

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama melaksanakan magang di Vision+ Academy, posisi yang ditempati adalah sebagai *Software Developer Intern* pada divisi *frontend*, yang berada langsung di bawah naungan *Frontend Department*. Dalam pelaksanaan magang, setiap proyek umumnya dikerjakan oleh dua orang penanggung jawab, yaitu satu orang dari divisi *frontend* dan satu orang dari divisi *backend*. Penentuan penanggung jawab proyek ditetapkan berdasarkan tingkat prioritas dan skala proyek yang sedang berjalan.

Proses koordinasi proyek dilakukan melalui berbagai media, antara lain *WhatsApp Chat*, *Gitlab*, *Jira Software*, serta laporan *daily progress*. Koordinasi ini bertujuan untuk memantau perkembangan pengerjaan proyek sekaligus menjadi wadah diskusi dalam mengatasi kendala yang muncul selama proses pengembangan.

3.2 Tugas yang Dilakukan

Tugas utama yang dilaksanakan selama program kerja magang adalah mengembangkan sebuah *dashboard FAQ* untuk situs web Vision+. Proyek ini berfokus pada pembuatan sistem manajemen konten yang dinamis, sehingga memudahkan pihak admin dalam menambah, mengubah, maupun menghapus artikel *FAQ* tanpa harus melakukan proses teknis secara manual. Selain itu, *dashboard* juga dilengkapi dengan fitur analitik yang memungkinkan tim internal untuk memantau kinerja artikel serta melakukan evaluasi terhadap kebutuhan pembaruan konten.

Dengan adanya *dashboard* ini, pengelolaan informasi menjadi lebih efisien, terstruktur, dan sesuai dengan kebutuhan pengguna. Admin dapat mengatur artikel berdasarkan kategori/topik, meninjau efektivitas artikel melalui data interaksi, serta menjaga agar konten *FAQ* tetap relevan dan mudah diakses.

Adapun fitur-fitur utama pada *dashboard FAQ* meliputi:

1. *Article Management*: Menyediakan fungsi untuk membuat, mengedit, menghapus, serta mengarsipkan artikel *FAQ* secara praktis.

2. *Multi-language Support*: Menghubungkan artikel dengan berbagai bahasa dan menetapkan *fallback* apabila terjemahan belum tersedia.
3. *Engagement Metrics*: Menampilkan data performa artikel seperti jumlah tayangan, suka, dan tidak suka.
4. *Admin Dashboard Interface*: Menyediakan tampilan yang memudahkan admin dalam memfilter konten berdasarkan kategori atau bahasa, sekaligus memantau artikel yang perlu diperbarui.

3.3 Uraian Pelaksanaan Magang

Dalam pengembangan *project* ini, digunakan berbagai teknologi pendukung untuk memastikan sistem berjalan secara optimal dan mudah dikelola. Pada sisi *frontend*, diterapkan **Nuxt** yang terintegrasi dengan **TypeScript** guna membangun antarmuka pengguna yang dinamis, terstruktur, dan mudah dikembangkan. Selain itu, penggunaan *Pinia* sebagai *state management* membantu pengelolaan data artikel, topik, dan bahasa secara efisien.

Untuk menciptakan tampilan yang konsisten serta responsif di berbagai perangkat, digunakan *Tailwind CSS*, sebuah *framework* CSS modern yang mempercepat proses *styling* dan menjaga konsistensi desain. Sistem ini juga dilengkapi dengan dukungan multi bahasa, yang memungkinkan pengguna untuk menampilkan dan mengelola konten *FAQ* dalam berbagai bahasa melalui pengaturan yang terintegrasi dengan komponen antarmuka.

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 dilaksanakan setiap minggu

Minggu Ke -	Pekerjaan yang dilakukan
1	Pada minggu pertama, kegiatan difokuskan pada pengembangan awal tampilan <i>dashboard</i> . Proses ini meliputi pembuatan struktur dasar antarmuka serta penyusunan komponen utama yang digunakan dalam halaman <i>overview</i> . Pengembangan kemudian dilanjutkan dengan penyempurnaan halaman <i>overview</i> agar lebih informatif dan sesuai dengan rancangan awal. Di akhir minggu, dilakukan evaluasi melalui <i>weekly report</i> untuk meninjau kemajuan yang telah dicapai.
2	Memasuki minggu kedua, fokus pekerjaan beralih pada pengembangan halaman <i>manajemen konten</i> . Kegiatan mencakup pembuatan komponen utama untuk menampilkan dan mengelola data artikel, topik, serta subtopik. Selama minggu ini, proses pengembangan dilakukan secara berkelanjutan dengan penyempurnaan fitur tampilan dan logika <i>frontend</i> . Kegiatan diakhiri dengan <i>weekly report progress</i> sebagai bentuk evaluasi terhadap hasil pekerjaan mingguan.
3	Pada minggu ketiga, kegiatan diarahkan pada pembuatan <i>stores management</i> sebagai pusat pengelolaan data aplikasi menggunakan <i>Pinia</i> . Pengembangan ini bertujuan untuk mempermudah pengaturan state antar komponen dan memastikan sinkronisasi data berjalan efisien. Selain itu, dilakukan pembuatan dan penyempurnaan komponen <i>sidebar menu</i> untuk navigasi halaman <i>dashboard</i> . Seperti minggu sebelumnya, kegiatan ditutup dengan <i>weekly report progress</i> untuk menilai hasil dan kendala selama proses pengembangan.
4	Minggu keempat difokuskan pada pembuatan <i>layout</i> halaman <i>Add Content</i> yang menjadi bagian utama dari fitur manajemen artikel. Dalam tahap ini juga dikembangkan <i>form</i> untuk menambahkan topik, subtopik, serta artikel baru. Pengujian dilakukan untuk memastikan setiap elemen <i>form</i> berfungsi dengan baik, dan kegiatan ditutup dengan pelaporan mingguan melalui <i>weekly report progress</i> .
Lanjut pada halaman berikutnya	

Tabel 3.1. Pekerjaan yang dilaksanakan setiap minggu (Lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
5	Pada minggu kelima, kegiatan diarahkan pada pembuatan halaman <i>Article Detail</i> yang menampilkan isi artikel secara lengkap. Selain itu, dilakukan instalasi <i>extension editor content</i> untuk mendukung pengeditan teks secara interaktif pada halaman artikel. Di akhir minggu, dikembangkan pula halaman <i>Edit Artikel</i> dan dilakukan pembaruan <i>weekly report progress</i> untuk memantau perkembangan pekerjaan.
6	Minggu keenam difokuskan pada proses migrasi proyek ke <i>repository utama</i> . Kegiatan mencakup pengaturan ulang konfigurasi proyek, instalasi dependensi yang diperlukan, serta penyesuaian struktur folder agar sesuai dengan standar pengembangan tim. Setelah semua penyesuaian dilakukan, proyek dijalankan dan diuji untuk memastikan tidak terdapat kendala pasca-migrasi. Evaluasi hasil kerja dilakukan melalui <i>weekly report progress</i> .
7	Pada minggu ketujuh, kegiatan difokuskan pada integrasi fitur <i>FAQ</i> ke dalam <i>sidebar menu</i> , sehingga fitur tersebut dapat diakses langsung dari navigasi utama. Selanjutnya dilakukan pembuatan <i>layout</i> untuk halaman <i>Overview Statistics FAQ</i> yang menampilkan data statistik terkait artikel dan topik. Pengembangan diakhiri dengan proses penyempurnaan tampilan dan pengujian fungsionalitas halaman.
8	Minggu kedelapan berfokus pada pengaturan <i>store management content</i> dan penerapan <i>language store</i> untuk mendukung sistem konten multibahasa. Tahap ini melibatkan sinkronisasi antara <i>content store</i> dan <i>language store</i> agar setiap konten dapat menyesuaikan bahasa yang dipilih pengguna secara otomatis.
Lanjut pada halaman berikutnya	

Tabel 3.1. Pekerjaan yang dilaksanakan setiap minggu (Lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
9	Pada minggu kesembilan, kegiatan diarahkan pada pengembangan halaman <i>Management Content</i> . Pekerjaan mencakup pembuatan <i>UI</i> seperti <i>search bar</i> , tombol <i>change language</i> , serta sinkronisasi komponen navigasi. Selain itu, dilakukan pembuatan tabel manajemen konten serta integrasi API untuk menampilkan daftar topik. Proses dilanjutkan dengan sinkronisasi sistem multi bahasa dan evaluasi hasil kerja melalui <i>weekly report progress</i> .
10	Minggu kesepuluh difokuskan pada integrasi API untuk subtopik dan artikel. Pengembangan meliputi pembuatan daftar subtopik berdasarkan <i>topic ID</i> , serta daftar artikel berdasarkan <i>subtopic ID</i> . Selain itu, dilakukan penyesuaian agar seluruh data dapat ditampilkan dalam berbagai bahasa melalui sinkronisasi API multibahasa. Evaluasi mingguan dilakukan melalui <i>weekly report progress</i> .
11	Pada minggu kesebelas, kegiatan mencakup instalasi <i>extension</i> tambahan untuk editor konten serta pengembangan halaman <i>Get Article Detail</i> . Setelah itu, dilakukan penyempurnaan tampilan halaman detail artikel agar lebih rapi dan informatif. Progres pekerjaan dilaporkan melalui <i>weekly report progress</i> .
12	Minggu kedua belas difokuskan pada pengembangan halaman <i>Edit Article</i> dan integrasi API untuk fitur pengeditan artikel. Selain itu, dilakukan perapihan tampilan antarmuka serta evaluasi hasil pengembangan melalui <i>weekly report update progress</i> .
13	Pada minggu ketiga belas, kegiatan dimulai dengan instalasi <i>Vue Draggable Extension</i> sebagai dukungan untuk fitur <i>drag and drop</i> . Fitur ini kemudian diintegrasikan dengan backend untuk mengatur urutan topik dan subtopik secara interaktif. Pengembangan bertujuan meningkatkan efisiensi pengelolaan konten di halaman admin.
Lanjut pada halaman berikutnya	

Tabel 3.1. Pekerjaan yang dilaksanakan setiap minggu (Lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
14	Minggu keempat belas melanjutkan integrasi fitur <i>drag and drop</i> untuk artikel. Selain itu, dilakukan perbaikan terhadap bug yang muncul pada fitur perubahan bahasa (<i>change language content fetch</i>). Kegiatan diakhiri dengan pelaporan hasil pengembangan melalui <i>weekly report update progress</i> .
15	Pada minggu kelima belas, dilakukan penambahan fitur <i>live chat button</i> pada halaman utama <i>FAQ</i> untuk mempermudah interaksi pengguna. Setelah fitur berhasil ditambahkan, dilakukan perbaikan tata letak tombol agar sesuai dengan desain tampilan. Minggu ini juga mencakup perbaikan bug pada halaman <i>Get Article Detail</i> serta pelaporan hasil pekerjaan melalui <i>weekly report progress</i> .
16	Minggu keenam belas menjadi tahap akhir pengembangan dengan fokus pada penyempurnaan tampilan dan struktur sistem. Perbaikan dilakukan pada tombol navigasi di tampilan <i>mobile</i> , serta restrukturisasi pada bagian <i>store management content</i> agar lebih efisien dan mudah dikelola. Kegiatan ditutup dengan pembaruan laporan mingguan dan evaluasi keseluruhan hasil magang.

3.4 Perancangan Sistem

Sistem yang dikembangkan difokuskan pada perancangan dan implementasi *dashboard admin* untuk pengelolaan konten *FAQ* pada platform Vision+. Dashboard ini berfungsi sebagai pusat kendali bagi admin dan agen untuk menambah, mengedit, menghapus, serta menganalisis konten *FAQ* yang ditampilkan kepada pengguna.

Perancangan sistem dilakukan dengan mengutamakan kemudahan pengelolaan konten, fleksibilitas antarmuka, serta dukungan terhadap sistem multi-bahasa agar konten dapat diakses dalam berbagai bahasa secara konsisten. Fitur-fitur utama pada sistem ini mencakup pengelolaan konten berbasis kategori, dukungan multimedia, pengaturan urutan artikel menggunakan *drag-and-drop*, serta sistem umpan balik pengguna yang terintegrasi dengan panel analitik.

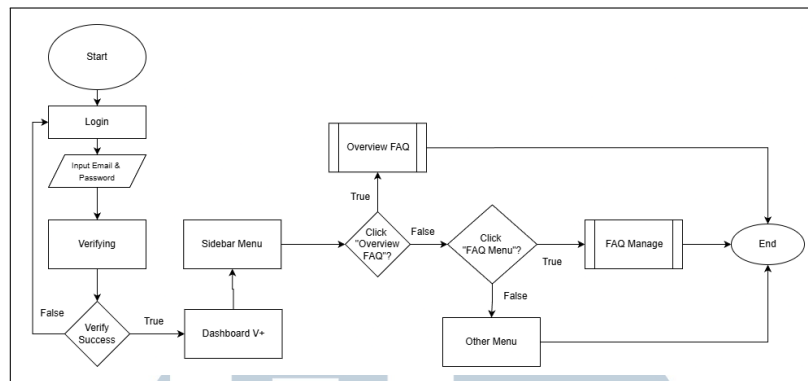
3.4.1 *Admin Requirements*

Berikut merupakan *admin requirements* yang dirancang dalam sistem ini untuk mendukung pengelolaan konten *FAQ* secara terpusat dan efisien:

- a. *Admin* dapat membuat, mengedit, dan menghapus artikel *FAQ*, kategori, serta subtopik melalui antarmuka *dashboard*.
- b. *Admin* dapat menyimpan artikel dalam status *draft*, *publish*, atau *archive*.
- c. *Admin* dapat menambahkan dan mengelola konten *FAQ* dalam berbagai bahasa (*multi-language support*).
- d. Sistem mendukung penyimpanan versi revisi artikel sehingga admin dapat melakukan *rollback* ke versi sebelumnya.
- e. *Admin* dapat menyisipkan konten multimedia berupa teks, gambar, dan video ke dalam artikel *FAQ*.
- f. Sistem menyediakan fitur *preview* untuk melihat tampilan artikel sebelum dipublikasikan.
- g. *Admin* dapat mengatur urutan artikel, kategori, dan subtopik menggunakan fitur *drag-and-drop*.
- h. *Admin* dapat melihat dan menganalisis data statistik artikel *FAQ* melalui fitur *dashboard analytics*.
- i. *Admin* dapat mengekspor data analitik ke dalam format CSV atau Excel.
- j. Sistem menampilkan artikel dengan performa terbaik berdasarkan tingkat keterlibatan pengguna.

3.4.2 *Flowchart Utama Halaman Dashboard*

Alur utama sistem dashboard admin Vision+ ditunjukkan pada Gambar 3.1. Proses dimulai ketika *admin* mengakses *dashboard* melalui website Vision+. Sistem kemudian menampilkan halaman *login* untuk melakukan proses autentikasi dengan memasukkan *email* dan *password*. Data yang dimasukkan akan diverifikasi oleh sistem untuk menentukan hak akses pengguna.



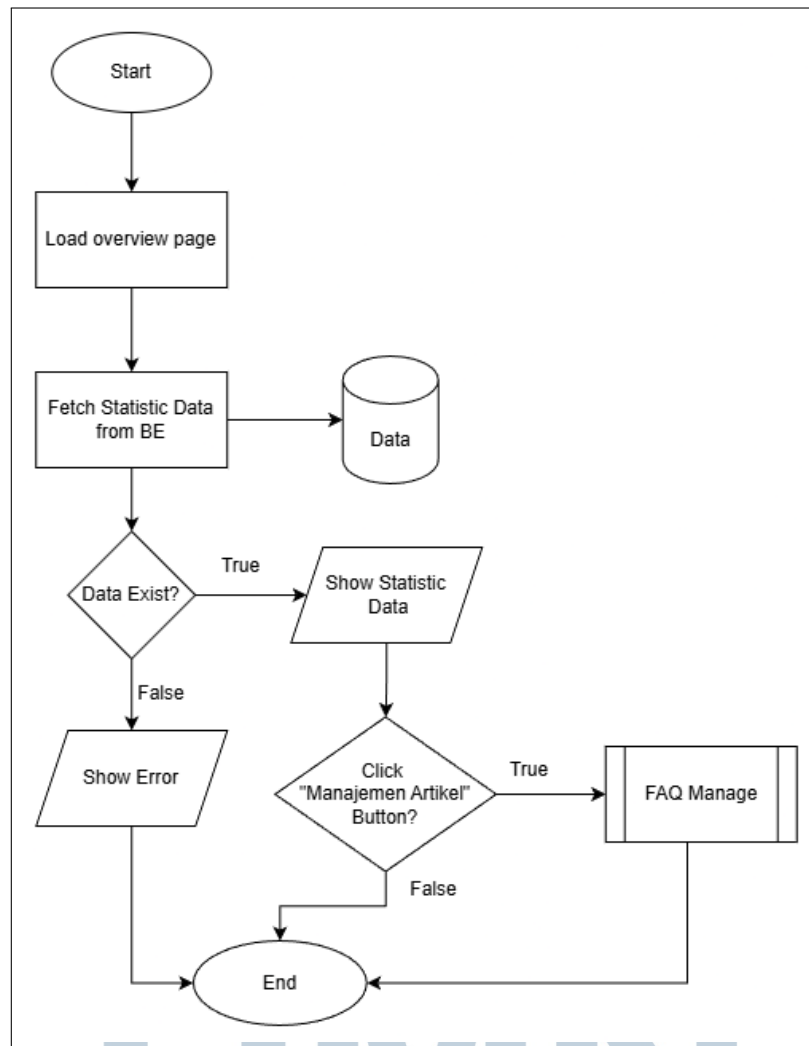
Gambar 3.1. *Flowchart* Utama Halaman *Dashboard*

Berdasarkan alur pada Gambar 3.1, apabila proses autentikasi gagal, sistem akan mengembalikan *admin* ke halaman *login* untuk melakukan pengisian ulang data. Sebaliknya, jika autentikasi berhasil, sistem akan menavigasikan *admin* ke halaman utama *Dashboard Vision+*. Pada halaman ini, *admin* dapat mengakses menu *Overview FAQ* untuk melihat ringkasan data atau menu *FAQ* untuk melakukan pengelolaan konten.

3.4.3 *Flowchart* Halaman *Overview FAQ*

Gambar 3.2 memperlihatkan alur proses yang terjadi pada halaman *Overview FAQ*. Proses dimulai ketika *admin* memilih menu *Overview FAQ* dari dashboard utama, kemudian sistem melakukan permintaan data statistik ke *backend*.

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

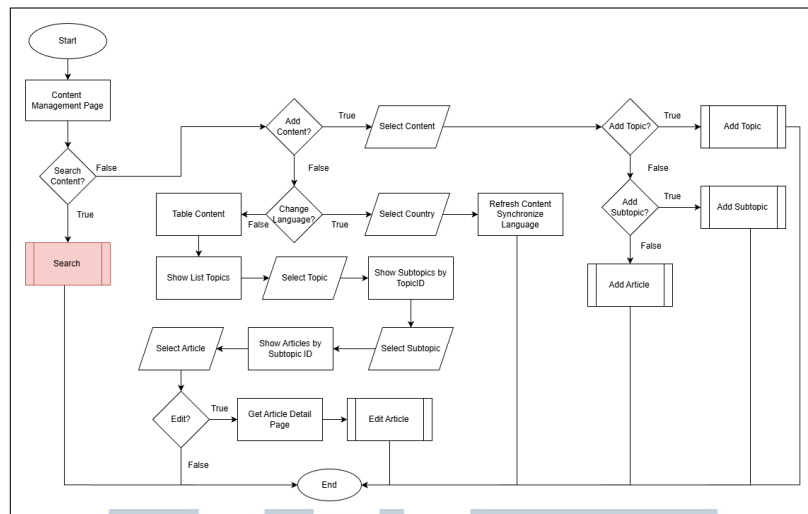


Gambar 3.2. *Flowchart* Halaman Overview FAQ

Berdasarkan alur tersebut, sistem menampilkan data statistik berupa jumlah topik, subtopik, artikel, serta performa artikel dalam bentuk visual seperti *cards* dan grafik. Jika *admin* menekan tombol “Manajemen Artikel”, sistem akan mengarahkan ke halaman Manajemen FAQ, sedangkan jika tidak, *admin* tetap berada pada halaman *Overview FAQ*.

3.4.4 *Flowchart* Halaman Manajemen FAQ

Alur pengelolaan konten FAQ pada halaman Manajemen FAQ ditunjukkan pada Gambar 3.3. Flowchart ini menggambarkan proses pengelolaan topik, subtopik, dan artikel yang dilakukan oleh *admin* dalam satu halaman yang terintegrasi.



Gambar 3.3. Flowchart Halaman Manajemen FAQ

Berdasarkan alur pada Gambar 3.3, proses dimulai ketika *admin* mengakses halaman Manajemen FAQ. Pada tahap awal tersedia fitur *search* untuk pencarian konten, namun pada tahap pengembangan saat ini fitur tersebut masih berupa antarmuka dan belum terintegrasi secara fungsional. Halaman ini juga menyediakan fitur *Add Content* yang memungkinkan *admin* menambahkan konten baru berupa topik, subtopik, atau artikel sesuai kebutuhan.

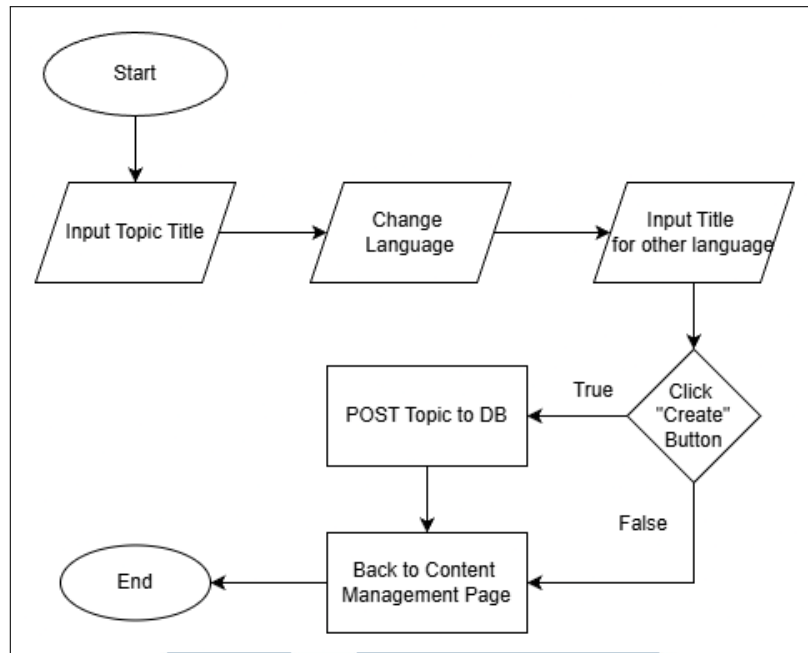
Halaman Manajemen FAQ menampilkan tabel pengelolaan yang terbagi menjadi dua kolom utama. Kolom sebelah kiri berisi daftar topik dalam struktur hierarki yang dilengkapi dengan *dropdown* untuk menampilkan subtopik. Ketika *admin* memilih salah satu subtopik, kolom sebelah kanan akan menampilkan daftar artikel yang terkait beserta informasi statistik seperti jumlah *views*, *likes*, dan *dislikes*. Selain itu, setiap elemen konten dilengkapi fitur *drag and drop* untuk mengatur urutan tampilan konten.

Apabila *admin* memilih salah satu artikel, sistem akan menampilkan opsi *Edit* yang mengarahkan ke halaman *Detail Article*. Dari halaman tersebut, *admin* dapat melanjutkan ke halaman *Edit Article* untuk melakukan pembaruan konten. Dengan alur ini, seluruh proses pengelolaan FAQ dapat dilakukan secara terpusat, efisien, dan terstruktur.

3.4.5 Flowchart Tambah Topik

Gambar 3.4 menunjukkan alur proses penambahan topik baru pada sistem FAQ. Proses ini dilakukan oleh *admin* melalui fitur *Add Content* pada halaman

Manajemen FAQ.

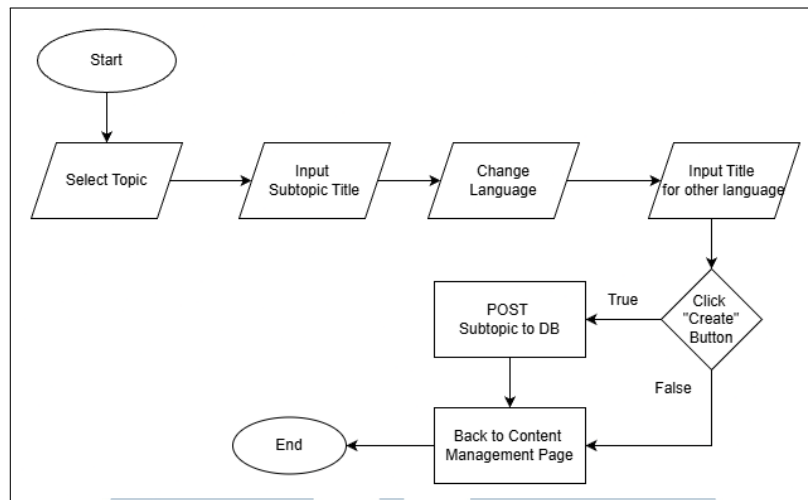


Gambar 3.4. *Flowchart* Tambah Topik

Berdasarkan alur pada Gambar 3.4, proses dimulai dengan pengisian judul topik sesuai bahasa utama yang sedang aktif. Selanjutnya, *admin* dapat menggunakan fitur *Change Language* untuk menambahkan judul topik dalam bahasa lain tanpa perlu membuat data baru. Sistem secara otomatis mengaitkan seluruh versi bahasa ke dalam satu entitas topik yang sama. Setelah seluruh judul topik diisi, *admin* menekan tombol “*Create*” untuk menyimpan data.

3.4.6 *Flowchart* Tambah Subtopik

Alur proses penambahan subtopik pada sistem FAQ ditunjukkan pada Gambar 3.5. Berbeda dengan penambahan topik, proses ini diawali dengan pemilihan topik induk terlebih dahulu.



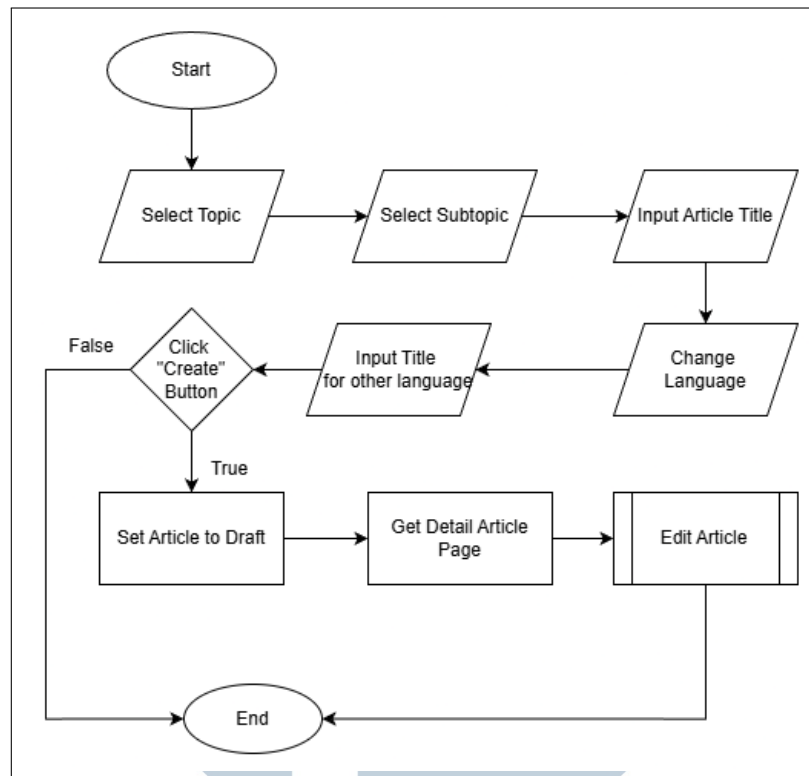
Gambar 3.5. *Flowchart* Tambah Subtopik

Berdasarkan flowchart tersebut, setelah topik induk dipilih, *admin* menginput judul subtopik pada bahasa utama. Selanjutnya, melalui fitur *Change Language*, *admin* dapat menambahkan judul subtopik dalam bahasa lain. Sistem secara otomatis menyesuaikan bahasa pada topik induk, sehingga konsistensi struktur konten tetap terjaga. Setelah seluruh versi bahasa selesai diinput, subtopik disimpan dengan menekan tombol “*Create*”.

3.4.7 *Flowchart* Tambah Artikel

Gambar 3.6 memperlihatkan alur proses penambahan artikel FAQ oleh *admin*. Proses ini merupakan tahap lanjutan setelah topik dan subtopik tersedia dalam sistem.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

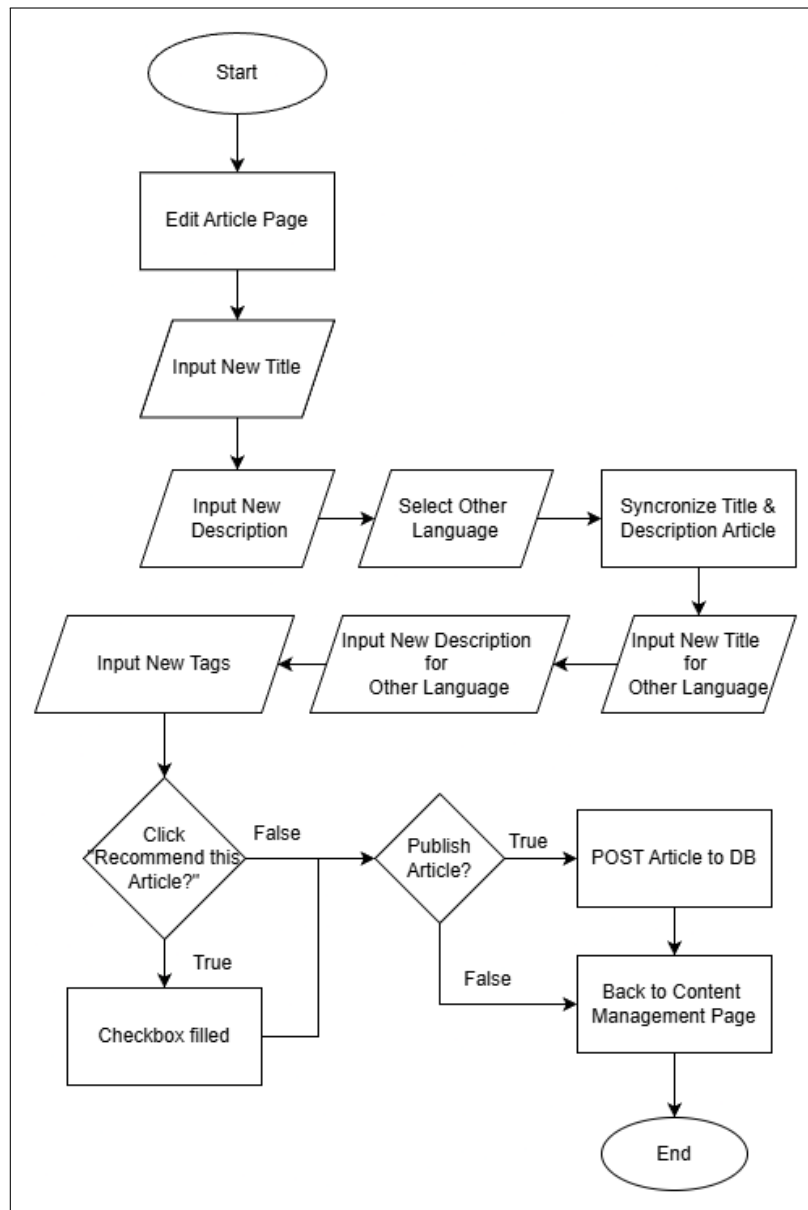


Gambar 3.6. Flowchart Tambah Artikel

Berdasarkan alur pada Gambar 3.6, *admin* terlebih dahulu memilih topik dan subtopik, kemudian menginput judul artikel pada bahasa utama. Selanjutnya, melalui fitur *Change Language*, *admin* dapat menambahkan judul artikel dalam bahasa lain tanpa perlu memilih ulang struktur konten. Setelah artikel dibuat, sistem menetapkan status artikel sebagai *draft* dan menampilkan halaman *Detail Article*. Dari halaman ini, *admin* dapat melanjutkan ke halaman *Edit Article* untuk melengkapi isi artikel, mengatur *tags*, rekomendasi, serta pengaturan publikasi sebelum artikel dipublikasikan.

3.4.8 Flowchart Edit Artikel

Gambar 3.7 menunjukkan alur proses pengeditan artikel pada sistem manajemen *FAQ*. Proses ini dilakukan ketika *admin* mengakses halaman *Edit Article* untuk memperbarui konten artikel yang telah dibuat sebelumnya.



Gambar 3.7. Flowchart Edit Artikel

Berdasarkan alur pada Gambar 3.7, *admin* dapat melakukan perubahan terhadap judul dan deskripsi artikel sesuai kebutuhan. Selanjutnya, melalui fitur pemilihan bahasa, *admin* dapat mengedit versi artikel dalam bahasa lain. Sistem akan melakukan sinkronisasi struktur artikel secara otomatis dan menyediakan kolom pengisian judul serta deskripsi untuk bahasa yang dipilih, sehingga konsistensi konten antarbahasa tetap terjaga.

Selain itu, *admin* dapat menambahkan atau memperbarui *tags* sebagai kata kunci pencarian, serta menandai artikel sebagai artikel rekomendasi melalui

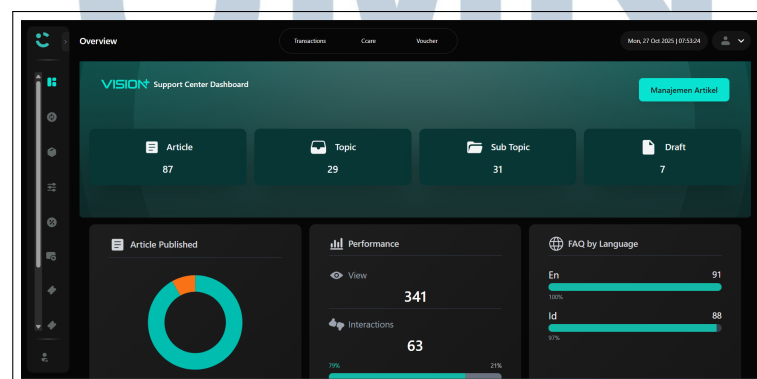
opsi *Recommend this Article*. Pada tahap akhir, *admin* diberikan pilihan untuk mempublikasikan artikel. Jika artikel dipublikasikan, sistem akan menyimpan seluruh perubahan ke dalam *database* dan mengarahkan *admin* kembali ke halaman Manajemen Konten. Sebaliknya, apabila artikel tidak dipublikasikan, *admin* tetap diarahkan ke halaman Manajemen Konten tanpa menyimpan perubahan. Dengan alur ini, proses pengeditan artikel dapat dilakukan secara terstruktur, fleksibel, dan mendukung pengelolaan konten multibahasa secara efisien.

3.5 Implementasi

3.5.1 Halaman *Overview FAQ*

Halaman *Overview FAQ* berfungsi sebagai pusat informasi bagi admin untuk memantau kondisi umum konten *FAQ*. Pada halaman ini, admin dapat melihat ringkasan data konten secara cepat melalui komponen visual yang disediakan. Implementasi tampilan halaman *Overview FAQ* ditunjukkan pada Gambar 3.8.

Pada bagian atas halaman, terdapat empat kartu informasi yang menampilkan jumlah total artikel, topik, subtopik, serta artikel dengan status *draft*. Kartu-kartu ini memberikan gambaran awal mengenai struktur dan status konten yang dikelola dalam sistem. Sementara itu, pada bagian bawah halaman disajikan komponen analitik berupa grafik “*Article Published*”, “*Performance*”, dan “*FAQ by Language*” yang digunakan untuk memantau status publikasi, performa interaksi pengguna, serta distribusi artikel berdasarkan bahasa.



Gambar 3.8. Halaman *Overview FAQ*

Berdasarkan tampilan pada Gambar 3.8, halaman ini juga dilengkapi dengan tombol “*Manajemen Artikel*” yang memungkinkan admin untuk langsung mengakses halaman pengelolaan konten *FAQ*. Keberadaan tombol ini

mempermudah alur kerja admin karena proses pengelolaan konten dapat dilakukan tanpa harus berpindah menu secara manual.

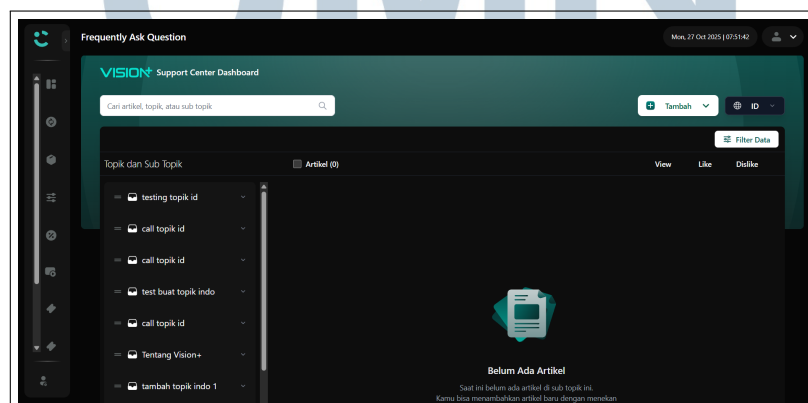
3.5.2 Halaman Manajemen Konten *FAQ*

Halaman Manajemen Konten *FAQ* digunakan sebagai pusat pengelolaan seluruh data topik, subtopik, dan artikel oleh admin. Implementasi tampilan halaman ini ditunjukkan pada Gambar 3.9.

Pada bagian atas halaman, tersedia tombol *Add Content* untuk menambahkan topik, subtopik, atau artikel baru, serta tombol *Change Language* yang digunakan untuk mengganti bahasa tampilan sesuai kebutuhan pengelolaan konten multibahasa. Tampilan halaman terbagi menjadi dua bagian utama, yaitu panel kiri yang menampilkan struktur hierarki topik dan subtopik, serta panel kanan yang menampilkan daftar artikel berdasarkan subtopik yang dipilih, lengkap dengan informasi jumlah *view*, *like*, dan *dislike*. Selain itu, tersedia fitur *Search* dan *Filter Data* yang pada tahap pengembangan ini masih berupa antarmuka dan belum terintegrasi dengan *backend*.

A Tampilan Halaman Tanpa Artikel

Apabila pada suatu subtopik belum terdapat artikel, sistem akan menampilkan notifikasi “Belum Ada Artikel” yang disertai panduan untuk menambahkan artikel baru. Kondisi ini ditunjukkan pada Gambar 3.9.



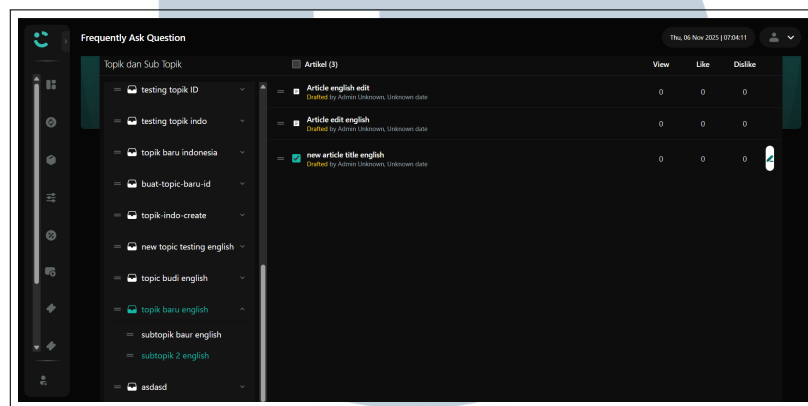
Gambar 3.9. Halaman Manajemen Konten *FAQ* (tanpa artikel)

Berdasarkan tampilan tersebut, sistem secara jelas memberikan umpan balik kepada admin mengenai kondisi data sekaligus mengarahkan tindakan yang dapat

dilakukan selanjutnya.

B Tampilan Halaman dengan Artikel

Ketika subtopik telah memiliki artikel, sistem akan menampilkan daftar artikel secara terstruktur seperti yang ditunjukkan pada Gambar 3.10. Setiap artikel ditampilkan beserta status (*draft* atau *published*), informasi penulis, serta statistik interaksi pengguna. Admin juga dapat melakukan pengeditan artikel secara langsung melalui ikon *edit*.



Gambar 3.10. Halaman Manajemen Konten *FAQ* (dengan artikel)

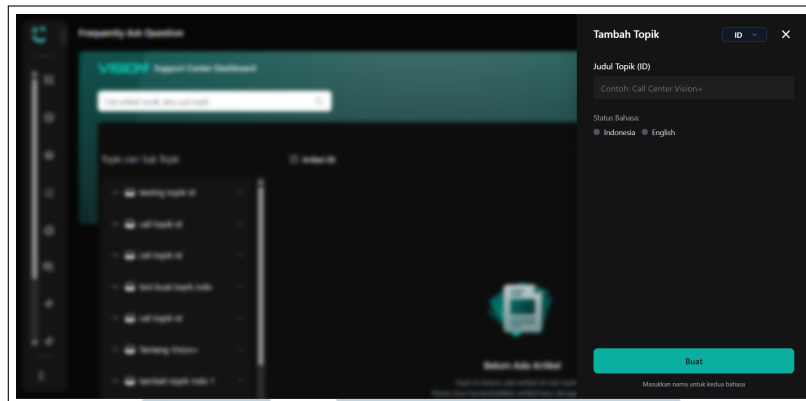
Selain itu, berdasarkan Gambar 3.10, sistem dilengkapi fitur *drag and drop* yang memungkinkan admin mengatur ulang urutan topik, subtopik, dan artikel dengan mudah. Fitur ini memberikan fleksibilitas dalam penyesuaian prioritas tampilan konten tanpa memerlukan pengaturan tambahan.

3.5.3 Fitur Tambah Konten

Fitur ini digunakan untuk menambahkan topik, subtopik, dan artikel baru ke dalam sistem *FAQ*. Fitur ini dirancang untuk mendukung pengelolaan konten multibahasa dengan menjaga konsistensi struktur data antarversi bahasa.

A Tambah Topik

Proses penambahan topik dilakukan melalui formulir yang ditunjukkan pada Gambar 3.11. Admin mengisi judul topik pada bahasa utama yang sedang dipilih, kemudian menyimpan data tersebut.

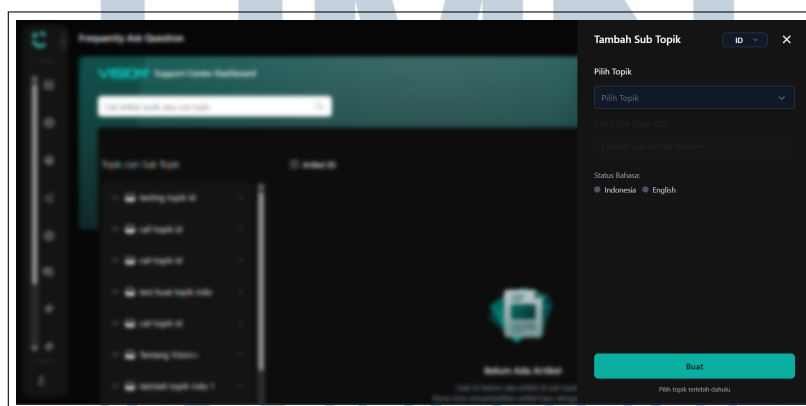


Gambar 3.11. Formulir Tambah Topik Multibahasa

Setelah topik tersimpan, admin dapat menambahkan judul topik dalam bahasa lain melalui fitur pemilihan bahasa. Berdasarkan mekanisme ini, sistem secara otomatis menghubungkan seluruh versi bahasa dalam satu topik yang sama sehingga konsistensi struktur *FAQ* tetap terjaga.

B Tambah Subtopik

Formulir penambahan subtopik ditunjukkan pada Gambar 3.12. Pada tahap ini, admin memilih topik induk terlebih dahulu, kemudian mengisi judul subtopik sesuai bahasa aktif.

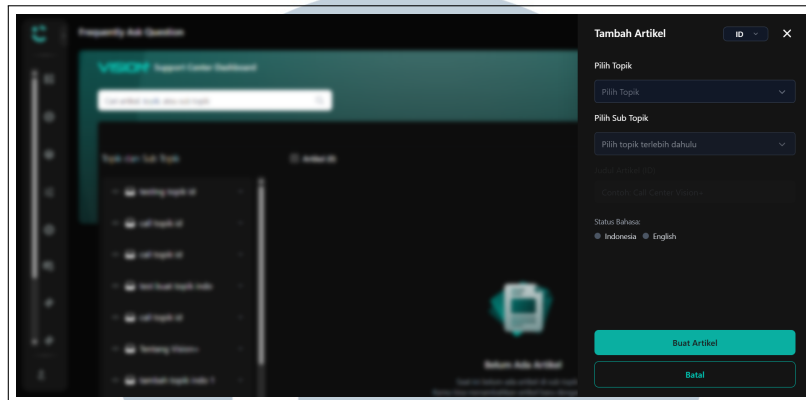


Gambar 3.12. Formulir Tambah Subtopik Multibahasa

Berdasarkan tampilan tersebut, admin dapat menambahkan versi bahasa lain tanpa perlu membuat ulang relasi topik–subtopik, karena sistem menjaga keterkaitan antarbahasa secara otomatis.

C Tambah Artikel

Proses penambahan artikel dilakukan melalui formulir “Tambah Artikel” seperti ditunjukkan pada Gambar 3.13. Admin memilih topik dan subtopik, lalu mengisi judul artikel sesuai bahasa yang dipilih.

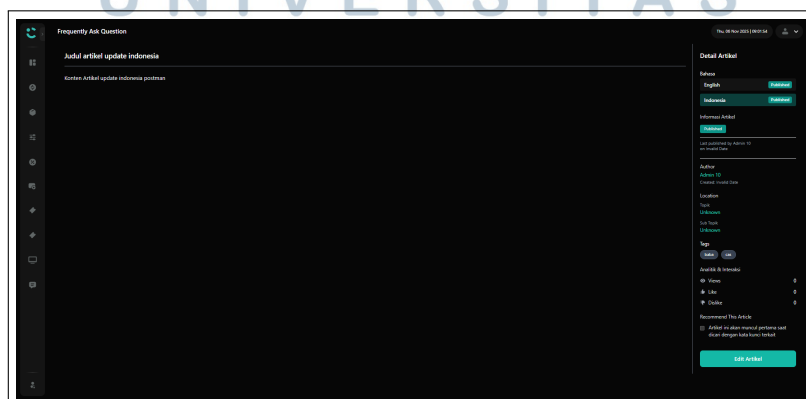


Gambar 3.13. Formulir Tambah Artikel Multibahasa

Setelah data awal diisi, admin dapat menambahkan judul artikel pada bahasa lain tanpa memilih ulang struktur konten. Sistem kemudian mengarahkan admin ke halaman *Edit Artikel* untuk melengkapi isi, *tags*, serta pengaturan publikasi. Alur ini memastikan struktur artikel telah benar sebelum konten dilengkapi secara penuh.

3.5.4 Halaman Detail Artikel

Halaman *Detail Artikel* digunakan untuk menampilkan informasi lengkap terkait suatu artikel. Implementasi tampilan halaman ini ditunjukkan pada Gambar 3.14.

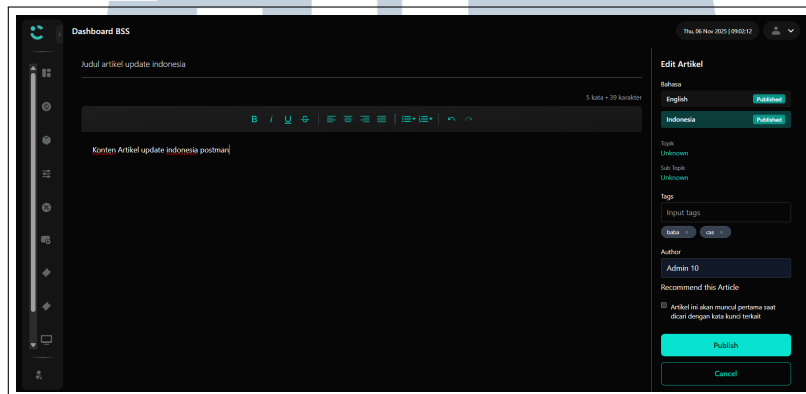


Gambar 3.14. Halaman Detail Artikel

Berdasarkan Gambar 3.14, halaman ini menampilkan informasi seperti tanggal publikasi, nama penulis, posisi artikel dalam struktur *FAQ*, serta metrik interaksi berupa *views* dan *likes*. Selain itu, ditampilkan pula status rekomendasi dan daftar *tags*. Halaman ini juga menyediakan tombol “*Edit Article*” sebagai akses lanjutan untuk melakukan penyuntingan artikel.

3.5.5 Halaman Edit Artikel

Halaman *Edit Artikel* digunakan oleh admin untuk memperbarui konten artikel. Tampilan halaman ini ditunjukkan pada Gambar 3.15.



Gambar 3.15. Halaman Edit Artikel

Berdasarkan tampilan tersebut, admin dapat mengubah judul, isi artikel, *tags*, serta status publikasi melalui editor teks yang dilengkapi fitur pemformatan. Panel informasi di sisi kanan menampilkan pengaturan bahasa, topik, subtopik, penulis, serta opsi *Recommend this Article*. Setelah perubahan dilakukan, admin dapat memilih untuk menyimpan dan menerbitkan artikel atau membatalkan proses penyuntingan.

3.6 Spesifikasi Sistem

Berikut adalah spesifikasi dari *software* yang digunakan dalam pembuatan *project* pada *website* Vision+.

1. *Gitlab*
2. *Brave*
3. *Google Chrome*

4. *Visual Studio Code*

5. *Figma*

6. *Diagrams.net*

7. *Jira Software*

Berikut merupakan spesifikasi *hardware* yang digunakan.

1. *Operating Systems: Windows*
2. *Random Access Memory (RAM): 16 GB*
3. *Storage: 500 GB (SSD)*
4. *Processor: 12th Gen Intel(R) Core(TM) i5-12500H*

3.7 Kendala dan Solusi yang Ditemukan

3.7.1 Kendala

Berikut ini merupakan kendala yang dialami selama pengerjaan magang ini.

1. Adanya kendala saat melakukan migrasi dari *project* lokal ke *repository* utama, seperti konflik versi dan perbedaan konfigurasi.
2. Integrasi dengan *backend* sempat terhambat karena adanya perbedaan struktur data pada API yang diterima dengan kebutuhan di sisi *frontend*.
3. Implementasi fitur *multi-language* cukup kompleks, terutama dalam memastikan konten *FAQ* dapat ditampilkan dengan konsisten sesuai bahasa yang dipilih.
4. Terjadi *bug* pada tampilan *dashboard* ketika menampilkan daftar artikel yang memiliki banyak kategori atau subtopik.

3.7.2 Solusi

Berikut ini merupakan solusi yang ditemukan untuk kendala yang dialami.

1. Melakukan sinkronisasi rutin melalui *Git* dan menyepakati standar konfigurasi agar proses migrasi lebih lancar.

2. Berkoordinasi dengan tim *backend* untuk menyesuaikan struktur data API serta menambahkan *middleware* di sisi *frontend* agar data lebih mudah digunakan.
3. Melakukan pengujian lebih mendalam (*debugging* dan *unit testing*) untuk memperbaiki bug yang muncul pada daftar artikel.
4. Menentukan skala prioritas pekerjaan, serta menjaga komunikasi dan koordinasi antartim agar pengerjaan proyek tetap berjalan sesuai target.

