel*. Pada fitur *Kelola Admin*, *Superadmin* dapat menambah, menghapus, atau memperbarui data admin yang terdaftar dalam sistem. Fitur *Kelola Produk* memungkinkan *Superadmin* untuk mengelola produk yang ada di website, termasuk menambah, mengedit, atau menghapus produk. Sementara itu, pada *Kelola*



menarik.

Untuk *Kelola Artikel*, *Admin* hanya memiliki akses untuk melakukan persetujuan atau penolakan terhadap artikel yang telah dibuat oleh *User*. *Admin* dapat memverifikasi dan memutuskan apakah artikel yang diajukan oleh *User* akan dipublikasikan atau tidak. Namun, *Admin* tidak dapat membuat atau mengedit artikel yang sudah ada, kecuali yang telah dibuat oleh *User*.

Setelah selesai mengelola fitur-fitur tersebut, *Admin* dapat memilih untuk logout, yang akan mengarahkan mereka kembali ke halaman login.



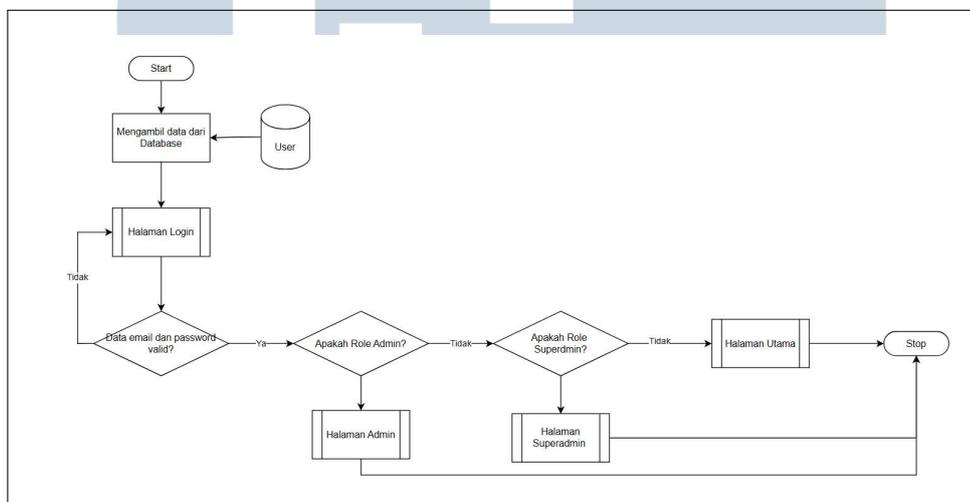
Gambar 3.4. Flowchart Admin Baduy Profile

### C Flowchart Halaman Login

Diagram alur pada gambar 3.5 menjelaskan proses kerja halaman *Login* pada website. Proses dimulai ketika pengguna mengakses halaman *Login* dan memasukkan email serta password mereka. Setelah data dimasukkan, sistem akan memverifikasi informasi yang diberikan dengan membandingkannya dengan data yang tersimpan di dalam database. Jika informasi yang dimasukkan salah, sistem akan menampilkan pesan kesalahan yang jelas dan meminta pengguna untuk mencoba lagi. Pengguna diberi kesempatan untuk memperbaiki data yang salah dan mengulang proses *login*. Apabila email dan password yang dimasukkan benar, sistem akan melanjutkan proses verifikasi dengan memeriksa *role* atau peran pengguna, yang dapat berupa *Superadmin*, *Admin*, atau *User*.

Berdasarkan peran yang terdeteksi, sistem akan mengarahkan pengguna ke halaman yang sesuai dengan hak akses mereka. Jika pengguna terdeteksi sebagai *Superadmin*, mereka akan diarahkan ke *Superadmin Page*, yang memberi

mereka akses penuh untuk mengelola seluruh konten dan fitur di website, termasuk mengelola *admin* lainnya dan produk. Pengguna dengan peran *Admin* akan diarahkan ke *Admin Page*, di mana mereka hanya dapat mengelola produk, artikel, dan *carousel*, namun tidak memiliki akses untuk mengelola *admin* lainnya. Sedangkan, jika pengguna adalah *User*, mereka akan diarahkan ke *User Dashboard*, yang memungkinkan mereka untuk melihat produk, artikel, dan membuat artikel baru. Setelah selesai menggunakan sistem, pengguna dapat memilih untuk *logout*, yang akan mengakhiri sesi mereka dan mengarahkan mereka kembali ke halaman *Login* untuk memastikan keamanan dan pengelolaan sesi yang aman.



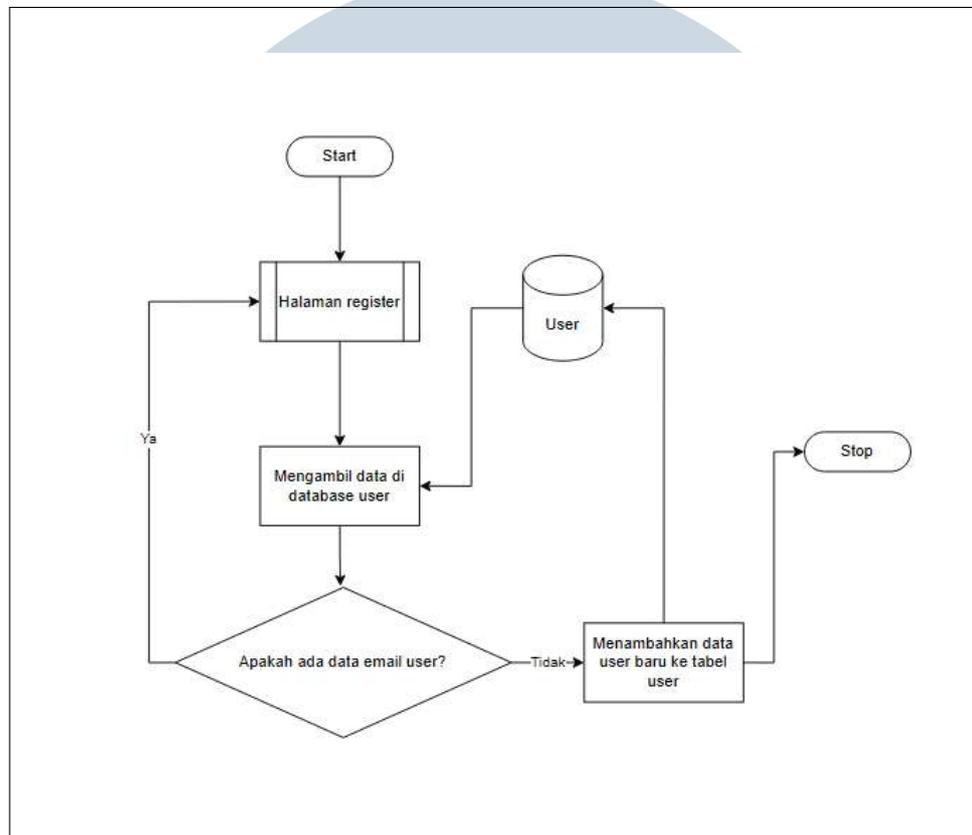
Gambar 3.5. Flowchart Login Baduy Profile

#### D Flowchart Halaman Register

Diagram alur pada gambar 3.6 menjelaskan proses kerja halaman *Register* pada website. Ketika sistem dijalankan, pengguna akan diarahkan ke halaman *Register* untuk membuat akun baru. Pada halaman ini, pengguna diminta untuk menginputkan data pribadi mereka, seperti nama, email, password, dan konfirmasi password. Setelah data dimasukkan, sistem akan memanggil database pengguna untuk memeriksa apakah sudah ada *user* dengan email yang sama.

Jika sistem menemukan bahwa email yang dimasukkan sudah terdaftar dalam database, pengguna akan diarahkan kembali ke halaman *Register* dengan pesan yang memberi tahu bahwa email tersebut sudah terdaftar. Pengguna kemudian diminta untuk menggunakan email yang berbeda. Namun, jika tidak ditemukan email yang sama, sistem akan melanjutkan untuk memasukkan data pengguna baru ke dalam database dan menetapkan *role* pengguna sebagai default

(User). Setelah data berhasil disimpan, proses registrasi selesai, dan pengguna dapat langsung *login* menggunakan email dan password yang telah didaftarkan.



Gambar 3.6. Flowchart Login Baduy Profile

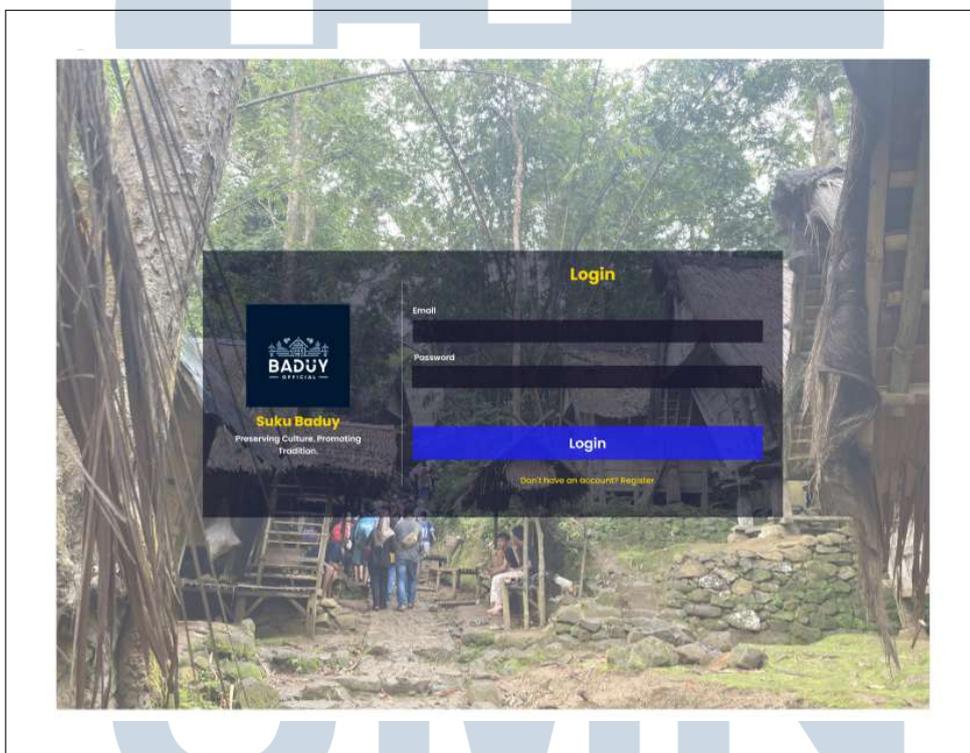
### 3.3.3 Perancangan Halaman Website

#### A Mockup Halaman Login

Halaman login Gambar 3.2 merupakan bagian yang penting dalam sistem, meskipun bukan halaman pertama yang diakses pengguna. Pada halaman ini, pengguna diminta untuk memasukkan email dan kata sandi mereka untuk mengakses berbagai fitur yang tersedia di website. Desain halaman login ini cukup sederhana namun jelas, dengan kolom input yang mudah diidentifikasi dan tombol login yang cukup mencolok, sehingga memudahkan pengguna untuk melakukan navigasi. Biasanya, pada bagian bawah halaman terdapat opsi untuk mengingatkan kata sandi atau untuk mengarahkan pengguna ke halaman reset kata sandi jika mereka lupa.

Dalam konteks *Human-Computer Interaction* (HCI), penting untuk

memperhatikan pengalaman pengguna yang nyaman dan efisien. Halaman login harus memiliki kolom input yang mudah dibaca dengan label yang jelas, sehingga pengguna dapat memasukkan informasi mereka dengan mudah. Umpan balik juga berperan penting, seperti pesan kesalahan yang muncul ketika pengguna salah memasukkan email atau kata sandi. HCI juga memastikan bahwa halaman login responsif, sehingga dapat digunakan dengan nyaman di berbagai perangkat, baik itu desktop maupun perangkat mobile. Selain itu, adanya petunjuk yang jelas jika pengguna lupa kata sandi sangat diperlukan untuk menghindari kebingungannya dan untuk meningkatkan kualitas interaksi pengguna dengan sistem.

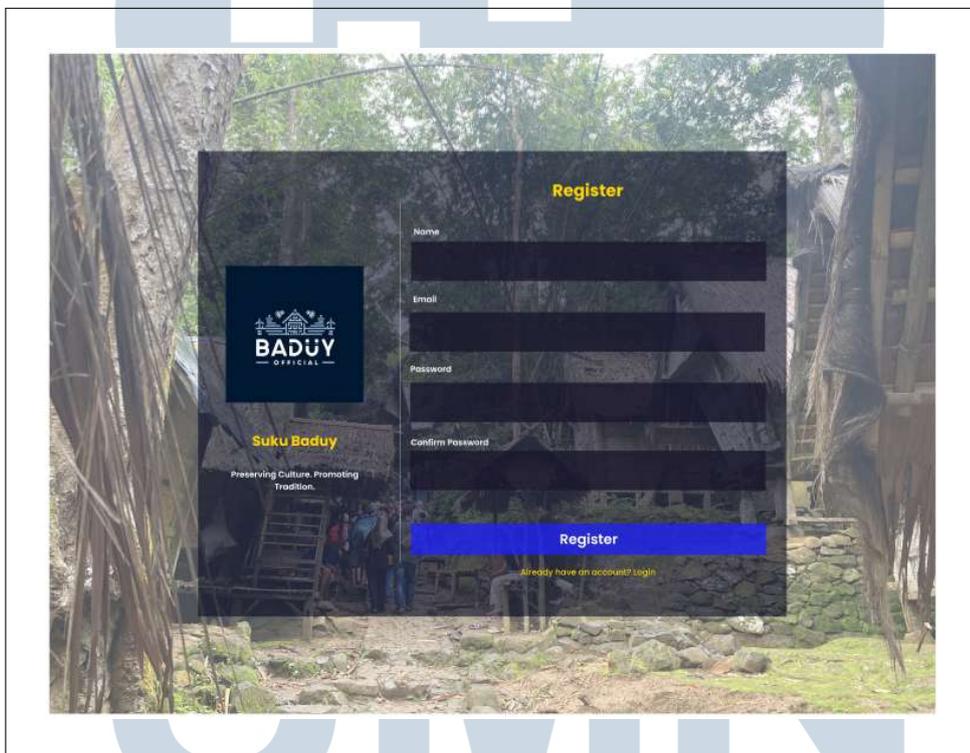


Gambar 3.7. *Mockup* Halaman Login

## B Mockup Halaman Register

Halaman register Gambar 3.3 dirancang untuk pengguna baru yang ingin membuat akun pada sistem. Pengguna diminta untuk mengisi informasi pribadi seperti nama, email, dan kata sandi. Desain halaman ini harus sederhana, jelas, dan mudah dipahami, dengan kolom input yang terorganisir dengan baik dan tombol register yang mudah ditemukan oleh pengguna. Setelah berhasil mendaftar, pengguna akan diarahkan ke halaman login untuk masuk ke akun yang baru saja dibuat.

Dari perspektif HCI, halaman register harus dirancang untuk memudahkan pengguna dalam mengisi data tanpa merasa tertekan. Kolom input yang terstruktur dengan baik dan umpan balik instan apabila ada kesalahan dalam pengisian (seperti email yang sudah terdaftar) akan meningkatkan kenyamanan pengguna saat melakukan pendaftaran. HCI juga mendorong penggunaan form yang tidak terlalu panjang agar pengguna dapat menyelesaikan proses pendaftaran dengan cepat dan tanpa kebingungannya. Penggunaan elemen desain yang familiar, seperti tombol "Daftar" atau "Kembali", akan memudahkan pengguna dalam berinteraksi secara lebih intuitif dengan sistem.



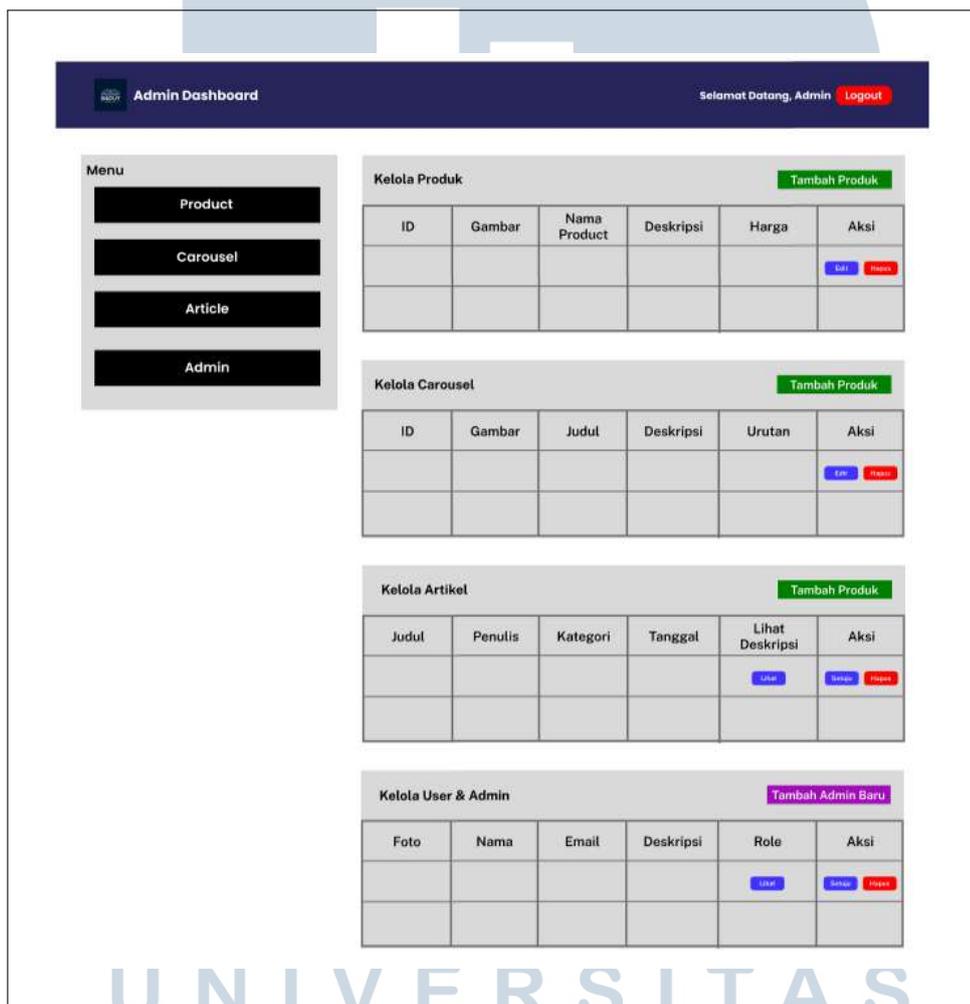
Gambar 3.8. *Mockup* Halaman Register

### C Mockup Halaman Superadmin

Halaman superadmin Gambar 3.4 memberikan akses penuh kepada pengguna dengan peran superadmin untuk mengelola seluruh aspek dari website. Pada halaman ini, superadmin dapat menambah, mengedit, atau menghapus admin, produk, artikel, dan gambar, serta mengatur hak akses dan pengelolaan sistem secara keseluruhan.

Dalam hal HCI, desain halaman superadmin harus memastikan bahwa navigasi antarmuka sangat jelas dan mudah diakses. Mengingat banyaknya

fitur pengelolaan yang harus ditangani oleh superadmin, sangat penting untuk mengelompokkan menu dengan cara yang logis serta memastikan tampilan yang bersih agar superadmin tidak merasa kebingungan. Penggunaan tombol dan dropdown yang intuitif untuk mengelola data sangat penting untuk meningkatkan efisiensi interaksi. Selain itu, penting juga untuk menyediakan konfirmasi atau peringatan yang muncul sebelum superadmin melakukan tindakan penting, seperti menghapus admin atau produk, untuk mencegah kesalahan yang tidak diinginkan.



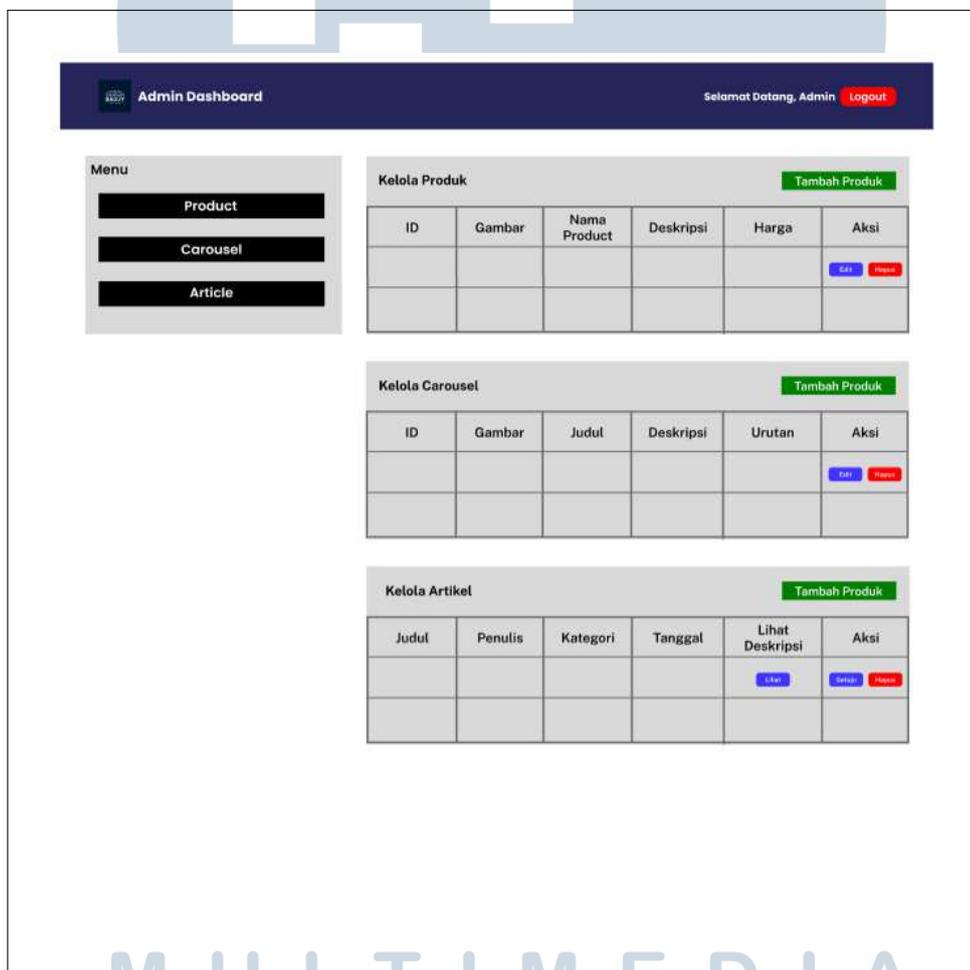
Gambar 3.9. *Mockup* Halaman Superadmin

#### D Mockup Halaman Admin

Halaman admin Gambar 3.5 memberikan akses kepada admin untuk mengelola produk, artikel, dan gambar yang ditampilkan di website. Admin dapat menambah, mengedit, atau menghapus produk dan artikel sesuai dengan kebutuhan.

Dari perspektif HCI, halaman admin harus dirancang untuk memudahkan

admin dalam mengelola konten dengan tampilan yang terorganisir dengan baik. Penggunaan tabel yang jelas untuk produk dan artikel, beserta tombol untuk menambah dan mengedit, harus ditempatkan dengan tepat agar admin dapat mengakses dan mengelola data dengan efisien. HCI juga mendorong penggunaan visual yang konsisten, seperti warna atau ikon yang membedakan setiap fungsi pengelolaan yang ada. Umpan balik yang jelas, seperti konfirmasi setelah admin melakukan suatu tindakan atau pemberitahuan jika terjadi kesalahan, sangat penting untuk meningkatkan efisiensi kerja dan mengurangi potensi kesalahan dalam pengelolaan data yang dapat memengaruhi kualitas website.



Gambar 3.10. Mockup Halaman Admin

### 3.3.4 Database

Dalam pengembangan website untuk suku Baduy, perancangan dan implementasi model basis data memainkan peran yang sangat penting untuk

memastikan bahwa setiap data tersimpan dengan rapi dan dapat diakses dengan efisien. Desain database yang baik akan memudahkan pengelolaan berbagai jenis data, seperti informasi pengguna, produk, artikel, serta data gambar yang digunakan dalam *carousel* dan produk. Oleh karena itu, beberapa model database dirancang untuk memenuhi berbagai kebutuhan sistem, dengan mempertimbangkan bagaimana data diorganisir, dihubungkan, dan diproses di dalam sistem.

Pada website ini, beberapa tabel utama telah dirancang untuk menyimpan informasi yang beragam. Tabel *users* menyimpan data pengguna, termasuk nama, email, password, dan *role* pengguna (*Superadmin*, *Admin*, atau *User*). Tabel *products* digunakan untuk menyimpan data produk yang dijual oleh masyarakat Baduy, yang mencakup nama produk, deskripsi, harga, kategori, dan pengelola produk. Sementara itu, tabel *articles* digunakan untuk menyimpan artikel yang ditulis oleh pengguna dan dikelola oleh *Admin* atau *Superadmin*, dengan fitur persetujuan untuk mempublikasikan artikel. Tabel *images* menyimpan gambar yang terkait dengan produk, artikel, dan *carousel*, dengan hubungan *polymorphic* untuk memastikan fleksibilitas penyimpanan gambar pada berbagai entitas. Untuk mengelola gambar yang ditampilkan di *homepage*, digunakan tabel *carousels* yang menyimpan data gambar *carousel*.

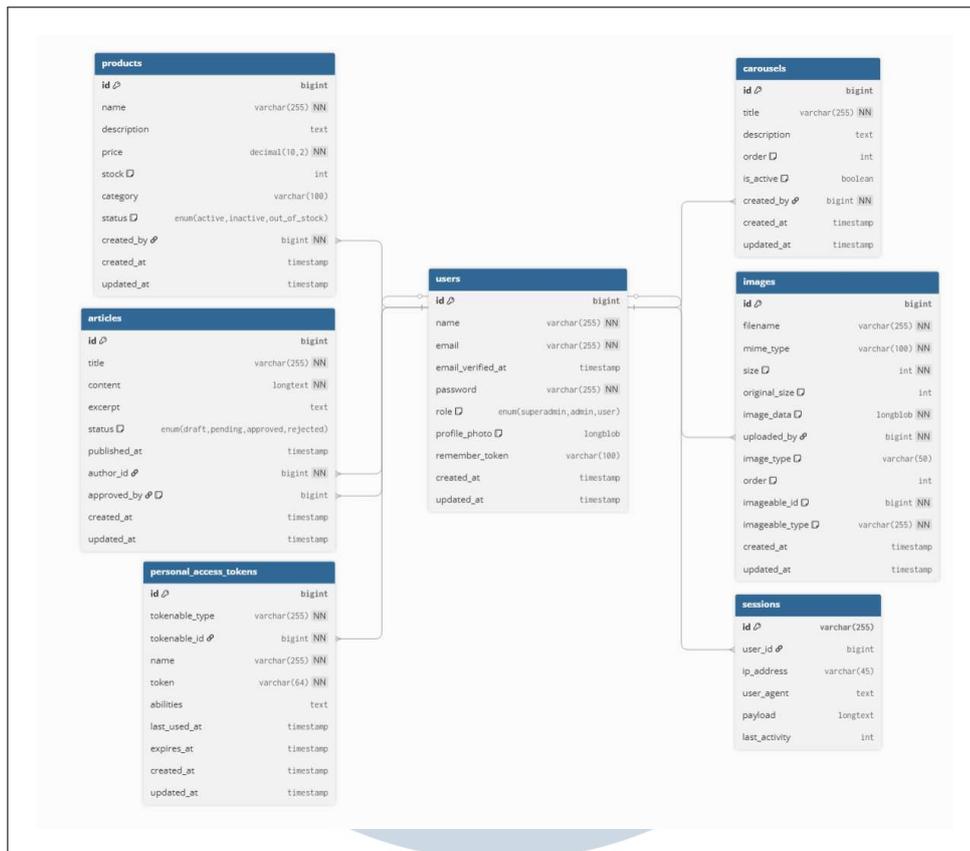
Relasi antar tabel sangat penting untuk memastikan integrasi antar data. Misalnya, tabel *users* memiliki relasi satu ke banyak dengan tabel *articles* dan *products*, yang menunjukkan bahwa satu pengguna dapat membuat banyak artikel atau produk. Selain itu, gambar yang diunggah akan terhubung dengan artikel atau produk melalui relasi *polymorphic* di tabel *images*, yang memungkinkan satu gambar untuk digunakan pada berbagai entitas (*artikel*, *produk*, atau *carousel*). Semua tabel dan relasi antar tabel ini dirancang dengan mempertimbangkan efisiensi dalam pencarian data, pengelolaan integritas referensial, serta validasi yang ketat agar tidak terjadi kesalahan data yang dapat mempengaruhi sistem secara keseluruhan.

## A Skema Database

Skema database pada website ini dirancang untuk memastikan penyimpanan data yang terstruktur dengan baik dan dapat diakses secara efisien. Tabel *users* menyimpan informasi mengenai pengguna, termasuk *id* (*primary key*), *email*, *password*, dan *role* yang menentukan hak akses pengguna (*Superadmin*, *Admin*, atau *User*). Tabel ini menjadi dasar dalam mengelola hak akses dan otorisasi

pengguna. Tabel *articles* menyimpan artikel yang ditulis oleh pengguna, dengan *user\_id* yang berfungsi sebagai *foreign key* yang menghubungkan artikel dengan penulisnya (*referensi ke users.id*). Artikel ini juga memiliki status persetujuan, di mana hanya artikel yang disetujui oleh *Admin* atau *Superadmin* yang dipublikasikan di website. Tabel *products* digunakan untuk menyimpan informasi produk yang dijual, termasuk nama, deskripsi, harga, dan kategori produk. Setiap produk dihubungkan dengan *user\_id* yang merujuk pada pengguna yang membuat produk tersebut (*referensi ke users.id*). Tabel *images* menyimpan gambar-gambar yang digunakan pada artikel, produk, dan *carousel*. Gambar ini disimpan menggunakan *BLOB storage* dan dihubungkan ke entitas terkait menggunakan hubungan *polymorphic*, memungkinkan satu gambar digunakan untuk berbagai entitas dalam sistem. Setiap gambar dalam tabel *images* memiliki *imageable\_id* yang dapat merujuk ke entitas seperti *carousels.id*, *products.id*, atau *articles.id*, tergantung pada jenis gambar yang diunggah. Tabel *carousels* menyimpan data gambar yang ditampilkan di bagian *carousel* halaman utama website, yang juga terhubung dengan gambar melalui *imageable\_id* dan *imageable\_type*.

Relasi antar tabel diorganisasikan dengan sangat hati-hati, seperti pada hubungan *users* dengan *articles* dan *users* dengan *products*, yang menggambarkan bahwa setiap pengguna dapat membuat banyak artikel atau produk, tetapi setiap artikel atau produk hanya dimiliki oleh satu pengguna. Hubungan serupa terjadi pada tabel *articles* dengan *images*, *products* dengan *images*, dan *carousels* dengan *images*, di mana setiap entitas (*artikel*, *produk*, atau *carousel*) dapat memiliki banyak gambar, namun setiap gambar hanya terhubung dengan satu entitas. Dalam desain ini, *primary key* digunakan untuk memastikan setiap entitas dalam tabel memiliki ID yang unik, sementara *foreign key* menghubungkan tabel satu dengan yang lainnya untuk memastikan integritas data. Dengan desain ini, database dapat mengelola berbagai jenis data secara efisien, memastikan integritas data, serta memfasilitasi akses dan pembaruan data secara cepat. Skema database yang baik ini sangat penting untuk mendukung pengoperasian website, memastikan bahwa setiap data yang diperlukan untuk tampilan dan interaksi pengguna dapat diakses dengan mudah dan aman.



Gambar 3.11. Skema Database Baduy Profile

## B Model Users

Model *Users* pada website ini berfungsi untuk menyimpan informasi terkait setiap pengguna yang terdaftar dalam sistem. Tabel *users* merupakan komponen penting dalam sistem karena mengelola data pengguna yang memiliki berbagai peran, seperti *Superadmin*, *Admin*, dan *User*. Setiap pengguna diidentifikasi dengan *id* yang unik, yang menjadi kunci utama (*primary key*) dalam tabel ini. Kolom *email* pada tabel ini berfungsi sebagai *username* yang juga memiliki aturan unik (*unique key*) untuk memastikan bahwa tidak ada dua pengguna yang memiliki alamat email yang sama.

Selain *id* dan *email*, tabel *users* menyimpan *password* yang telah dienkripsi untuk memastikan keamanan data pengguna. Setiap pengguna memiliki *role*, yang berupa *enum* dan mendefinisikan hak akses mereka di dalam sistem. *Role* ini menentukan apakah pengguna dapat mengakses halaman dan fitur tertentu. Misalnya, *Superadmin* memiliki akses penuh untuk mengelola semua data dan fitur, sementara *Admin* hanya memiliki akses terbatas, dan *User* hanya bisa mengakses

fitur tertentu seperti melihat produk dan artikel.

Kolom *created\_at* menyimpan *timestamp* yang mencatat waktu pembuatan akun pengguna, yang berguna untuk melacak aktivitas pendaftaran pengguna. Model *users* ini juga memungkinkan sistem untuk memverifikasi dan mengelola hak akses pengguna secara efisien, sehingga memberikan kontrol yang lebih baik terhadap keamanan dan pengelolaan data di dalam website. Dengan desain tabel ini, database dapat memfasilitasi pengelolaan pengguna yang terstruktur dengan baik, memungkinkan implementasi sistem autentikasi dan otorisasi yang aman.



users	
id	bigint
name	varchar(255) NN
email	varchar(255) NN
email_verified_at	timestamp
password	varchar(255) NN
role	enum(superadmin,admin,user)
profile_photo	longblob
remember_token	varchar(100)
created_at	timestamp
updated_at	timestamp

Gambar 3.12. Model User

### C Model Carousels

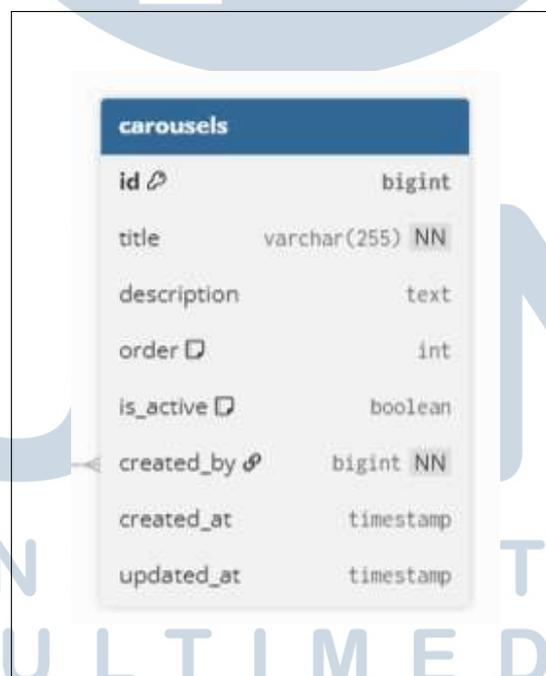
Model *Carousels* pada website ini digunakan untuk menyimpan dan mengelola gambar-gambar yang ditampilkan dalam bagian *carousel* di *homepage* atau halaman lainnya. Tabel *carousels* berperan penting untuk menampilkan gambar-gambar dinamis yang dapat menarik perhatian pengunjung website. Setiap gambar yang ditampilkan di *carousel* memiliki atribut yang mendeskripsikan gambar tersebut, termasuk *title*, *description*, *link\_url*, dan *is\_active*.

Kolom *id* pada tabel ini bertindak sebagai *primary key* yang unik untuk setiap entri *carousel*. Kolom *title* menyimpan judul atau nama dari gambar *carousel* yang dapat digunakan untuk memberikan informasi lebih lanjut mengenai gambar tersebut. *Description* berisi deskripsi singkat yang menjelaskan gambar

atau informasi terkait yang ditampilkan di *carousel*. Kolom *link\_url* menyimpan URL yang akan diakses ketika pengguna mengklik gambar *carousel*, yang memungkinkan pengunjung untuk diarahkan ke halaman tertentu di website atau ke *link* eksternal lainnya.

Kolom *order* digunakan untuk mengatur urutan tampilan gambar dalam *carousel*, sehingga gambar yang diatur dengan urutan lebih tinggi akan ditampilkan lebih dahulu atau diutamakan dalam tampilan. Kolom *is\_active* berfungsi untuk menandakan apakah gambar *carousel* aktif dan dapat ditampilkan di halaman atau tidak. Gambar dengan *is\_active* bernilai true akan ditampilkan, sedangkan gambar dengan *is\_active* bernilai false tidak akan muncul di tampilan website.

Dengan struktur tabel ini, sistem dapat mengelola gambar-gambar *carousel* secara fleksibel, memastikan bahwa hanya gambar yang aktif yang tampil di website, serta memungkinkan pengaturan urutan gambar dengan mudah. Tabel *carousels* juga terhubung dengan tabel *images* untuk menyimpan data gambar itu sendiri, menggunakan hubungan *one-to-many* antara tabel *carousels* dan *images*, memungkinkan banyak gambar untuk digunakan dalam satu entri *carousel*.



The image shows a screenshot of a database table structure for 'carousels'. The table has the following columns and data types:

Column Name	Data Type
id	bigint
title	varchar(255) NN
description	text
order	int
is_active	boolean
created_by	bigint NN
created_at	timestamp
updated_at	timestamp

Gambar 3.13. Model Carousels

#### D Model Products

Model *Products* pada website ini digunakan untuk menyimpan informasi terkait produk yang dijual oleh masyarakat Baduy di *marketplace* website. Tabel

*products* memiliki peranan penting dalam mengelola data produk yang tersedia, mulai dari nama produk, deskripsi, harga, kategori, hingga pengelolaan apakah produk tersebut ditampilkan sebagai produk unggulan atau tidak. Dengan adanya tabel ini, sistem dapat menyimpan dan mengelola data produk secara efisien, yang dapat diakses oleh pengguna dan pengelola website.

Kolom *id* pada tabel ini bertindak sebagai *primary key* yang unik untuk setiap produk. Kolom *name* menyimpan nama produk, yang akan tampil di halaman produk dan halaman detail produk. Kolom *description* menyimpan deskripsi dari produk, yang memberikan informasi lebih lanjut mengenai produk yang dijual, seperti bahan, kegunaan, atau keunikan produk tersebut. *Price* menyimpan harga produk yang dijual, sementara *category* menyimpan kategori produk yang memudahkan pengguna dalam mencari produk berdasarkan jenis atau klasifikasi tertentu.

Kolom *created\_by* adalah *foreign key* yang menghubungkan produk dengan pengguna yang membuat atau mengelola produk tersebut. Kolom ini mengacu pada *id* pengguna dalam tabel *users*, yang berarti setiap produk memiliki pengelola yang tercatat sebagai *User*, *Admin*, atau *Superadmin*. Kolom *is\_featured* digunakan untuk menandakan apakah produk tersebut ditampilkan sebagai produk unggulan di halaman utama atau tidak. Jika kolom ini bernilai *true*, maka produk akan ditampilkan di bagian produk unggulan website, menarik perhatian pengunjung yang melihat produk-produk tersebut.

Dengan desain tabel ini, sistem dapat mengelola produk dengan baik, memungkinkan *Admin* dan *Superadmin* untuk menambah, mengedit, dan menghapus produk yang ada, serta memberikan kontrol penuh terhadap daftar produk yang ditampilkan di website. Struktur ini juga memungkinkan pengelolaan produk secara dinamis, termasuk menampilkan produk unggulan yang lebih menonjol untuk meningkatkan daya tarik dan penjualan.

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

products	
id	bigint
name	varchar(255) NN
description	text
price	decimal(10,2) NN
stock	int
category	varchar(100)
status	enum(active, inactive, out_of_stock)
created_by	bigint NN
created_at	timestamp
updated_at	timestamp

Gambar 3.14. Model Produk

## E Model Articles

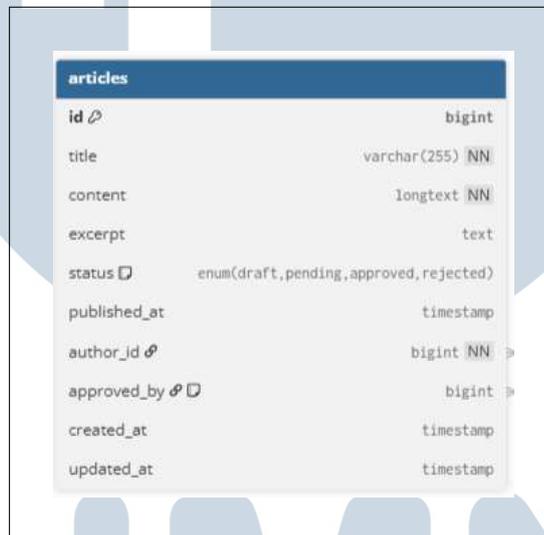
Model *Articles* pada website ini digunakan untuk menyimpan informasi mengenai artikel yang ditulis oleh pengguna, yang dapat berupa tulisan tentang kebudayaan, produk, atau informasi lain yang relevan dengan suku Baduy. Tabel *articles* berfungsi untuk mengelola konten yang ditulis oleh *User*, dan dapat dipantau oleh *Admin* dan *Superadmin* untuk proses persetujuan atau penolakan sebelum dipublikasikan di website.

Kolom *id* pada tabel ini bertindak sebagai *primary key* yang unik untuk setiap artikel. Kolom *title* menyimpan judul artikel, yang akan ditampilkan di halaman artikel atau daftar artikel. Kolom *content* menyimpan isi artikel, yang berisi teks atau informasi lengkap yang ingin dibagikan oleh pengguna. Kolom *genre* digunakan untuk mengategorikan artikel berdasarkan jenisnya, misalnya kategori kebudayaan, produk, atau sejarah. Kolom *status* menyimpan status artikel yang menunjukkan apakah artikel tersebut disetujui, ditolak, atau dalam proses persetujuan. Status ini dikelola oleh *Admin* atau *Superadmin*, yang memiliki wewenang untuk menentukan apakah artikel akan dipublikasikan atau tidak.

Kolom *user\_id* adalah *foreign key* yang menghubungkan artikel dengan pengguna yang menulisnya. *user\_id* merujuk pada *id* pengguna yang terdaftar dalam tabel *users*, sehingga artikel dapat dikaitkan dengan penulis yang sesuai. Kolom *approved\_by* adalah *foreign key* yang menghubungkan artikel dengan *Superadmin* atau *Admin* yang menyetujui artikel tersebut. Kolom

*view\_count* menyimpan jumlah tampilan atau kunjungan yang diterima oleh artikel, memberikan wawasan mengenai seberapa banyak pembaca yang tertarik dengan artikel tersebut. Kolom *created\_at* menyimpan *timestamp* yang mencatat kapan artikel tersebut dibuat.

Dengan struktur tabel ini, sistem dapat mengelola artikel dengan efisien, memungkinkan proses persetujuan dan penolakan artikel oleh *Admin* atau *Superadmin* sebelum dipublikasikan. Tabel *articles* juga memberikan kontrol yang jelas mengenai siapa yang menulis artikel dan siapa yang bertanggung jawab dalam proses persetujuan artikel tersebut, menjaga kualitas konten yang dipublikasikan di website.



articles	
id	bigint
title	varchar(255) NN
content	longtext NN
excerpt	text
status	enum(draft, pending, approved, rejected)
published_at	timestamp
author_id	bigint NN
approved_by	bigint
created_at	timestamp
updated_at	timestamp

Gambar 3.15. Model Articles

## F Model Images

Model *Images* pada website ini digunakan untuk menyimpan dan mengelola gambar-gambar yang digunakan dalam berbagai bagian website, seperti artikel, produk, dan *carousel*. Tabel *images* berfungsi sebagai tempat penyimpanan gambar dengan metode *BLOB storage (Binary Large Object)*, yang memungkinkan gambar disimpan langsung dalam database dalam bentuk binary data. Dengan cara ini, gambar dapat dengan mudah dikelola dan diakses dari sistem tanpa perlu bergantung pada penyimpanan eksternal.

Kolom *id* pada tabel ini bertindak sebagai *primary key* yang unik untuk setiap gambar yang diunggah. Kolom *filename* menyimpan nama file gambar yang diunggah, yang membantu dalam mengidentifikasi gambar secara cepat. Kolom *image\_data* menyimpan data gambar itu sendiri dalam bentuk *BLOB*, yang

memungkinkan penyimpanan gambar dalam database. Kolom *imageable\_type* digunakan untuk menunjukkan jenis entitas yang menggunakan gambar tersebut, misalnya apakah gambar tersebut terkait dengan artikel, produk, atau *carousel*. Kolom *imageable\_id* menghubungkan gambar dengan entitas yang relevan (seperti artikel atau produk) dan memungkinkan gambar ditampilkan bersama data yang sesuai.

Kolom *image\_type* menyimpan informasi mengenai jenis gambar, seperti format file (*JPEG, PNG, GIF, dll.*), yang dapat digunakan untuk optimasi gambar atau pengolahan lebih lanjut. Kolom *order* digunakan untuk mengatur urutan gambar, terutama pada entitas seperti *carousel* di *homepage*, untuk memastikan gambar ditampilkan sesuai urutan yang diinginkan. Kolom *uploaded\_by* adalah *foreign key* yang merujuk pada *id* pengguna yang mengunggah gambar, memungkinkan sistem untuk melacak siapa yang mengupload gambar tersebut.

Dengan desain tabel ini, sistem dapat mengelola gambar dengan lebih fleksibel, menyimpan berbagai gambar dalam satu tempat, dan menghubungkannya dengan entitas yang relevan, seperti artikel, produk, atau *carousel*. Selain itu, penggunaan *BLOB storage* memungkinkan gambar disimpan langsung dalam database, memberikan kemudahan dalam pengelolaan dan pengambilan data gambar untuk ditampilkan di website.



images	
id	bigint
filename	varchar(255) NN
mime_type	varchar(100) NN
size	int NN
original_size	int
image_data	longblob NN
uploaded_by	bigint NN
image_type	varchar(50)
order	int
imageable_id	bigint NN
imageable_type	varchar(255) NN
created_at	timestamp
updated_at	timestamp

Gambar 3.16. Model Images

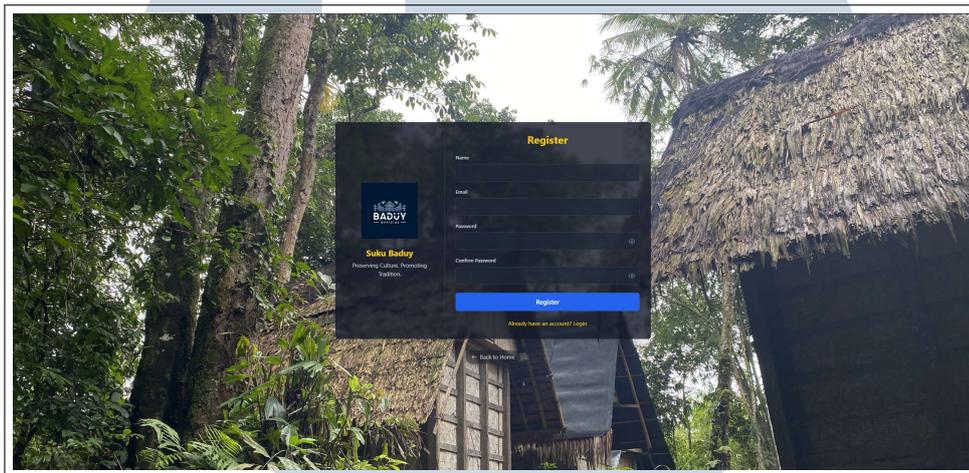
### 3.3.5 Implementasi Tampilan Halaman Website Baduy

#### A Halaman Register

Halaman register memungkinkan pengguna baru untuk membuat akun pada sistem dengan mengisi data pribadi seperti nama, email, dan kata sandi. Desain halaman ini mengutamakan kesederhanaan dan kemudahan penggunaan, dengan formulir input yang terstruktur dan jelas, memastikan pengguna dapat mengisi data tanpa kebingungan. Untuk meningkatkan keamanan, sistem real-time validation diterapkan untuk memverifikasi keunikan email dan kekuatan kata sandi yang dimasukkan. Fitur *password strength meter* memberikan indikator visual untuk menginformasikan pengguna mengenai sejauh mana kata sandi yang mereka buat memenuhi persyaratan keamanan. Selain itu, terdapat fitur *password match indicator* untuk memastikan bahwa kata sandi yang dimasukkan sesuai dengan konfirmasi yang diberikan oleh pengguna.

Setelah pengguna berhasil melakukan registrasi, mereka akan diarahkan ke halaman login untuk memulai sesi mereka dengan akun yang baru dibuat. Proses pendaftaran ini juga dilengkapi dengan pengiriman email verifikasi, yang

berisi tautan untuk mengaktifkan akun pengguna. Sistem ini menggunakan AJAX untuk memeriksa keunikan email secara langsung saat pengguna mengisi formulir, sementara indikator yang menunjukkan apakah kata sandi memenuhi standar keamanan juga memberikan umpan balik langsung kepada pengguna. Halaman ini juga menyediakan opsi *forgot password* yang memungkinkan pengguna untuk memulihkan kata sandi mereka jika lupa.



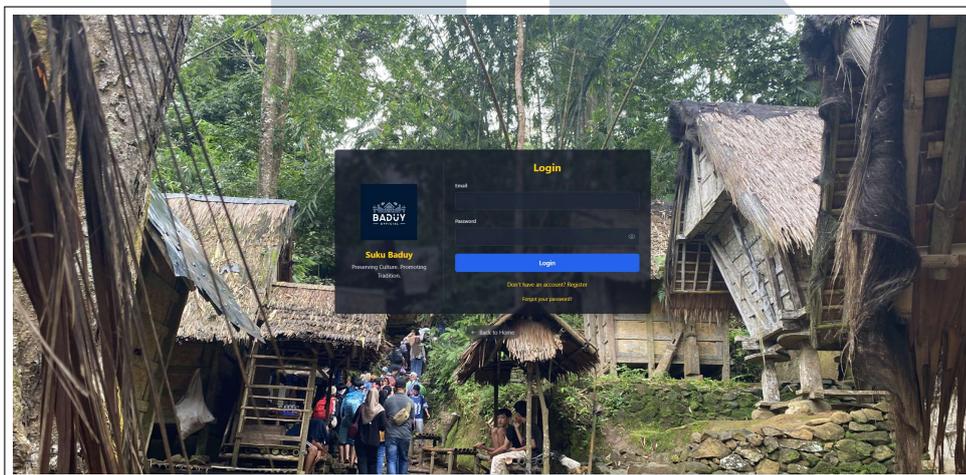
Gambar 3.17. Halaman *Register*

## B Halaman Login

Pada halaman login, pengguna diminta untuk memasukkan email dan kata sandi yang telah terdaftar di sistem untuk dapat mengakses fitur-fitur lebih lanjut dalam website. Fitur desain responsif memastikan bahwa tampilan halaman login disesuaikan dengan baik di berbagai perangkat, baik desktop maupun perangkat mobile, guna memberikan pengalaman yang konsisten di semua platform. Formulir pada halaman ini dirancang secara minimalis dengan kolom input yang jelas serta tombol login yang mencolok, sehingga memudahkan pengguna dalam navigasi. Selain itu, terdapat fitur *remember me* yang memungkinkan pengguna untuk tetap login pada sesi berikutnya tanpa harus mengisi kembali data login mereka.

Dari segi keamanan, halaman login dilengkapi dengan berbagai proteksi yang bertujuan untuk melindungi integritas dan kerahasiaan sesi pengguna. Fitur *CSRF protection* yang disediakan oleh *Laravel* mencegah potensi serangan pemalsuan permintaan lintas situs, sedangkan *rate limiting* berfungsi untuk mencegah terjadinya serangan brute force pada akun pengguna. Sistem manajemen sesi yang aman memastikan bahwa sesi pengguna tetap terlindungi selama interaksi dengan sistem. Pengguna juga diberikan opsi untuk

menampilkan/menyembunyikan kata sandi, yang memudahkan mereka memeriksa kata sandi yang dimasukkan. Jika terjadi kesalahan pada proses login, pesan error yang jelas dan informatif akan muncul untuk memberikan umpan balik yang langsung kepada pengguna.

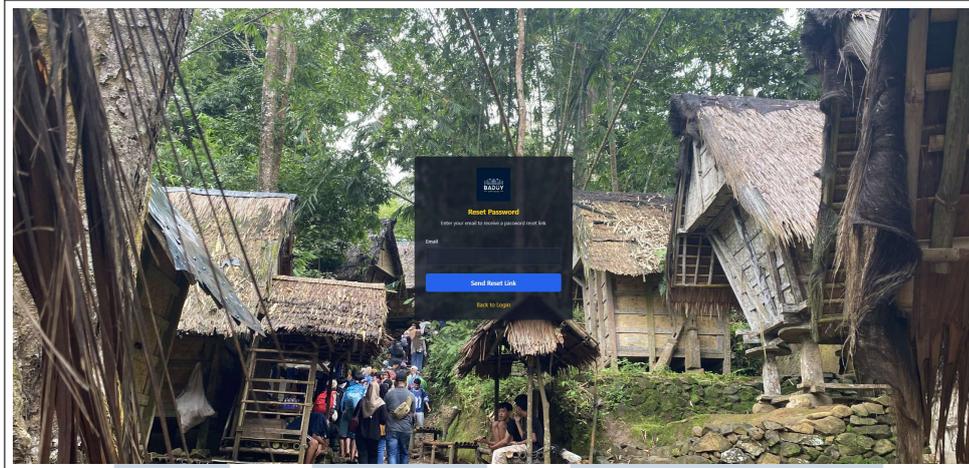


Gambar 3.18. Halaman *Login*

### C Halaman **Forgot Password**

Pada halaman forgot password, pengguna yang lupa kata sandi dapat memulai proses pemulihan akun mereka. Fitur ini dimulai dengan permintaan email address dari pengguna yang terdaftar. Setelah pengguna memasukkan alamat email mereka, sistem akan mengirimkan *reset link* yang berisi token untuk mengatur ulang kata sandi. *Token-based reset* digunakan untuk memastikan bahwa proses pemulihan hanya dapat dilakukan oleh pemilik akun yang sah. Setiap token yang dihasilkan bersifat *single-use* dan memiliki masa kedaluwarsa satu jam, menjamin keamanan proses reset kata sandi.

Selain itu, fitur ini juga mencakup *IP tracking* untuk mencatat dan memantau lokasi atau alamat IP dari permintaan reset, memberikan jejak audit keamanan jika terjadi aktivitas yang mencurigakan. Dengan cara ini, pengguna dapat dengan mudah mengatur ulang kata sandi mereka setelah memverifikasi identitas mereka melalui email. Sistem ini membantu menjaga keamanan akun dengan memastikan hanya pengguna yang sah yang dapat mengakses dan memulihkan akun mereka.

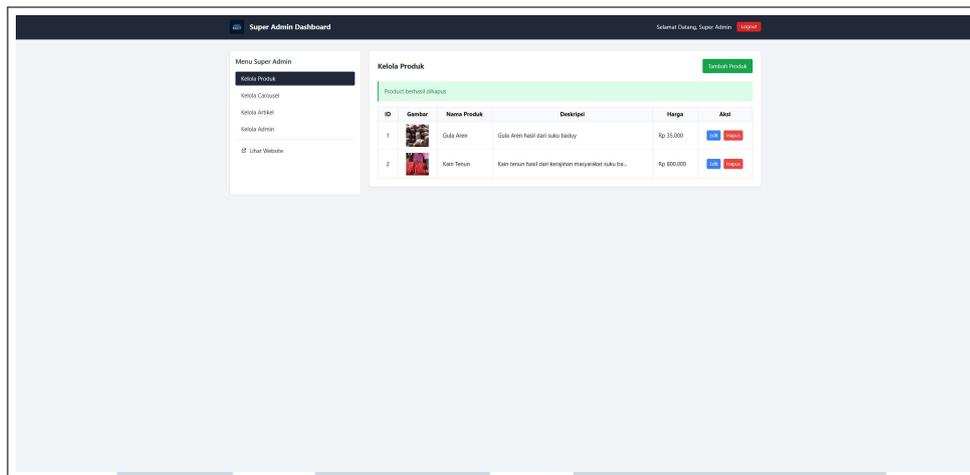


Gambar 3.19. Halaman *Forgot Password*

#### D Halaman Superadmin

Halaman superadmin memberikan akses penuh kepada pengguna dengan peran superadmin untuk mengelola seluruh aspek sistem, termasuk pengelolaan pengguna, produk, artikel, dan gambar yang ada pada website. Fitur *advanced search* memungkinkan superadmin untuk mencari pengguna, produk, atau artikel berdasarkan parameter tertentu seperti nama, email, dan peran, memudahkan dalam pencarian data. Selain itu, terdapat fitur *bulk actions*, yang memungkinkan superadmin untuk melakukan perubahan massal pada data, seperti menghapus atau memodifikasi beberapa entri secara bersamaan. Fitur *role management* memungkinkan superadmin untuk mengatur hak akses bagi admin dan pengguna lain di dalam sistem, sehingga memastikan kontrol penuh atas siapa yang dapat mengakses dan mengelola data tertentu.

Halaman ini juga dilengkapi dengan analitik pengguna, laporan performa sistem, dan laporan keamanan, yang memberikan informasi tentang aktivitas pengguna, tren pendaftaran, serta potensi ancaman atau masalah keamanan yang mungkin terjadi. Dengan fitur-fitur ini, superadmin dapat memantau dan mengelola seluruh sistem dengan lebih efektif. Selain itu, pengelolaan database dan laporan sistem memungkinkan superadmin untuk melakukan pemeliharaan data serta memastikan bahwa kinerja sistem tetap optimal.

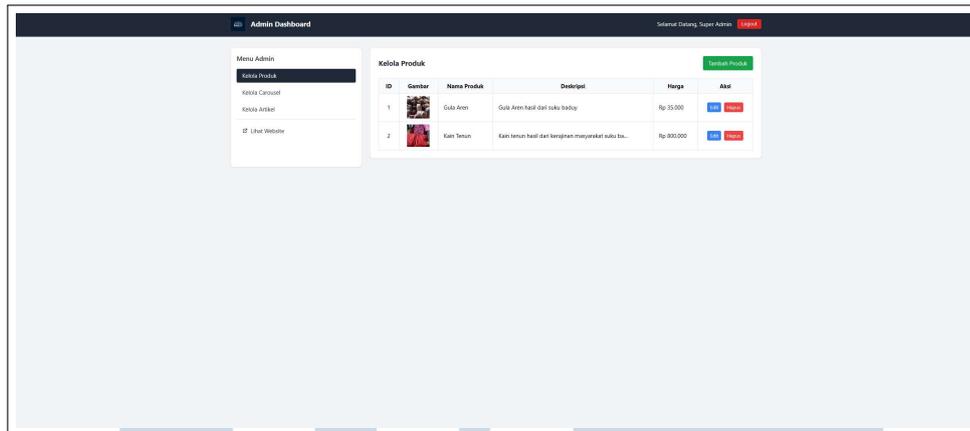


Gambar 3.20. Halaman *Superadmin*

## E Halaman Admin

Halaman admin memberikan akses terbatas dibandingkan dengan superadmin, namun tetap memungkinkan admin untuk mengelola produk, artikel, dan gambar yang ditampilkan di website. Admin dapat melakukan *bulk actions* untuk mempublikasikan, menolak, atau menghapus artikel secara massal. Halaman ini juga menyertakan *product management*, yang memungkinkan admin untuk mengedit detail produk seperti kategori, harga, dan deskripsi. Selain itu, fitur *carousel management* memungkinkan admin untuk mengelola gambar yang ditampilkan di homepage, termasuk mengatur urutan gambar dan memastikan gambar yang ditampilkan selalu relevan dan berkualitas.

Admin memiliki kendali penuh atas pengelolaan konten yang ditampilkan di website, namun akses mereka terbatas untuk pengelolaan pengguna atau pengaturan sistem lainnya yang hanya dapat dilakukan oleh superadmin. Dengan menggunakan tabel manajemen produk yang terstruktur dengan baik, admin dapat dengan mudah melacak inventaris, memperbarui produk, dan mengatur kategori produk yang sesuai. Fitur-fitur ini dirancang untuk meningkatkan efisiensi dalam pengelolaan konten, memastikan bahwa admin dapat melakukan tugas mereka dengan cepat dan akurat.



Gambar 3.21. Halaman Admin

### 3.3.6 Testing

Pada bagian ini, dilakukan pengujian terhadap berbagai fitur yang telah dikembangkan untuk memastikan fungsionalitas dan kualitas sistem. Pengujian ini mencakup berbagai aspek, mulai dari pengujian fungsionalitas *login*, pengelolaan produk, pengelolaan gambar, hingga pengelolaan artikel oleh *Admin* dan *Superadmin*. Tabel berikut menunjukkan hasil pengujian yang dilakukan selama proses pengembangan *website* profil Suku Baduy.

Rincian aktivitas selama pelaksanaan diuraikan pada tabel 3.2.

Tabel 3.2. Hasil Pengujian Fitur Website

No	Aktivitas Pengujian	Halaman	Hasil yang Diharapkan	Keterangan
1	Pengujian Login	Halaman Login	Pengguna dapat login dengan akun yang valid	Berhasil
2	Pengujian Register	Halaman Register	Pengguna dapat mendaftarkan akun baru	Berhasil
3	Pengujian Forgot Password	Halaman Forgot Password	Pengguna dapat mengatur ulang password melalui link	Berhasil
Lanjut pada halaman berikutnya				

No	Aktivitas Pengujian	Halaman	Hasil yang Diharapkan	Keterangan
4	Pengujian CRUD Produk	Halaman Admin (Produk), Halaman Superadmin	Admin dan Superadmin dapat menambah, mengedit, menghapus produk	Berhasil
5	Pengujian CRUD Carousel	Halaman Admin (Carousel), Halaman Superadmin	Admin dan Superadmin dapat menambah, mengedit, menghapus gambar carousel	Berhasil
6	Pengujian Upload Gambar (BLOB) Produk	Halaman Admin (Produk), Halaman Superadmin	Admin dan Superadmin dapat mengupload dan menampilkan gambar produk dengan benar	Berhasil
7	Pengujian Upload Gambar (BLOB) Carousel	Halaman Admin (Carousel), Halaman Superadmin	Admin dan Superadmin dapat mengupload dan menampilkan gambar carousel dengan benar	Berhasil
8	Pengujian Kelola Artikel	Halaman Admin (Artikel), Halaman Superadmin	Admin dan Superadmin dapat menambah, mengedit, menghapus artikel, dan persetujuan artikel	Berhasil

## 3.4 Kendala dan Solusi yang Ditemukan

### 3.4.1 Kendala

Selama pelaksanaan proyek ini, terdapat beberapa kendala yang dihadapi, antara lain:

- Saya sebelumnya telah mempelajari *Laravel* versi 9 dalam perkuliahan, namun pada proyek ini digunakan *Laravel* versi 11 yang memiliki sejumlah perbedaan struktur dan fitur. Perubahan ini, terutama dalam hal penggunaan *Eloquent ORM*, *Artisan CLI*, dan manajemen *routing* serta *middleware*, memerlukan proses penyesuaian yang cukup signifikan. Hal ini sempat memperlambat pengembangan karena saya harus memahami dokumentasi terbaru dan mencoba implementasi secara langsung.
- Metode penyimpanan gambar menggunakan *BLOB* dalam *database* merupakan hal yang baru bagi saya. Saya menghadapi tantangan teknis dalam integrasi antara proses unggah gambar dan penampilan data di sisi *frontend* dan *backend*, termasuk dalam memastikan format data gambar dapat tersimpan dan dibaca dengan benar.
- Komunikasi antar anggota tim pada tahap awal proyek belum berjalan secara optimal. Kurangnya koordinasi dalam pembagian tugas dan pelaporan *progres* menyebabkan beberapa pekerjaan menjadi terlambat atau tidak berjalan sesuai dengan rencana.

### 3.4.2 Solusi

Untuk mengatasi berbagai kendala tersebut, langkah-langkah berikut telah dilakukan:

- Mendalami dokumentasi resmi *Laravel* versi 11 dan membandingkannya dengan versi sebelumnya yang pernah saya pelajari, serta mengikuti tutorial *online* untuk memahami perubahan struktur dan fitur terbaru.
- Berdiskusi dengan anggota tim yang lebih berpengalaman untuk mendapatkan masukan praktis dalam mengimplementasikan fitur menggunakan *Laravel* versi terbaru.

- Mengakses forum dan komunitas pengembang seperti *Stack Overflow* dan *GitHub Discussions* untuk mencari solusi terhadap permasalahan teknis, khususnya terkait implementasi metode *BLOB*.
- Meningkatkan frekuensi rapat rutin secara *daring*, serta menerapkan dokumentasi tugas dan *progres* secara jelas dan terstruktur untuk memastikan setiap anggota tim memahami perannya dan perkembangan proyek.

