

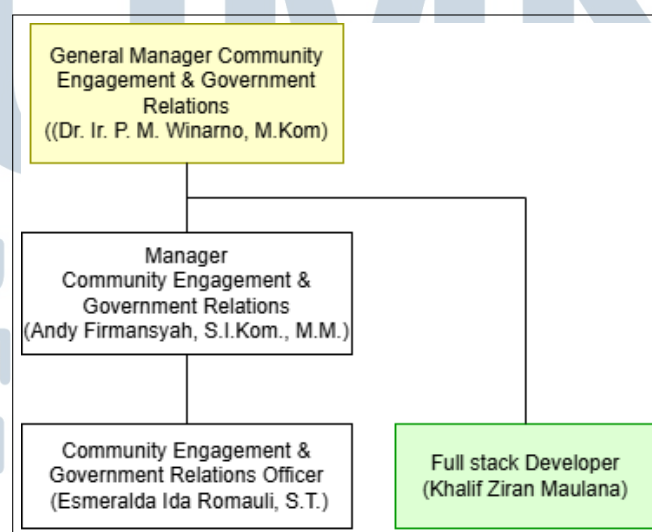
BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pembagian awal seluruh tim dilakukan bersama *supervisor* yang melibatkan enam peserta magang. Seluruh peserta dibagi menjadi dua tim berbeda, masing-masing terdiri dari tiga anggota, dengan proyek yang ditugaskan secara terpisah. Tim pertama mengerjakan pengembangan sistem koperasi untuk UMKM, sementara tim kedua bertanggung jawab membangun *website* untuk artikel dan penjualan produk Suku Baduy. Pada rapat perdana, *supervisor* magang, Bapak Winarno menjelaskan tujuan, ruang lingkup, dan ekspektasi dari kedua proyek tersebut harus dilakukan.

Tim proyek Suku Baduy, yang terdiri dari tiga anggota, difokuskan pada pengembangan *full stack* menggunakan *framework* Laravel. Setiap anggota tim diberi tanggung jawab utama sesuai spesialisasi satu orang mengelola desain antarmuka (*front end*), satu orang menangani logika bisnis dan *database* (*back end*), serta satu orang bertindak sebagai koordinator yang memastikan integrasi antar modul dan menangani desain serta sistem *database* (*full stack*). Koordinasi internal tim dilakukan melalui rapat tatap muka mingguan dan komunikasi mingguan via grup WhatsApp, sementara pelaporan progres disampaikan ke *supervisor* melalui pertemuan Zoom. Struktur kedudukan kerja dapat dilihat pada gambar 3.1



Gambar 3.1. Kedudukan struktur kerja

3.2 Tugas yang Dilakukan

Selama program magang di LPPM UMN, tanggung jawab yang diemban berfokus pada pengembangan sistem berbasis *web* untuk mendukung penjualan produk sekaligus mengenalkan Suku Baduy. Tugas yang dilakukan mencakup analisis kebutuhan, perancangan, hingga implementasi aplikasi yang dirancang untuk mempermudah pengelolaan secara terstruktur.

Adapun tugas yang di dapatkan dari hasil pembagian tugas mandiri secara tim adalah sebagai berikut:

- Analisis kebutuhan klien: Melakukan identifikasi dan pendalaman kebutuhan klien terkait aspek desain visual, termasuk preferensi bentuk, palet warna, tata letak, serta *fitur* fungsional yang diharapkan.
- Pengembangan halaman *blog* artikel: Merancang dan mengimplementasikan halaman *blog* khusus yang memuat konten bertema budaya Suku Baduy, meliputi artikel tentang adat istiadat, alat tradisional, cerita rakyat, dan kearifan lokal. Halaman dilengkapi *fitur* tambahan seperti sistem kategori artikel, pencarian berdasarkan kata kunci.
- Form pengisian artikel: Membangun modul formulir submission artikel yang memungkinkan pengguna terdaftar mengirimkan konten secara mandiri. Artikel yang dikirim akan melalui proses validasi oleh admin melalui sistem approval berbasis *role*. Konten baru hanya akan ditampilkan di halaman publik setelah mendapatkan persetujuan dari pihak admin.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

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

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Pengenalan tim serta pengarahan tugas dari <i>supervisor</i> dan pembuatan tugas yang diberikan.
2	Analisis dan perancangan awal platform <i>website</i> yang diinginkan.
3	Kunjungan pertama ke Suku Baduy bersama <i>Supervisor</i> dan tim. Pencatatan hasil yang di inginkan oleh klien.
4 - 5	Desain <i>sitemap</i> struktur navigasi aplikasi Baduy. Pembuatan <i>flowchart sign-in, sign-up</i> , tambah artikel, pembelian produk.
6 - 9	Pembuatan rancangan awal untuk <i>header</i> , halaman autentikasi, halaman artikel, dan halaman <i>dashboard</i> beserta repositori Github bersama. Membuat ERD dan set up <i>framework</i> Laravel beserta <i>database</i> menggunakan Mysql/Mariadb
10 - 12	Penyelesaian awal desain antar muka artikel, detail artikel, tambah artikel. Menyambungkan pemanggilan data dari <i>database</i> ke artikel dengan data <i>dummy</i> hingga berhasil, Penambahan beberapa <i>dependency</i> untuk <i>framework</i> .
13 - 14	Mengganti kode bagian artikel untuk dipindahkan ke Tailwind dari css, penambahan <i>fitur</i> minor pada fungsi artikel dan detail artikel, melanjutkan penyambungan relasi <i>database</i> .
15 - 16	Memperbaiki <i>error</i> minor pada beberapa <i>fitur</i> sekaligus <i>testing</i> pada <i>server</i> lokal, penggabungan file artikel dengan gabungan <i>web</i> Baduy milik rekan kelompok.
17	Perbaikan <i>error</i> minor jika masih terjadi, perubahan bentuk penerimaan gambar ke <i>database</i> , <i>hosting</i> produk dan demo aplikasi ke klien

3.3 Uraian Pelaksanaan Magang

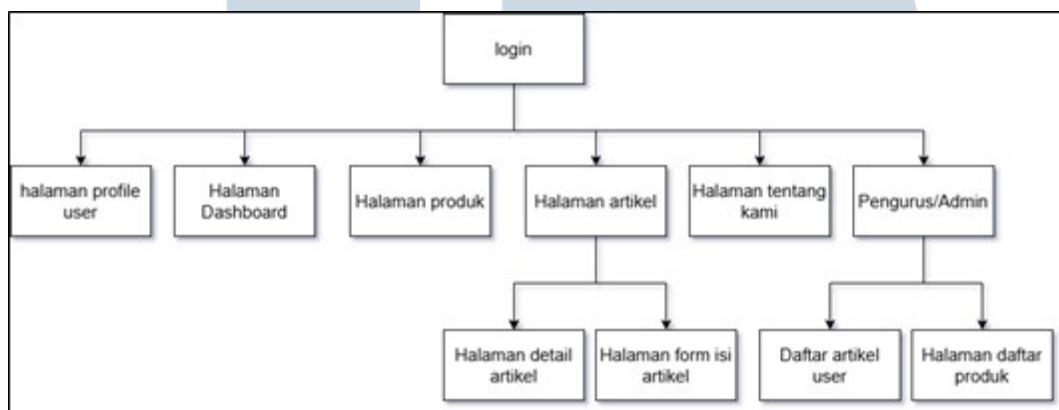
Dalam pelaksanaan pengembangan *website* Suku Baduy diawali dengan rapat koordinasi perdana yang dipimpin oleh *supervisor* magang. Setelah itu disampaikan gambaran umum mengenai proyek-proyek pengembangan sistem yang akan dikerjakan selama periode magang. Seluruh peserta magang yang telah dibagi menjadi dua tim kerja berdiskusi tentang kebutuhan pembuatan sistemnya, yang akan

dibahas lagi setelah bertemu klien.

3.3.1 Perancangan Website

A Sitemap

Sitemap adalah salah satu alat bantu yang berguna bagi pengunjung situs untuk melakukan navigasi di dalam sebuah *website*



Gambar 3.2. Tampilan struktur *sitemap*

Gambar 3.2. mengilustrasikan arsitektur navigasi sistem *website* budaya dan produk Suku Baduy. Struktur ini dirancang untuk memfasilitasi dua jenis pengguna utama yaitu *end-user* (pengunjung atau pembeli) dan *administrator*.

Akses awal sistem diawali melalui halaman *Login*, yang membedakan hak akses berdasarkan peran pengguna. Setelah *autentikasi*, pengguna diarahkan ke Halaman *Dashboard* sebagai pusat kontrol untuk menjelajahi seluruh *fitur* inti. Pada *dashboard*, tersedia akses langsung ke Halaman Produk yang memuat katalog komoditas unggulan Suku Baduy dilengkapi informasi detail seperti asal-usul produk, dan teknik pembuatan tradisional.

Sementara itu, Halaman Artikel berfungsi sebagai *repository* digital yang memberikan konten edukatif meliputi ritual adat, filosofi kehidupan masyarakat, serta dokumentasi kearifan lokal. Pengguna berkontribusi dalam memperbanyak konten melalui halaman yang disediakan untuk mengunggah artikel, di mana *submisi* artikel akan tercatat pada daftar artikel pengguna sebagai ruang penyimpanan sementara sebelum dipublikasikan.

Di sisi manajemen, pengurus atau *administrator* memiliki otoritas untuk mengawasi seluruh aktivitas sistem melalui modul khusus. *Fitur* ini mencakup

validasi konten artikel di halaman detail artikel, pengaturan inventaris produk di halaman daftar produk untuk melakukan CRUD, serta pemantauan akun pengguna. Adapun halaman *profile* pengguna memungkinkan personalisasi data pengguna, sedangkan halaman tentang kami menyajikan narasi historis dan tujuan pengembangan platform sebagai media pelestarian budaya.

Struktur ini menekankan segregasi peran yang ketat, di mana interaksi pengguna umum terbatas pada eksplorasi konten dan transaksi, sementara *administrator* bertugas menjaga integritas data serta kelancaran operasional sistem.

B Flowchart

Flowchart merupakan diagram alur yang merepresentasikan langkah-langkah proses sistem secara visual menggunakan simbol standar[7]. Pada pengembangan *website* ini, *flowchart* dibuat untuk memetakan alur interaksi antara pengguna *user* atau *administrator* dengan sistem secara intuitif tanpa menggunakan istilah teknis rumit.

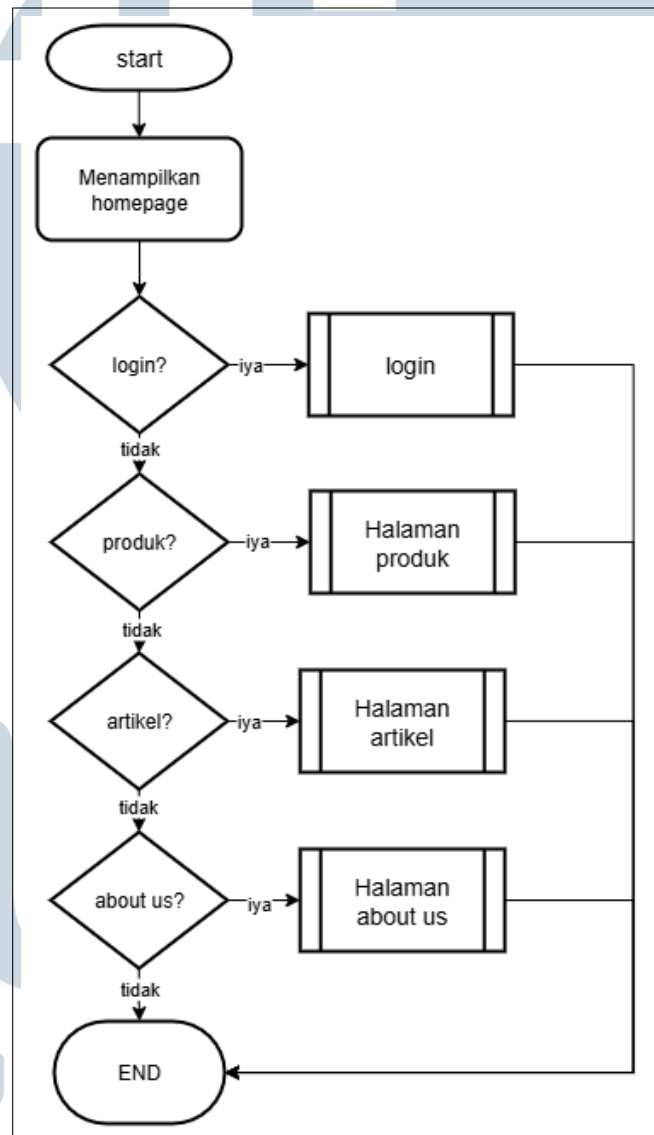
Pada sistem awal ini menggambarkan alur logika yang terjadi sejak pengguna mengakses *homepage* hingga sistem menampilkan halaman yang diinginkan atau berakhir. Proses dimulai dari pengguna membuka aplikasi dan sistem menampilkan *homepage*. Selanjutnya, sistem melakukan pengecekan terhadap permintaan pengguna, dimulai dengan menanyakan apakah pengguna ingin melakukan *login*. Jika pengguna memilih *login*, maka akan diarahkan langsung ke halaman *login*.

Apabila pengguna tidak memilih *login*, pengguna bisa melanjutkan dengan fitur berikutnya, yaitu apakah pengguna ingin mengakses halaman produk. Jika iya, pengguna akan diarahkan ke halaman produk. Jika tidak, sistem kembali melakukan pengecekan terhadap permintaan akses ke halaman artikel. Jika pengguna memilih untuk membuka artikel, maka sistem akan menampilkan halaman artikel.

Jika permintaan terhadap halaman artikel juga tidak ada, pengguna dapat melanjutkan ke fitur terakhir, yaitu apakah pengguna ingin mengakses halaman *about us*. Apabila pengguna menginginkan informasi seputar warga Suku Baduy, sistem akan menampilkan halaman *about us*. Namun, jika tidak ada permintaan apa pun dari pengguna pada tahapan-tahapan tersebut, maka sistem akan mengakhiri proses dan tidak melakukan operasi lanjutan.

Dengan alur ini, *flowchart* menggambarkan proses pengambilan keputusan secara berurutan berdasarkan permintaan pengguna, sehingga setiap pengguna akan

diarahkan secara tepat ke halaman yang diinginkan dengan efisien dan sistematis. Alur ini juga memudahkan dalam proses pemeliharaan dan pengembangan sistem di masa mendatang karena setiap kemungkinan permintaan telah diatur dengan logika yang jelas dan terstruktur. Struktur *flowchart* bagi pengguna yang belum *login* dapat dilihat pada gambar 3.3



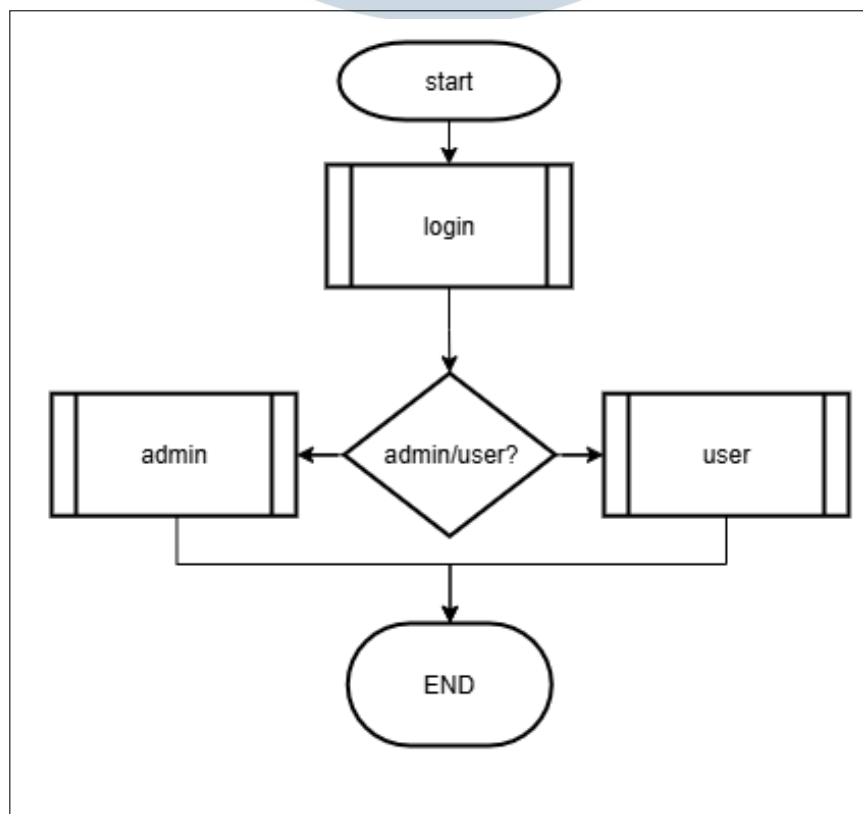
Gambar 3.3. *flowchart* awal sebelum *login*

Selanjutnya untuk *flowchart login* menggambarkan proses *autentikasi* pengguna serta pembagian peran (*role*) dalam sistem. Alur dimulai dengan tahap *start* yang menandakan inisiasi proses. Pengguna kemudian diarahkan ke halaman *login* untuk memasukkan kredensial seperti email dan *password*. Setelah berhasil

login, sistem akan melakukan verifikasi *role* pengguna untuk menentukan hak akses yang dimiliki.

Terdapat dua kemungkinan *role* yang diperiksa, yaitu *admin* dan *user* biasa. Jika pengguna teridentifikasi sebagai *admin*, sistem akan memberikan akses penuh ke fitur-fitur administratif yang telah ditentukan. Sebaliknya, jika pengguna terverifikasi sebagai *user* biasa, sistem akan membatasi akses hanya ke fitur-fitur yang diperuntukkan bagi pengguna umum. Proses ini memastikan bahwa setiap pengguna hanya dapat mengakses bagian sistem yang sesuai dengan peran mereka, sehingga menjaga keamanan dan integritas data. Setelah *role* berhasil ditentukan, alur proses berakhir, menandakan bahwa *autentikasi* dan pembagian hak akses telah selesai dilakukan.

Secara keseluruhan, flowchart ini menggambarkan mekanisme *autentikasi* yang efisien dan sistematis, dengan penekanan pada pentingnya pembagian *role* untuk mengontrol akses pengguna. Alur tersebut juga menunjukkan bagaimana sistem merespons keberhasilan *login* dengan menyesuaikan tampilan dan fungsi berdasarkan peran pengguna, sehingga memastikan operasi sistem yang terorganisir dan aman. Struktur *flowchart login* sesuai *role* dapat dilihat pada gambar 3.4



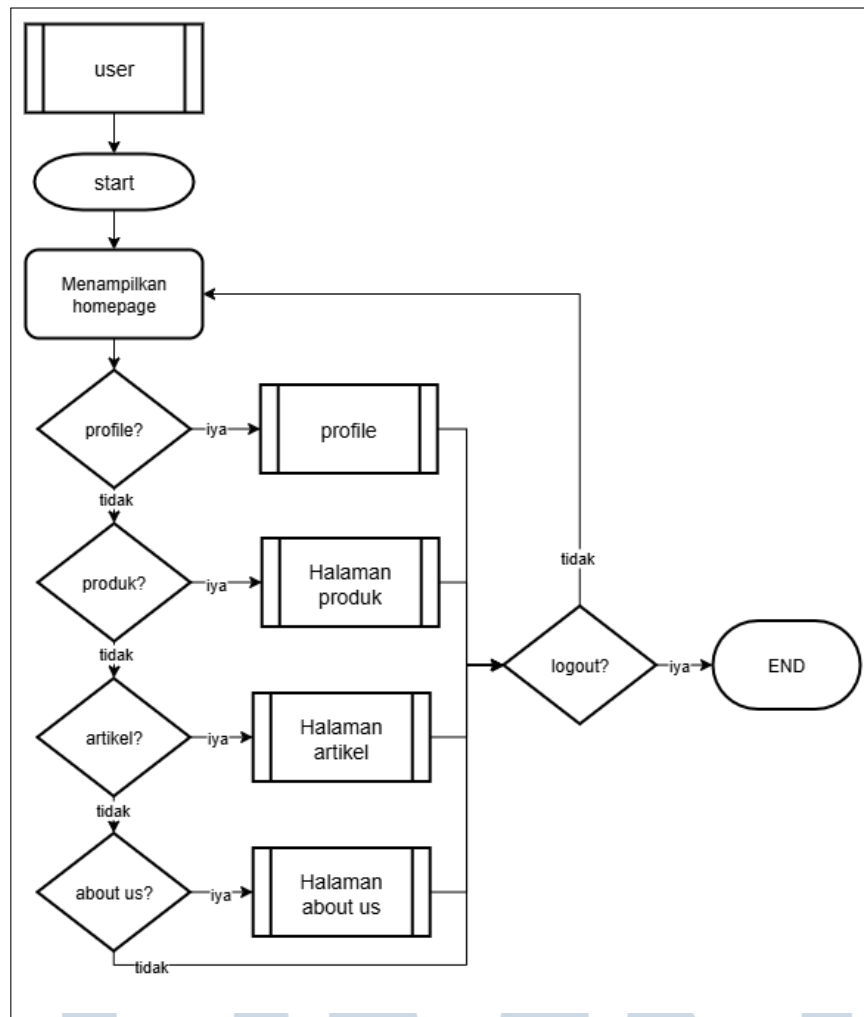
Gambar 3.4. *flowchart login* sesuai dengan *role*

Berikutnya pada sistem ini menggambarkan alur navigasi yang tersedia bagi pengguna dengan *role user* setelah berhasil melakukan *login* ke dalam sistem. Proses diawali dengan menampilkan *homepage* sebagai antarmuka utama yang memuat berbagai pilihan *menu navigasi*. Sistem kemudian memberikan beberapa pilihan akses kepada *user* sesuai dengan hak akses yang dimilikinya.

User pertama kali diberikan pilihan untuk mengakses halaman *profile* yang berisi informasi pribadi dan pengaturan akun. Jika *user* memilih untuk tidak mengakses *profile*, sistem akan menawarkan alternatif navigasi ke halaman produk yang menampilkan berbagai produk yang tersedia. Apabila *user* kembali memilih untuk tidak mengakses produk, sistem akan menyediakan opsi ketiga berupa halaman artikel yang berisi konten informatif. Jika opsi artikel pun tidak dipilih, *user* dapat memilih untuk mengakses halaman *about us* yang berisi informasi mengenai warga Suku Baduy.

Jika *user* tidak memilih satupun dari opsi yang tersedia, proses akan berlanjut ke tahap berikutnya tanpa perubahan tampilan hingga seluruh opsi telah ditawarkan. Alur ini dirancang untuk memberikan pengalaman navigasi yang mudah dan terstruktur bagi *user* biasa, sekaligus membatasi akses hanya pada fitur-fitur yang sesuai dengan hak akses yang dimilikinya. Struktur *flowchart* user yang telah login dapat dilihat pada gambar 3.5





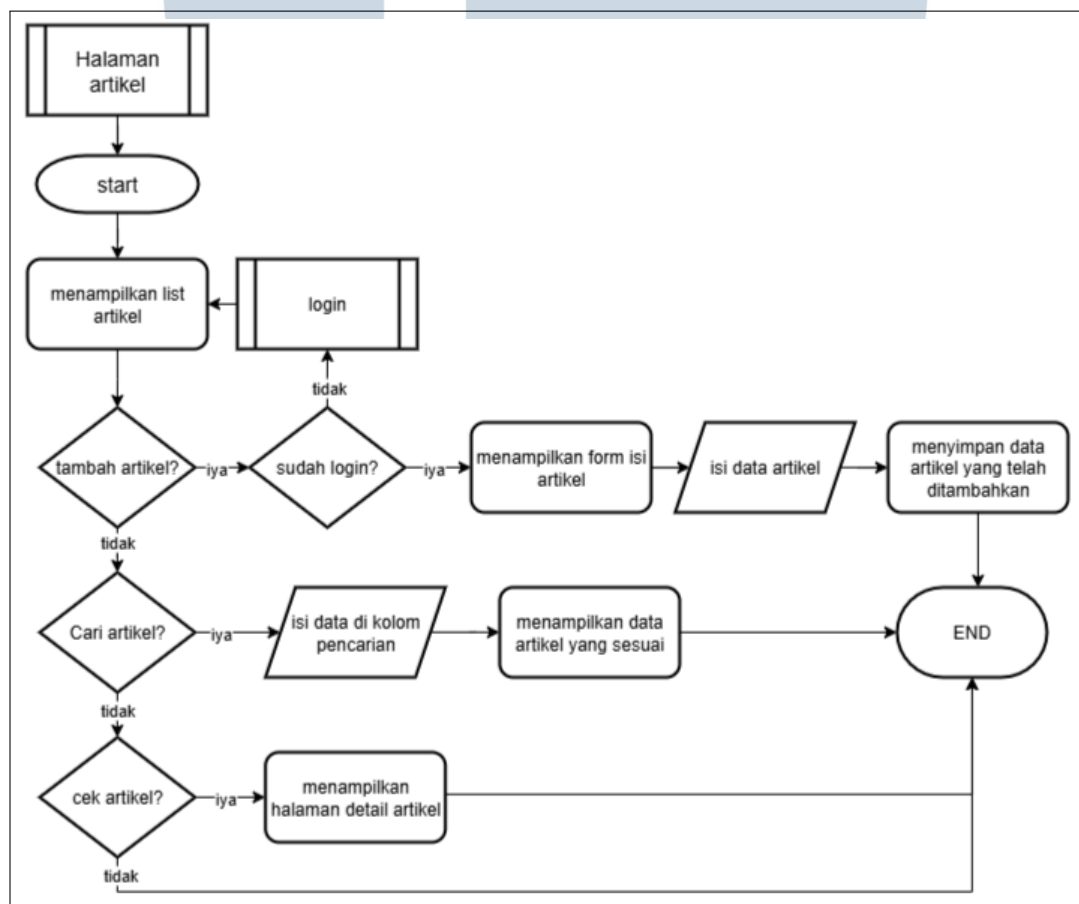
Gambar 3.5. *Flowchart* fitur user yang telah login

Pada *flowchart* berikutnya merupakan sistem yang menggambarkan proses alur kerja sistem untuk pengelolaan artikel, dengan fokus pada aksi-aksi seperti menambah, mencari, dan melihat detail artikel di sebuah aplikasi atau sistem informasi.

Proses diawali dari halaman utama artikel, dilanjutkan dengan menampilkan daftar artikel yang tersedia di sistem. Ketika pengguna ingin menambah artikel, sistem terlebih dahulu memeriksa apakah pengguna sudah *login*. Jika pengguna belum melakukan *login*, sistem akan mengarahkan ke halaman *login* sebelum mengizinkan langkah selanjutnya. Setelah berhasil *login*, sistem akan menampilkan *form* untuk pengisian data artikel baru. Pengguna kemudian mengisi data yang diperlukan pada *form* tersebut. Jika data sudah lengkap, sistem akan menyimpan data artikel yang sudah dimasukkan dan proses berakhir.

Jika pengguna tidak ingin menambah artikel, *flowchart* lanjut pada proses pencarian artikel. Sistem akan meminta pengguna untuk mengisi data pada kolom pencarian. Setelah pengguna mengisi data pencarian, sistem akan mencocokkan dan menampilkan data artikel yang sesuai dengan kata kunci yang dimasukkan oleh pengguna.

Selain itu, pengguna juga bisa memilih untuk memeriksa detail suatu artikel. Jika pengguna melakukan pengecekan artikel, sistem akan menampilkan halaman detail artikel yang berisi informasi lengkap dari artikel tersebut. Apabila tidak ada aksi penambahan, pencarian, maupun pengecekan artikel, maka proses akan diakhiri. Struktur *flowchart* halaman artikel dapat dilihat pada gambar 3.6



Gambar 3.6. *Flowchart* halaman artikel

Selanjutnya *flowchart* ini secara sistematis menggambarkan proses pengelolaan *profile* dan konten pribadi yang dilakukan oleh pengguna di dalam sistem informasi artikel. Proses dimulai dari tahap di mana pengguna mengakses halaman *edit profile*. Pada tahap awal tersebut, sistem membuka kemungkinan bagi

pengguna untuk melakukan penyesuaian terhadap berbagai data pribadi, sehingga memperkuat aspek personalisasi pada akun masing-masing.

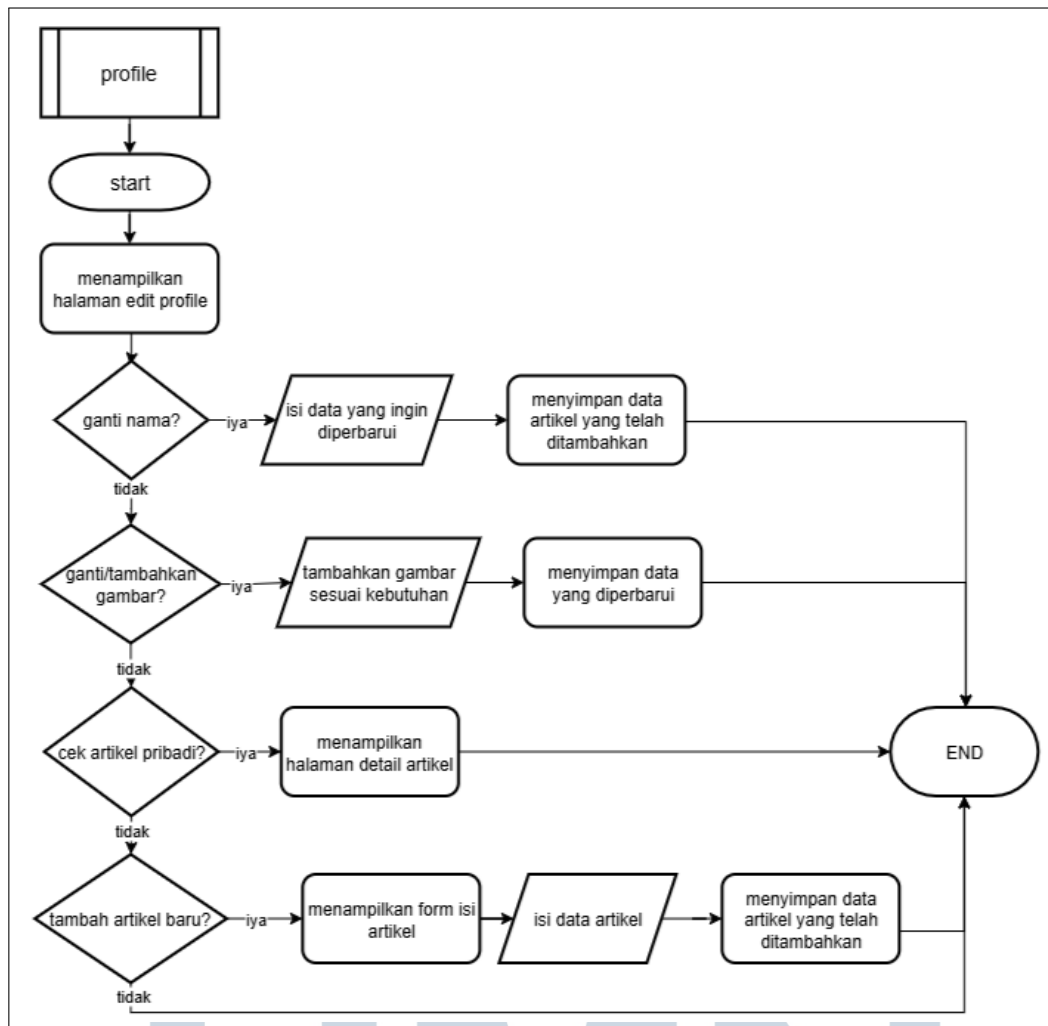
Pengguna akan terlebih dahulu dihadapkan pada pertanyaan terkait perubahan nama. Jika pengguna memilih untuk mengganti nama, sistem akan melaksanakan proses pembaruan dan memastikan data terbaru tersimpan dengan baik. Apabila tidak terdapat keinginan untuk mengubah nama, sistem secara otomatis menawarkan pilihan berikutnya, yakni perubahan atau penambahan gambar *profile*. Tahapan ini bertujuan untuk memberikan kemudahan bagi pengguna dalam mengelola identitas tampilan mereka di aplikasi.

Setelah langkah-langkah pada data *profile*, sistem memberikan opsi selanjutnya yang berkaitan dengan manajemen konten artikel. Pengguna dapat memilih untuk melihat daftar artikel pribadi yang sebelumnya telah dibuat. Apabila opsi ini dipilih, sistem akan menampilkan halaman rincian yang memuat informasi lengkap tentang artikel-artikel tersebut. Tidak hanya itu, pengguna juga diberikan kesempatan untuk menambah artikel baru melalui *form input* yang disediakan secara langsung setelah pilihan diambil.

Seluruh data yang mengalami perubahan, baik berupa *profile* maupun konten artikel, akan diproses dan disimpan di sistem secara terpisah guna menjamin keamanan dan integritas data. Setiap aktivitas yang dilakukan berujung pada penyimpanan data sesuai jenis perubahan, sehingga riwayat serta konsistensi informasi dapat tercatat dengan optimal.

Secara keseluruhan, *flowchart* ini mendemonstrasikan pola kerja sistem yang terstruktur, efisien, dan memberikan pengalaman pengguna yang intuitif. Setiap pilihan dan aksi pengguna diakomodasi melalui tahapan-tahapan terpisah yang jelas, memudahkan sekaligus memberikan rasa kontrol penuh atas data pribadi dan konten yang dimiliki di dalam sistem. Struktur *flowchart* untuk *edit profile* dapat dilihat pada gambar 3.7

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



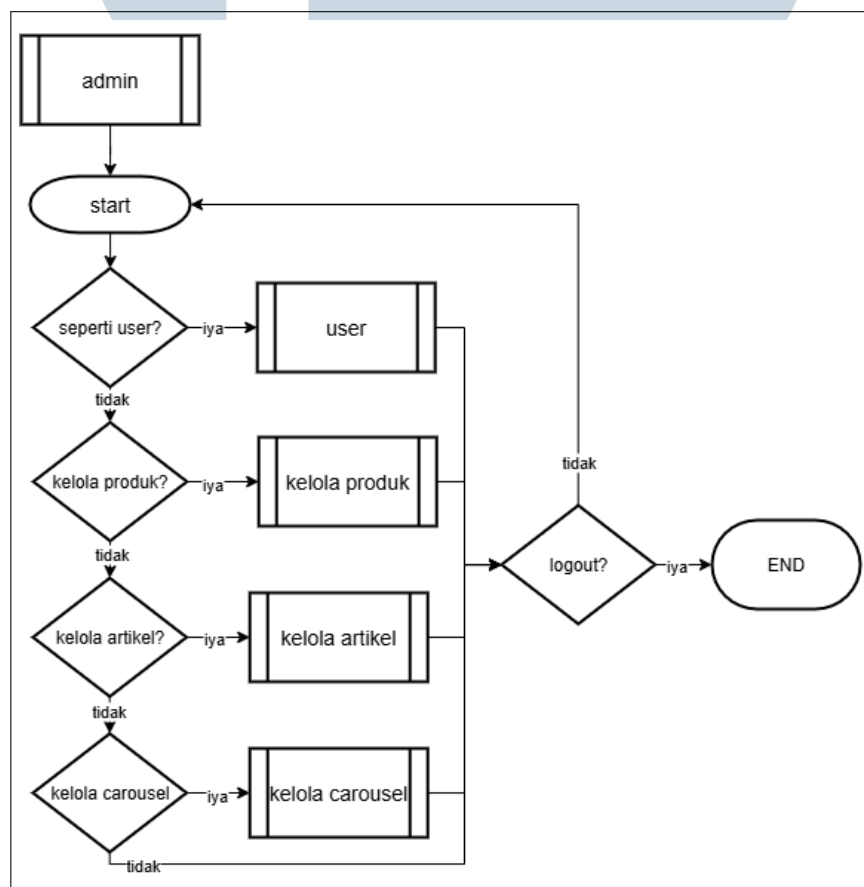
Gambar 3.7. Flowchart edit profile

Pada flowchart berikut ini memberikan proses terperinci mengenai alur kerja dan pembagian hak akses yang dimiliki oleh *admin* di dalam sistem. Alur dimulai sejak *admin* berhasil melakukan *autentikasi* melalui proses *login*. Setelah berhasil masuk, sistem terlebih dahulu menentukan apakah *admin* ingin beroperasi dengan hak akses standar seperti pengguna biasa (*user*). Jika opsi ini dipilih, maka sistem secara otomatis mengalihkan *admin* ke antarmuka dan fitur yang identik dengan peran *user* biasa, sehingga *admin* dapat mengakses fungsi-fungsi dasar sebagaimana pengguna pada umumnya.

Sebaliknya, jika *admin* memilih untuk tidak menggunakan fitur *user* biasa, sistem langsung menampilkan menu manajemen khusus yang hanya dapat diakses oleh *admin*. Menu pertama yang muncul adalah kelola produk, yang berfungsi untuk memberikan hak penuh kepada *admin* dalam melakukan aksi manipulasi

produk, seperti menambah, mengubah, dan menghapus data produk dari sistem. Apabila menu kelola produk dilewati, sistem menawarkan opsi kelola artikel, yang memungkinkan admin untuk melakukan pengelolaan konten berupa publikasi, penolakan, dan *review* artikel sesuai kebutuhan. Jika kedua opsi tersebut juga tidak dipilih, sistem selanjutnya menghadirkan menu kelola *carousel*. Melalui fitur ini, admin dapat menyesuaikan tampilan *slider* dan *banner* utama yang menjadi bagian penting dari antarmuka aplikasi.

Setiap menu manajemen yang dapat dipilih *admin* menggambarkan penerapan prinsip hak akses yang selektif, di mana fitur-fitur administratif benar-benar eksklusif hanya untuk *admin*. Proses berakhir jika *admin* memutuskan untuk tidak melakukan satupun tindakan administratif, menandakan adanya kebebasan navigasi di dalam sistem sesuai kebutuhan pengelolaan. Struktur *flowchart* untuk fitur apa saja yang dapat dilakukan admin dapat dilihat pada gambar 3.8



Gambar 3.8. *Flowchart* fitur yang dimiliki admin

Untuk proses *flowchart* kelola artikel oleh *admin* ini memetakan secara sistematis dan terstruktur seluruh tahapan yang tersedia dalam proses validasi

konten oleh *admin*. Proses diawali dengan *admin* memasuki menu utama pengelolaan artikel, di mana sistem langsung menawarkan berbagai opsi tindakan yang dapat dilakukan *admin*.

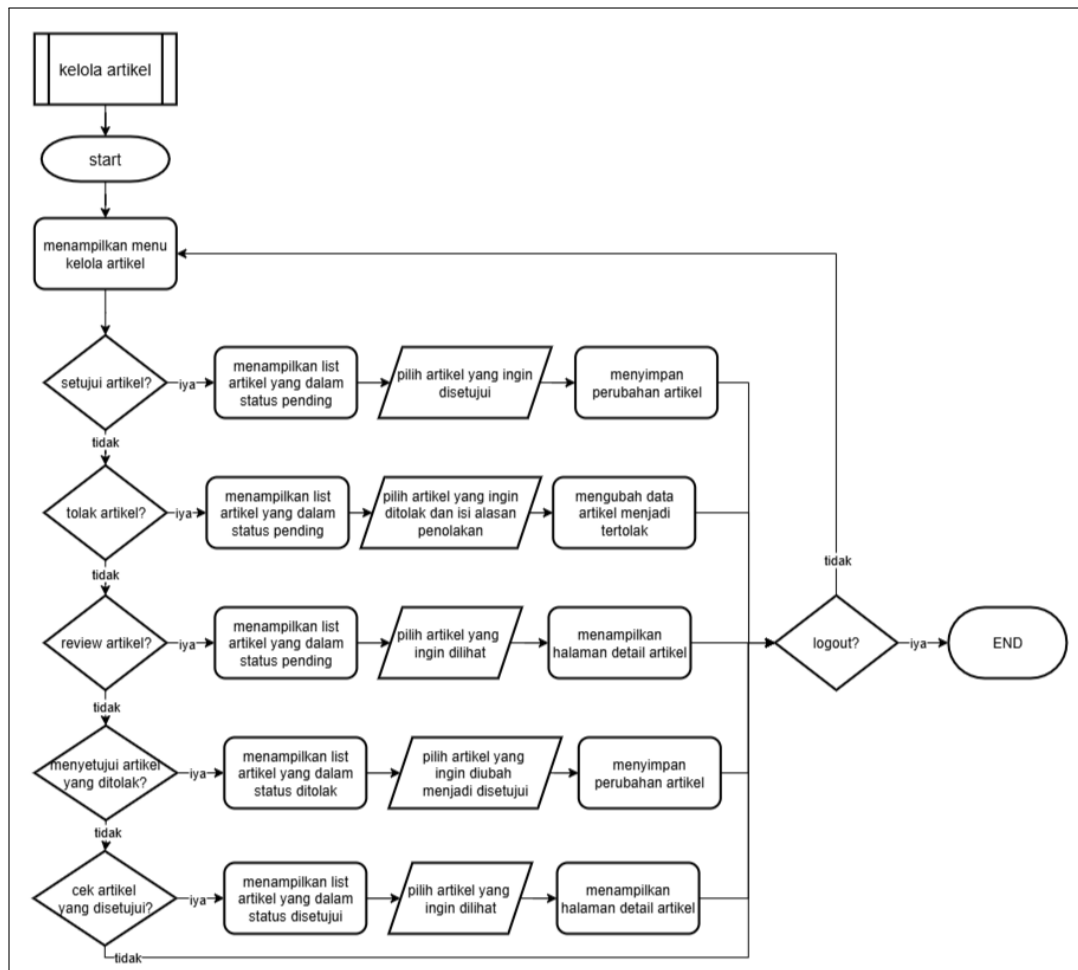
Tahapan pertama adalah persetujuan artikel apabila menu ini dipilih, sistem akan menampilkan daftar artikel berstatus *pending* yang menunggu verifikasi. *Admin* dapat menelaah daftar tersebut, memilih artikel tertentu, dan melanjutkan pengubahan status ke "disetujui" setelah proses seleksi dilakukan secara seksama. Seluruh perubahan status akan langsung tercatat oleh sistem, memastikan setiap aksi terdata secara tepat.

Jika *admin* tidak memilih untuk melakukan *approval*, sistem menyediakan opsi penolakan artikel. Pada proses ini, *admin* kembali diberikan daftar artikel yang masih dalam status *pending*. Setelah memilih artikel yang ingin ditolak, *admin* diwajibkan mengisi alasan penolakan guna menjaga rekam jejak keputusan. Setelah alasan diisi, status artikel otomatis diperbarui menjadi "ditolak", dan sistem mencatat proses tersebut secara rinci.

Selain dua aksi utama tersebut, *flowchart* juga mengakomodasi kebutuhan *admin* yang hanya ingin melihat detail artikel yang masih dalam status *pending* tanpa melakukan perubahan status. Pada pilihan ini, sistem akan menampilkan halaman rincian dari artikel yang dipilih, tanpa memengaruhi status atau menimbulkan konsekuensi lebih lanjut.

Untuk kasus artikel yang sebelumnya telah ditolak, tersedia pula mekanisme perubahan status. *Admin* diberikan kebebasan untuk melakukan peninjauan ulang jika diperlukan, mengubah status artikel dari "ditolak" menjadi "disetujui" sebagai bentuk tindak lanjut dari proses perubahan. *Flowchart* ini juga mengakomodasi *admin* yang ingin melakukan peninjauan terhadap artikel berstatus "disetujui", di mana fitur detail artikel tetap dapat diakses tanpa kompromi.

Setiap proses dalam *flowchart* ini menegaskan adanya validasi bertahap dan terdokumentasi, yang diperlukan untuk memastikan transparansi, konsistensi, serta kejelasan alur persetujuan konten dalam sistem. Rancangannya memungkinkan *admin* dapat melakukan tindakan melihat detail ataupun mengambil keputusan lanjutan berdasar hasil review, semuanya dalam satu rangkaian kerja yang fleksibel namun tetap terkoordinasi dan terdokumentasi. Struktur *flowchart* kelola artikel untuk *admin* dapat dilihat pada gambar 3.9



Gambar 3.9. Flowchart kelola artikel

C Entity Relation Diagram

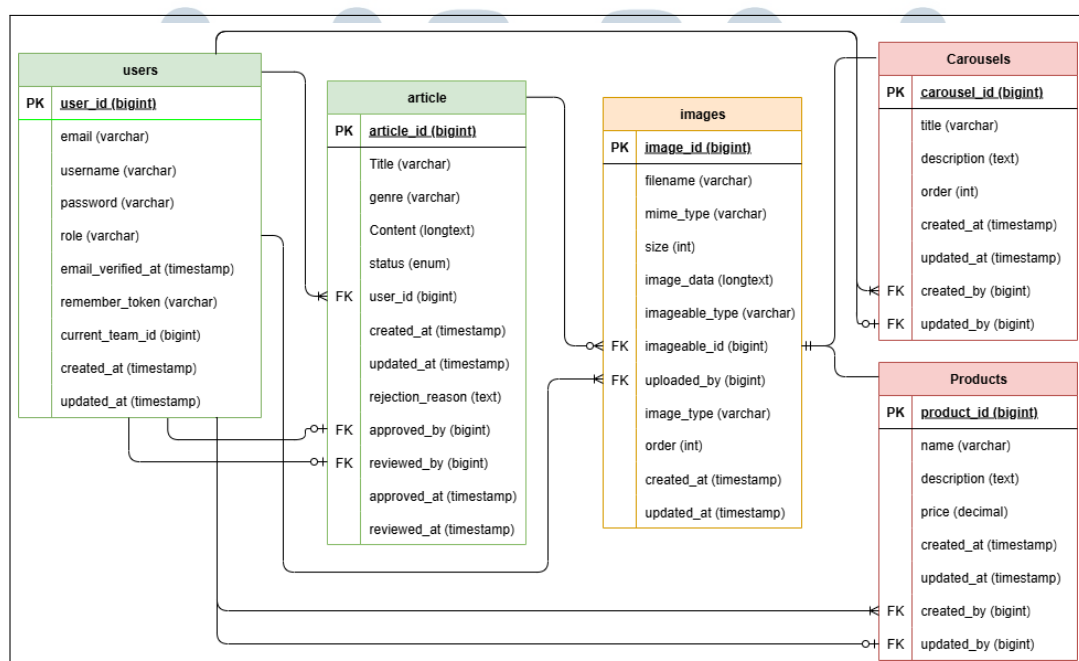
Entity Relationship Diagram (ERD) yang dirancang untuk sistem Website Baduy ini terdiri atas lima entitas utama, yaitu *Users*, *Articles*, *Images*, *Carousels*, dan *Products*. Untuk bagian website yang berfokus pada fungsi informasi dan edukasi terdapat pada entitas *Users*, *Articles*, dan *Images* saja. Tabel *Users* berfungsi sebagai inti sistem dengan "primary key id" yang menjadi pengidentifikasi unik setiap pengguna. Atribut-atribut seperti email, *username*, *password*, dan *role* digunakan untuk keperluan *autentikasi* dan otorisasi, sementara "email verified at" dan "remember token" mendukung proses verifikasi email dan manajemen sesi.

Tabel *Articles* dirancang untuk menyimpan konten terkait budaya Baduy, dengan relasi wajib *one-to-many* ke *Users* melalui "foreign key user id",

menunjukkan bahwa satu pengguna dapat membuat banyak artikel. Proses kurasi konten diimplementasikan melalui relasi opsional *one-to-one* dengan *Users* pada kolom "*approved by*" dan "*reviewed by*", memungkinkan artikel berada dalam status "*pending*" sebelum disetujui atau ditolak oleh admin. Atribut seperti *title*, *genre*, *content*, dan status mengatur isi tampilan dari sebuah artikel, sedangkan "*rejection reason*" memberikan transparansi dalam proses penolakan sebuah artikel ke pengguna.

Pada tabel *Images* menggunakan pola *polymorphic* untuk menghubungkan gambar ke berbagai tabel lain (*Articles*, *Carousels*, *Products*) melalui kombinasi "*imageable type*" dan "*imageable id*", memberikan fleksibilitas dalam manajemen media. Setiap gambar wajib terasosiasi dengan pengguna yang menambahkan melalui "*foreign key uploaded*" by dalam relasi *one-to-many*.

Seluruh tabel yang diwarnai hijau merupakan bagian tabel yang berfokus pada fungsi informasi edukasi untuk *website*, sedangkan untuk tabel yang diwarnai merah berfokus pada fungsi pemasaran produk pada *website*. Untuk tabel yang diwarnai kuning digunakan untuk campuran antara fungsi artikel sekaligus penjualan produk. Diagram ini menggambarkan arsitektur database yang mendukung operasional website secara lengkap mulai dari manajemen konten, media, hingga produk. Tampilan *Entity Relationship Diagram Web Baduy* terlampir pada gambar 3.10



Gambar 3.10. *Entity Relationship Diagram web Baduy*

D Desain *Wireframe*

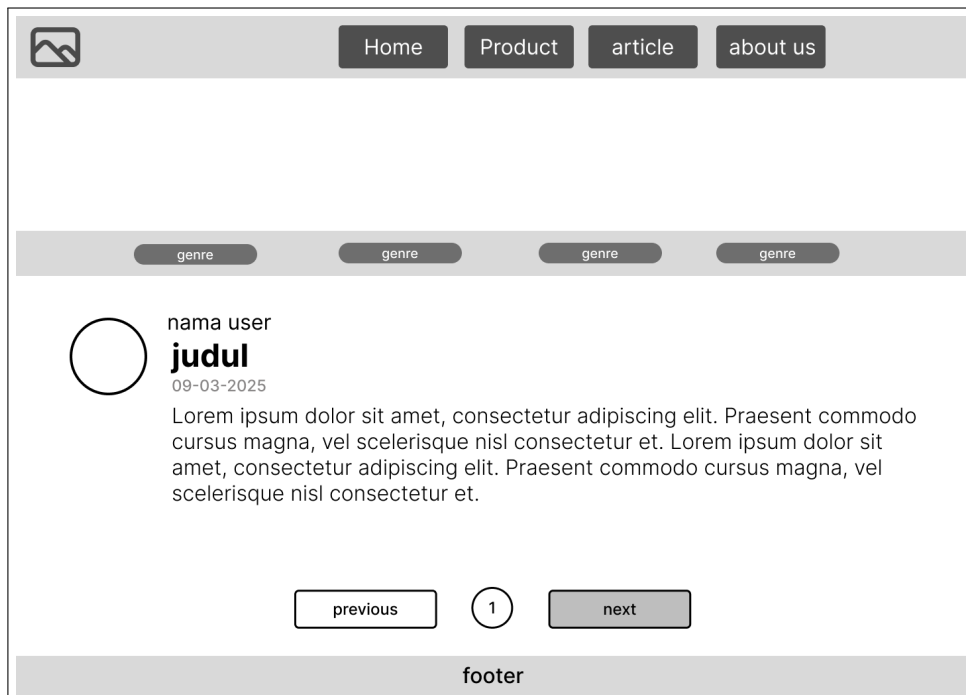
Sebagai fondasi dalam proses desain antarmuka, *wireframe* berperan penting dalam memvisualisasikan struktur dasar dan tata letak halaman sebelum masuk ke tahap pengembangan lebih lanjut. *Wireframe* tidak hanya menjadi panduan bagi tim desain dan pengembang, tetapi juga membantu dalam mengevaluasi alur navigasi serta pengalaman pengguna secara keseluruhan.

Pada halaman artikel menunjukkan tata letak dasar yang dirancang untuk menampilkan konten artikel dengan struktur yang jelas dan mudah dinavigasi. Pada bagian atas, terdapat menu navigasi utama yang mencakup opsi *Home*, *Product*, *Article*, dan *About Us*, memungkinkan pengguna untuk beralih antarhalaman dengan mudah. Di bawahnya, terdapat kategori artikel yang dibagi berdasarkan *genre* untuk membantu pengguna menemukan konten sesuai minat mereka.

Bagian utama halaman menampilkan konten artikel, dimulai dengan informasi penulis, judul artikel, dan tanggal publikasi. Di bawahnya, terdapat teks artikel yang nantinya akan diisi dengan konten aktual. Pada bagian bawah artikel, disediakan navigasi halaman untuk mengakses artikel lain, serta *footer* yang melengkapi struktur halaman. Desain ini menekankan kesederhanaan dan fungsionalitas, memastikan pengalaman membaca yang nyaman bagi pengguna.

Wireframe ini juga memberikan hierarki visual yang jelas untuk menuntun pembaca melalui konten. Kategori *genre* yang ditampilkan di bagian atas membantu dalam organisasi konten, sementara navigasi halaman memudahkan eksplorasi lebih lanjut. Tata letak ini dirancang untuk memastikan konsistensi dengan bagian lain dari *website*, sekaligus menyediakan ruang bagi elemen-elemen interaktif yang dapat dikembangkan lebih lanjut. Tampilan *wireframe* artikel terlampir pada gambar 3.11

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



Gambar 3.11. tampilan desain kasar artikel

Wireframe halaman tambah artikel dirancang dengan pendekatan fungsional dan minimalis untuk memastikan kemudahan penggunaan. Antarmuka ini mempertahankan konsistensi navigasi melalui menu utama di bagian atas, diikuti oleh judul halaman "Tambah Artikel" sebagai penanda konteks yang jelas. Formulir *input* disusun secara linear dengan tiga komponen utama *field* judul artikel, *selector genre*/kategori, dan area konten yang luas. Desain ini mengutamakan hierarki visual yang terstruktur dengan *whitespace* memadai untuk mengurangi beban kognitif pengguna, sekaligus menyiapkan dasar untuk pengembangan *fitur* tambahan seperti *toolbar formatting* dan tambah gambar.

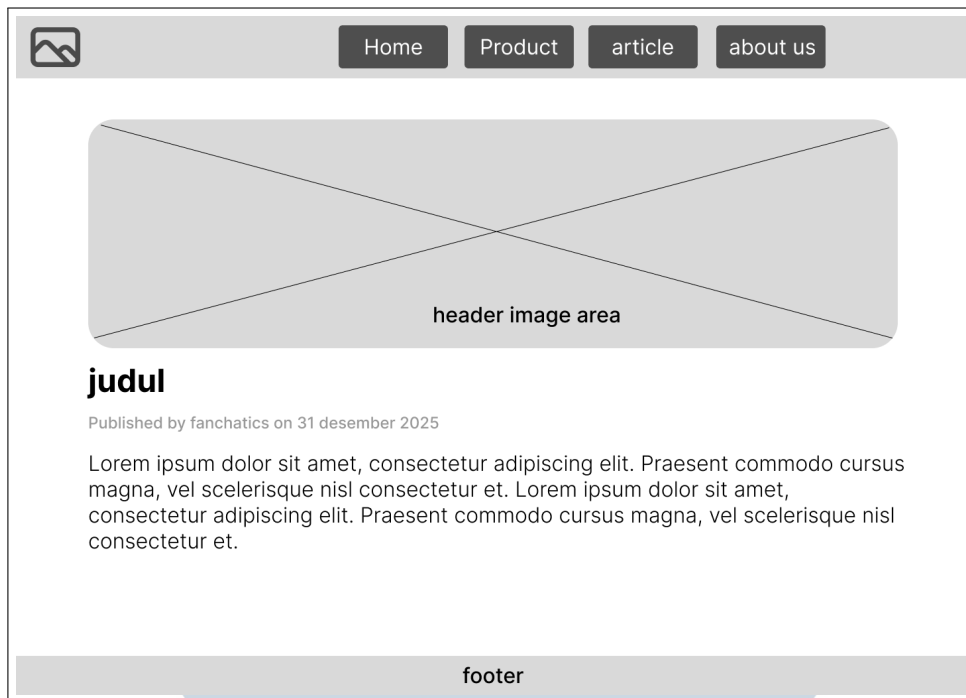
Tata letak *wireframe* menunjukkan kesadaran akan prinsip *UX* dengan alur kerja yang intuitif dari atas ke bawah, memungkinkan pengguna menyelesaikan tugas pembuatan artikel dengan langkah minimal. Penyederhanaan elemen antarmuka ini sengaja dilakukan untuk memfokuskan perhatian pada konten utama, sementara tetap mempertahankan fleksibilitas untuk penambahan komponen seperti tombol aksi dan sistem validasi *form* di tahap pengembangan berikutnya. *Wireframe* ini secara efektif menanggapi kebutuhan fungsional dasar sambil memberikan ruang bagi penyempurnaan *iteratif* tanpa mengorbankan kejelasan tujuan utama halaman. Tampilan *wireframe* tambah artikel terlampir pada gambar 3.12



Gambar 3.12. tampilan desain kasar tambah artikel

Wireframe halaman detail artikel menampilkan pendekatan desain yang berfokus pada keterbacaan dan penyajian konten yang optimal. Halaman ini menampilkan judul artikel sebagai elemen utama dengan ukuran yang menonjol, diikuti oleh informasi publikasi yang mencakup nama penulis dan tanggal posting. Struktur ini memastikan pengguna langsung memahami konteks artikel sekaligus menegaskan kredibilitas konten melalui penampilan informasi penulis yang jelas.

Bagian isi artikel menggunakan *placeholder teks* yang nantinya akan diisi dengan konten aktual, dengan jarak dan *spacing* yang cukup untuk meningkatkan kenyamanan membaca. Garis pemisah horizontal sebelum *footer* berfungsi sebagai penanda visual akhir konten, sementara *footer* itu sendiri memberikan penutup yang konsisten dengan desain keseluruhan *website*. *Wireframe* ini dirancang dengan kesederhanaan sebagai prioritas, meminimalisir elemen pengganggu sehingga fokus pengguna sepenuhnya berada pada konten artikel. Pendekatan ini tidak hanya memenuhi kebutuhan fungsional tetapi juga menciptakan dasar yang kuat untuk pengembangan elemen interaktif atau terkait lainnya di tahap selanjutnya. Tampilan *wireframe* detail artikel terlampir pada gambar 3.13



Gambar 3.13. tampilan desain kasar detail artikel

Pada bagian *wireframe* halaman daftar artikel pengguna dirancang dengan pendekatan minimalis untuk memudahkan pengelolaan konten pribadi secara efisien. Antarmuka ini menampilkan tabel sederhana berisi daftar artikel milik pengguna dengan dua kolom utama yaitu kolom judul artikel yang dapat diklik dan kolom status yang menggunakan simbol visual intuitif. Desain ini memungkinkan pengguna untuk dengan cepat memahami status artikel mereka sekilas, didukung oleh hierarki visual yang jelas dan penggunaan ruang yang optimal untuk meningkatkan keterbacaan.

Wireframe ini sengaja dibuat sederhana namun menyiapkan dasar untuk pengembangan *fitur* lebih lanjut seperti penyaringan berdasarkan status, fungsi pencarian, dan *paginasi*. Konsistensi dengan desain sistem secara keseluruhan terjaga melalui penempatan *footer* yang seragam, sementara penggunaan *ikon* universal memastikan kemudahan pemahaman tanpa perlu penjelasan *tekstual* yang panjang. Pendekatan ini tidak hanya memenuhi kebutuhan fungsional dasar tetapi juga memberikan fleksibilitas untuk penambahan elemen interaktif di tahap pengembangan selanjutnya. Tampilan *wireframe* list artikel terlampir pada gambar 3.14



Gambar 3.14. tampilan desain kasar list artikel

3.3.2 Implementasi Pengembangan Website

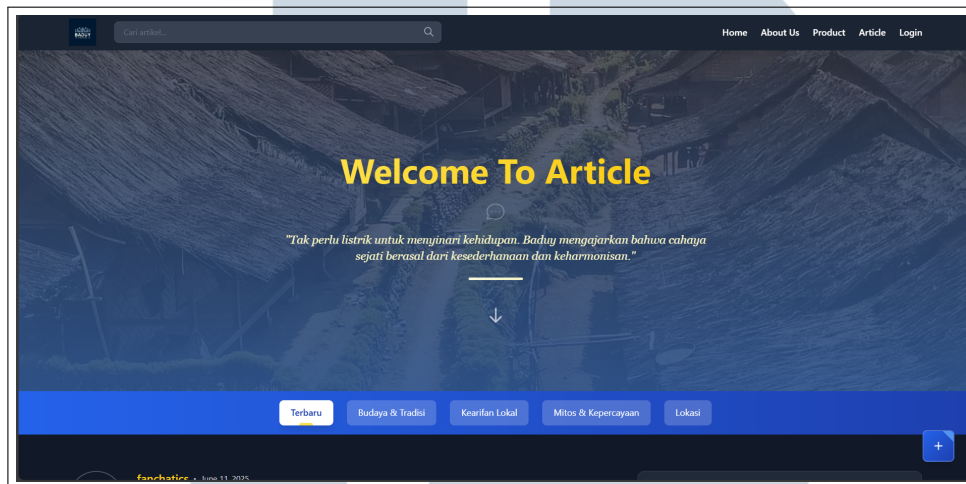
Selama kerja magang di LPPM UMN, ada berbagai jenis halaman dan sistem yang perlu dibuat sebagai *fullstack website developer*. Halaman dan sistem yang dikembangkan yaitu sebagai berikut.

A Halaman Artikel

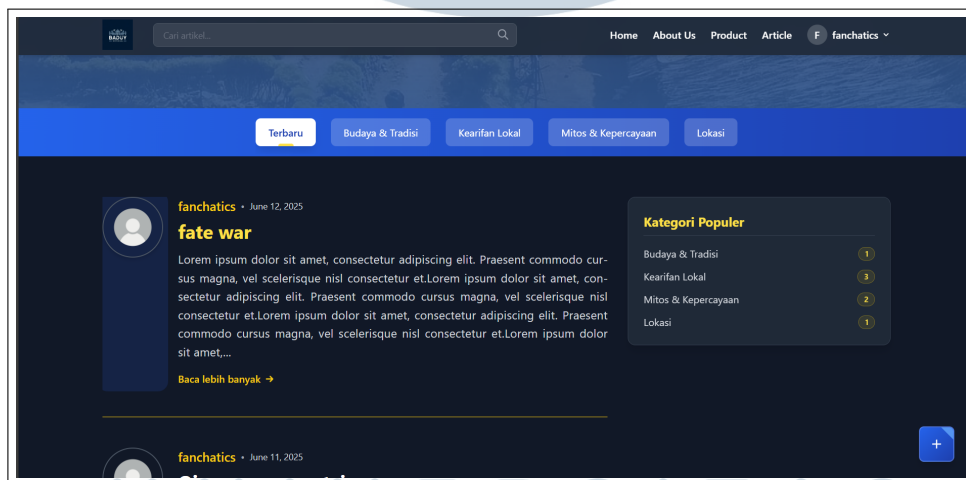
Pengembangan halaman *web* artikel merupakan *platform* yang menyajikan konten-konten terkait budaya, tradisi, kearifan lokal, mitos, kepercayaan, serta lokasi-lokasi tertentu. Tujuannya adalah untuk memberikan informasi yang mendalam dan autentik mengenai berbagai aspek kebudayaan, sehingga pembaca dapat memperoleh pemahaman yang lebih luas tentang nilai-nilai lokal. Dengan struktur yang terorganisir, halaman ini memudahkan pengguna dalam menelusuri topik-topik spesifik sesuai dengan minat mereka.

Konten pada halaman artikel ini mencakup tulisan-tulisan berbasis riset atau narasi yang mengangkat tema-tema kebudayaan, seperti kutipan filosofis tentang kehidupan sederhana atau ulasan mengenai tradisi masyarakat. Selain itu, terdapat *fitur* "Baca lebih banyak" dan kategori populer yang memungkinkan

pembaca untuk menjelajahi artikel-artikel terkait. Hal ini memperkaya pengalaman pengguna sekaligus mendukung penyebaran pengetahuan budaya secara lebih luas dan terstruktur. Tampilan halaman artikel bagian paling atas dan bawah terlampir pada gambar 3.15 dan 3.16



Gambar 3.15. Halaman artikel *banner*



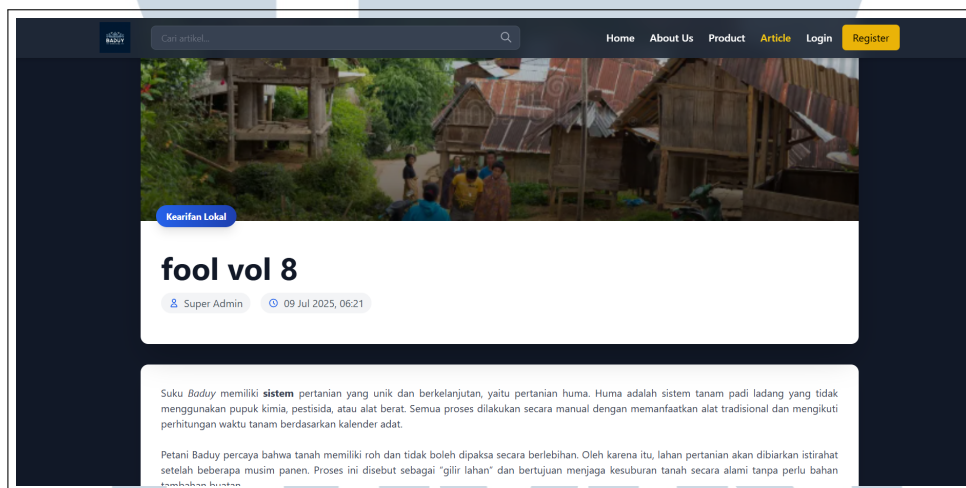
Gambar 3.16. Halaman artikel pada bagian bawah

B Halaman Detail Artikel

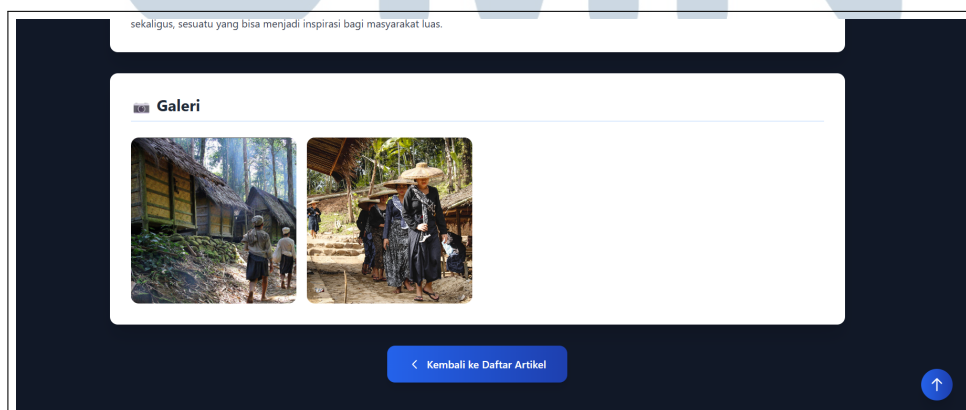
Halaman detail artikel dirancang untuk menampilkan konten secara lengkap dengan tata letak yang menarik dan mudah dibaca. Bagian utama halaman ini menampilkan gambar *header* artikel yang responsif, dengan efek perbesaran halus saat pengguna mengarahkan kursor, memberikan pengalaman visual yang dinamis.

Gambar tersebut dilengkapi dengan *overlay* gradien transparan yang memuat judul artikel dan label "Featured Article" untuk menonjolkan konten unggulan. Desain ini tidak hanya memperkuat estetika tetapi juga memastikan teks tetap terbaca dengan jelas di atas gambar latar.

Di bawah gambar *header*, terdapat informasi publikasi seperti nama penulis dan tanggal rilis, diikuti dengan konten artikel yang diformat secara rapi dengan paragraf yang mudah diikuti. Halaman ini juga dilengkapi dengan tombol "Back to Top" yang memudahkan pengguna untuk kembali ke bagian atas tanpa harus menggulir manual. Secara keseluruhan, halaman detail artikel ini menggabungkan fungsionalitas dan desain yang *user-friendly*, memastikan pembaca dapat fokus pada konten tanpa gangguan. Tampilan halaman detail artikel terlampir pada gambar 3.17 dan gambar 3.18



Gambar 3.17. Tampilan halaman detail artikel



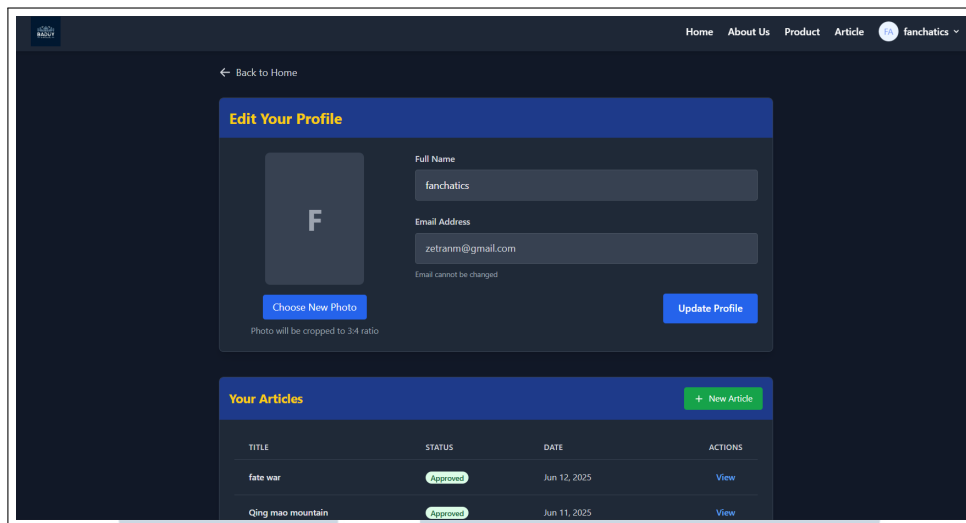
Gambar 3.18. Tampilan dokumentasi gambar tambahan pada bagian bawah

C Halaman Tambah Artikel

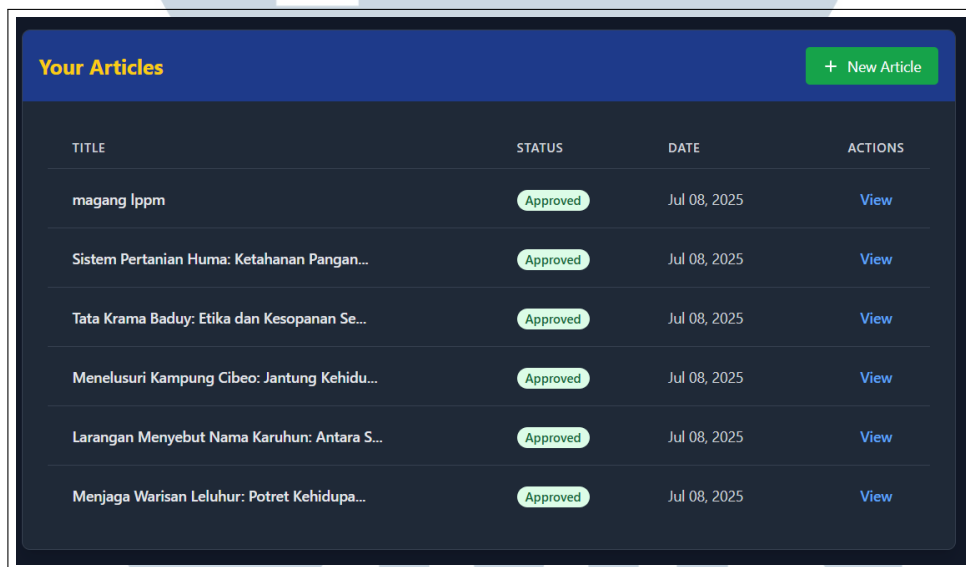
Halaman tambah artikel merupakan antarmuka yang dirancang untuk memudahkan penulis dalam membuat konten baru dengan format yang terstruktur. Halaman ini menampilkan *form input* yang mencakup beberapa elemen penting seperti judul artikel, kategori, gambar header, dan konten utama. Desain antarmuka menggunakan gradien warna dan efek visual modern yang memberikan kesan profesional namun tetap *user-friendly*, dengan penekanan pada pengalaman pengguna yang optimal.

Formulir ini dilengkapi dengan editor teks Trix yang memungkinkan penulis untuk mengubah konten secara fleksibel, termasuk penyisipan gambar dan pengaturan teks. Terdapat juga peringatan penting yang mengingatkan penulis untuk memeriksa kembali konten sebelum dipublikasi, karena artikel tidak dapat diubah setelah proses penyimpanan. Tombol *submit* dengan animasi interaktif memberikan *feedback* visual saat pengguna berinteraksi, menyempurnakan proses pengiriman data dengan pengalaman yang lebih dinamis dan menyenangkan. Tampilan halaman tambah artikel dan notifikasi artikel terlampir pada gambar 3.19 dan 3.20

Gambar 3.19. Tampilan halaman tambah artikel



Gambar 3.21. Tampilan halaman *edit profile*



Gambar 3.22. Tampilan halaman daftar artikel di *profile*

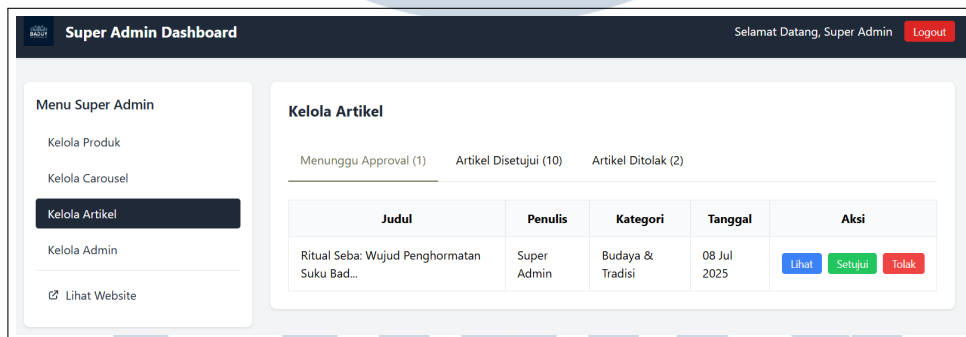
E Halaman Kelola Artikel Admin

Halaman kelola artikel pada panel admin dirancang untuk memudahkan administrator dalam memantau dan mengatur konten artikel yang diajukan oleh pengguna. Antarmuka ini terbagi menjadi dua tab utama: "Menunggu Approval" untuk artikel yang belum direview dan "Artikel Disetujui" untuk konten yang telah lolos verifikasi. Setiap artikel menampilkan informasi lengkap seperti judul, penulis, kategori, dan tanggal pembuatan, disajikan dalam format tabel yang

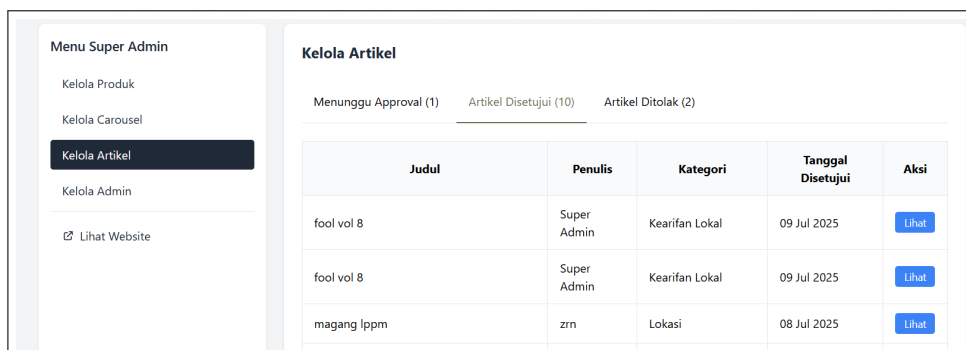
rapi dan mudah dipindai. Administrator dapat dengan cepat meninjau konten, menyetujui, atau menolak artikel dengan memberikan alasan spesifik melalui modal dialog yang tersedia.

Fitur penolakan artikel dilengkapi dengan formulir yang memungkinkan admin memberikan umpan balik konstruktif kepada penulis, sehingga mereka memahami alasan penolakan dan dapat melakukan perbaikan. Untuk artikel yang telah disetujui, admin tetap dapat mengakses pratinjau konten tanpa kemampuan mengedit, menjaga integritas konten yang telah dipublikasikan. Sistem ini dirancang responsif, memastikan pengalaman penggunaan yang optimal baik di desktop maupun perangkat mobile.

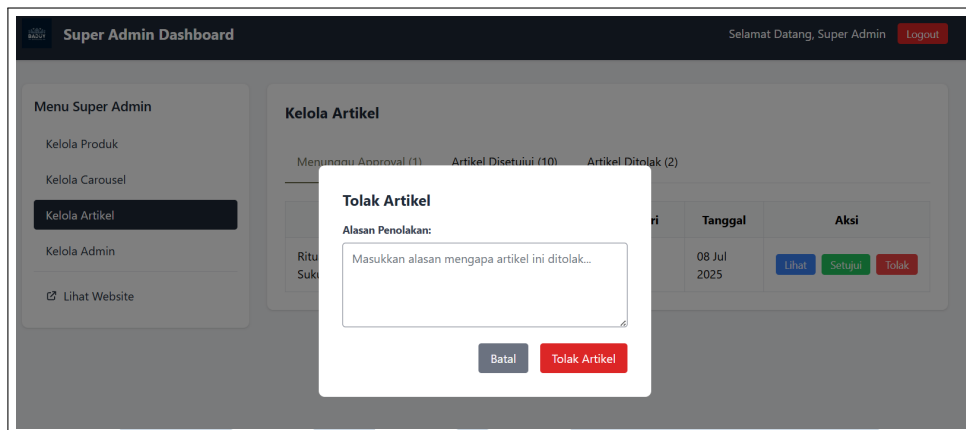
Antarmuka ini juga menampilkan notifikasi visual untuk mengkonfirmasi setiap tindakan yang dilakukan, seperti persetujuan atau penolakan artikel. Desain yang bersih dan navigasi yang intuitif memungkinkan administrator bekerja secara efisien dalam mengelola volume artikel yang masuk. Dengan kombinasi antara fungsionalitas lengkap dan antarmuka yang user-friendly, halaman ini menjadi alat yang efektif untuk menjaga kualitas konten pada platform. Tampilan semua menu halaman admin untuk kelola artikel terlampir pada beberapa gambar dibawah ini



Gambar 3.23. Tampilan halaman admin untuk menu kelola artikel



Gambar 3.24. Tampilan halaman admin untuk artikel yang telah disetujui



Gambar 3.25. Tampilan *pop-up* untuk memberikan alasan penolakan artikel



Gambar 3.26. Tampilan halaman admin untuk artikel yang telah ditolak

F Deployment

Deployment merupakan proses migrasi aplikasi dari lingkungan pengembangan ke lingkungan produksi agar dapat diakses secara luas oleh pengguna akhir. Dalam proyek ini, proses *deployment* dilakukan menggunakan layanan Niagahoster sebagai penyedia *hosting* yang mendukung *framework* Laravel. Niagahoster dipilih karena kemudahan dalam mengelola *server*, *fitur* yang lengkap, serta kompatibilitasnya yang baik dengan kebutuhan teknis aplikasi berbasis Laravel.

Proses *deployment* diawali dengan mempersiapkan *environment production* di Niagahoster, termasuk *database*, *setting* versi PHP yang kompatibel dengan Laravel, serta pengaturan *security*. Selanjutnya dilakukan transfer seluruh *source*

code aplikasi melalui File Manager atau Git, diikuti dengan instalasi *dependensi* melalui *Composer* dan *konfigurasi environment variables*. Setelah semua tahapan selesai, tim akan melakukan serangkaian *testing* untuk memastikan aplikasi berjalan baik di *server production*.

Setelah proses *deployment* selesai dan aplikasi berjalan dengan aman, tim bersama *supervisor* akan menyerahkan seluruh detail aplikasi kepada klien. Penyerahan ini mencakup dokumentasi teknis, *credential* akses, serta panduan *maintenance*. Dengan demikian, klien memiliki kendali penuh atas aplikasi dan dapat melakukan pengelolaan lebih lanjut sesuai kebutuhan.

3.4 Kendala dan Solusi yang Ditemukan

Selama pelaksanaan magang, muncul beberapa kendala dalam pengembangan aplikasi, baik teknis maupun non-teknis:

1. Kesalahan teknis pada pemanggilan gambar akibat kurangnya pemahaman alur kerja *back-end*. Gambar yang diunggah sering gagal muncul di antarmuka pengguna meskipun tersimpan di *database*, disebabkan ketidaksesuaian *path* penyimpanan dengan struktur folder di *server*.
2. Ketidakselarasan warna dengan desain tim, di mana *palette* yang dipilih tim desain tidak konsisten saat diimplementasikan ke kode, terutama karena perbedaan interpretasi nilai *hex code*. Hal ini memicu perdebatan antar anggota.

Berdasarkan kendala-kendala yang ditemukan selama proses program kerja magang, maka solusi yang dapat diterapkan untuk mengatasi kendala-kendala tersebut, sebagai berikut:

1. Perbaiki *debugging error* dengan mempelajari dokumentasi Laravel terkait file *storage*, lalu juga menyesuaikan konfigurasi *path* penyimpanan. Adapula menggunakan fungsi *asset* untuk memastikan *path* gambar menghasilkan dengan benar.
2. Penyelarasan warna melalui diskusi dengan tim desain untuk menyepakati *palette* warna yang konsisten. Pembuatan dokumentasi resmi berisi daftar warna beserta kode *hex*-nya sebagai acuan bersama.