

Implementation of a Drag-and-Drop Interface for Managing Dynamic Web Page Layouts

Farrel Satriano^{1*}, Nur Fitrianti Fahrudin¹, Deffin Achmadiffa²

¹Department of Information Systems, Institut Teknologi Nasional, Bandung, Indonesia

²Program Studi Ilmu Komputer, Universitas Pendidikan Indonesia, Bandung, Indonesia

*Email Korespondensi: sherifraynor2004@gmail.com

Naskah Masuk 30 November 2025 | Direvisi 30 Januari 2026 | Diterima 15 Februari 2026

ABSTRAK

Pengelolaan tata letak halaman web yang dinamis menuntut adanya antarmuka yang intuitif serta struktur data yang fleksibel untuk menangani hierarki konten yang kompleks dan bersarang. Kegiatan ini bertujuan untuk mengimplementasikan antarmuka Drag-and-Drop (DnD) berbasis pustaka DND Kit yang terintegrasi dengan struktur data JSON untuk merepresentasikan dan memanipulasi tata letak halaman secara real-time pada studi kasus Studi Kasus: Platform Web Content Management System (WCMS) Berbasis Headless. Sistem ini dirancang untuk memungkinkan pengguna menyusun elemen visual secara mandiri tanpa perlu berinteraksi langsung dengan kode program. Pengembangan sistem dilakukan menggunakan metodologi Agile Iterative guna memastikan adaptabilitas terhadap kebutuhan antarmuka pengguna yang terus berkembang. Fungsionalitas sistem divalidasi melalui pengujian Black Box Testing yang mencakup manajemen halaman, manipulasi tata letak, dan publikasi konten. Hasil pengujian terhadap 19 skenario uji utama menunjukkan tingkat keberhasilan sebesar 94,7%. Hasil ini membuktikan bahwa integrasi mekanisme DnD dengan struktur data JSON mampu memfasilitasi pengelolaan tata letak bersarang (nested layout) dan pembaruan properti komponen secara akurat dan responsif sesuai dengan rancangan fungsional, selain itu, kegiatan ini memberikan kontribusi pada perancangan model editor visual berbasis JSON yang bersifat schema-agnostic sehingga mudah dikembangkan dan diintegrasikan pada arsitektur Headless CMS, pendekatan ini juga mendukung agar perubahan tata letak dapat dikelola secara konsisten dan efisien.

Kata kunci: Drag-and-Drop, DND Kit, Struktur Data JSON, Tata Letak Dinamis, Headless CMS

ABSTRACT

Dynamic Dynamic web page layout management requires an intuitive interface and flexible data structure to handle complex and nested content hierarchies. This activity aims to implement a Drag-and-Drop (DnD) interface based on the DND Kit library integrated with JSON data structures to represent and manipulate page layouts in real-time in the case study: Headless-Based Web Content Management System (WCMS) Platform. This system is designed to allow users to arrange visual elements independently without the need to interact directly with the program code. The system was developed using the Agile Iterative methodology to ensure adaptability to evolving user interface requirements. System functionality was validated through Black Box Testing, which included page management, layout manipulation, and content publishing. Testing results for 19 key test scenarios showed a success rate of 94.7%. These results prove that the integration of the DnD mechanism with the JSON data structure is capable of facilitating the management of nested layouts and component property updates accurately and responsively in accordance with the functional design. In addition, this activity contributes to the design of a schema-agnostic JSON-based visual editor model that is easy to develop and integrate into a Headless CMS architecture. This approach also supports consistent and efficient management of layout changes.

Keywords: Drag-and-Drop, DND Kit, JSON Data Structure, Dynamic Layout, Headless CMS

1. PENDAHULUAN

Perkembangan Web Content Management System (WCMS) menunjukkan pergeseran signifikan menuju arsitektur yang lebih fleksibel dan terpisah antara pengelolaan konten dan lapisan presentasi, seiring meningkatnya kebutuhan aplikasi web modern yang harus berjalan lintas perangkat dan platform secara konsisten [1], [2]. Pendekatan Headless CMS menjadi solusi yang banyak diadopsi karena memungkinkan konten dikelola secara terpusat di sisi backend dan disajikan melalui berbagai antarmuka frontend menggunakan Application Programming Interface (API), sehingga meningkatkan skalabilitas dan fleksibilitas pengembangan sistem [2], [3], [4].

Dalam konteks pengembangan aplikasi berbasis web, antarmuka pengguna (User Interface/UI) memegang peranan penting dalam menentukan efektivitas interaksi antara pengguna dan sistem, khususnya pada sistem yang bersifat konfiguratif seperti editor halaman dan dashboard manajemen konten [5], [6]. Antarmuka yang interaktif tidak hanya berfungsi sebagai media visualisasi, tetapi juga sebagai sarana utama bagi pengguna untuk melakukan manipulasi struktur dan konten secara langsung tanpa memerlukan pemahaman teknis yang mendalam terhadap kode program [5], [7].

Salah satu pendekatan yang banyak digunakan untuk meningkatkan interaktivitas antarmuka adalah penerapan mekanisme drag-and-drop (DnD), yang memungkinkan pengguna memindahkan, menyusun ulang, dan mengelola elemen visual secara intuitif melalui interaksi langsung [8], [9]. Studi-studi terkini menunjukkan bahwa mekanisme drag-and-drop sangat efektif dalam mendukung proses penyusunan tata letak halaman dan workflow visual, terutama pada aplikasi editor berbasis web yang menuntut responsivitas dan kemudahan penggunaan [9], [10].

Pengelolaan tata letak halaman secara dinamis melalui interaksi drag-and-drop memerlukan struktur data yang fleksibel dan mampu merepresentasikan hierarki elemen yang bersarang (nested). Dalam hal ini, JavaScript Object Notation (JSON) telah menjadi format data yang dominan karena sifatnya yang ringan, mudah dibaca, serta didukung secara luas oleh berbagai teknologi frontend dan backend modern [11], [12]. Penggunaan JSON memungkinkan perubahan struktur tata letak dan properti komponen dilakukan secara dinamis tanpa ketergantungan pada skema basis data yang kaku, sehingga sangat sesuai untuk editor visual dan sistem konfiguratif [11].

Pemanfaatan JSON juga menjadi fondasi utama dalam implementasi Headless CMS, di mana seluruh konfigurasi konten dan tata letak disimpan sebagai representasi data yang dapat diolah dan dirender ulang oleh berbagai klien frontend [1], [4]. Pendekatan ini memungkinkan sinkronisasi dua arah antara antarmuka visual dan data backend, sehingga setiap perubahan yang dilakukan melalui UI dapat langsung direfleksikan ke dalam struktur data secara real-time [7], [11].

Meskipun berbagai kegiatan telah membahas penerapan antarmuka interaktif, drag-and-drop, serta penggunaan JSON dalam pengembangan aplikasi web modern, masih terdapat keterbatasan kajian yang secara spesifik mengintegrasikan mekanisme drag-and-drop dengan struktur data JSON yang bersifat schema-agnostic dalam konteks Headless CMS [8], [10], [12]. Selain itu, tantangan dalam menjaga konsistensi struktur data, mendukung tata letak bersarang, serta memastikan responsivitas antarmuka masih menjadi permasalahan yang relevan untuk diteliti lebih lanjut [6], [13].

Berdasarkan latar belakang tersebut, kegiatan ini berfokus pada implementasi dan pengujian antarmuka drag-and-drop berbasis pustaka DND Kit yang terintegrasi dengan struktur data JSON untuk mengelola tata letak halaman web secara dinamis pada platform WCMS berbasis Headless. Pendekatan yang diusulkan diharapkan mampu memberikan solusi praktis dalam pengembangan editor visual modern yang fleksibel, adaptif, dan mudah dikembangkan, serta mendukung kebutuhan pengelolaan konten pada aplikasi web skala menengah hingga besar.

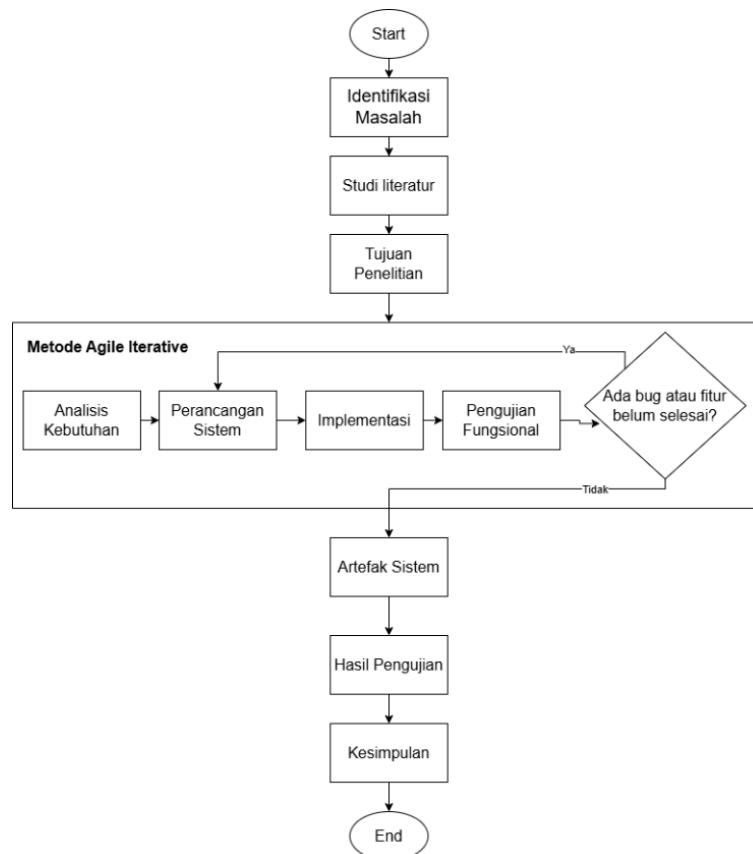
2. METODOLOGI

Fokus utama dari kegiatan ini adalah perancangan dan implementasi artefak perangkat lunak yang fungsional. Proses realisasinya diatur melalui siklus hidup pengembangan sistem (SDLC) dengan mengadopsi metodologi Agile Iterative, yang memungkinkan pengembangan sistem dilakukan secara bertahap dan adaptif berdasarkan evaluasi berkelanjutan terhadap kebutuhan pengguna [14].

Berbeda dengan model pengembangan linear seperti *waterfall*, metodologi Agile Iterative menekankan pengembangan berulang (*iterative development*) melalui siklus pendek (*timebox*) sehingga perubahan kebutuhan teknis dan penyempurnaan antarmuka pengguna dapat direspons secara cepat dan terkontrol [15].

Dengan menerapkan siklus ini, potensi kesalahan khususnya pada logika manipulasi struktur data JSON yang kompleks dapat diidentifikasi dan diperbaiki pada setiap iterasi tanpa harus menunggu hingga tahap akhir pengembangan. Hal ini memastikan bahwa artefak perangkat lunak yang dihasilkan memiliki kualitas yang terjaga. Sebagaimana diilustrasikan pada Gambar 1., alur metodologi diawali dengan inisiasi masalah dan studi literatur, yang kemudian diproses melalui serangkaian iterasi berulang mulai dari analisis, perancangan, hingga pengujian untuk memfasilitasi perbaikan fitur secara dinamis.

Landasan ini kemudian ditindaklanjuti dengan penerapan siklus Agile Iterative, di mana proses pengembangan tidak berjalan secara linear satu arah, melainkan melalui serangkaian iterasi berulang. Pendekatan ini dipilih untuk memfasilitasi mekanisme umpan balik yang berkelanjutan, memungkinkan perbaikan fitur secara dinamis mulai dari analisis, perancangan, hingga pengujian.



Gambar 1 Alur Metodologi Pengembangan Sistem Berbasis Agile Iterative

Untuk menerapkan metodologi pada Gambar 1., kegiatan ini dijabarkan ke dalam beberapa tahapan: Mengacu pada alur metodologi pengembangan yang tersaji dalam Gambar 1., studi ini menguraikan proses implementasi ke dalam tahapan-tahapan sistematis sebagai berikut:

1. Tahap Awal dan Studi Literatur

Tahap ini diawali dengan identifikasi masalah terkait pengelolaan tata letak dinamis, dilanjutkan dengan studi literatur untuk memahami teknologi yang relevan (seperti Headless CMS dan DnD Kit). Hasil dari tahap ini adalah perumusan tujuan kegiatan yang menjadi landasan pengembangan sistem.

Dalam tahap studi literatur, analisis mendalam dilakukan terhadap tantangan interoperabilitas data pada arsitektur Headless CMS. Literatur menunjukkan bahwa pendekatan tradisional sering kali memisahkan struktur data dari representasi visual secara kaku, yang menghambat fleksibilitas pengembangan antarmuka. Oleh karena itu, kajian difokuskan pada pemanfaatan struktur data JSON sebagai solusi penyimpanan yang skema-agnostik (schema-agnostic) dan integrasi pustaka DnD Kit yang mendukung manipulasi Document Object Model (DOM) secara virtual. Pemilihan teknologi ini didasarkan pada kebutuhan untuk menciptakan mekanisme sinkronisasi dua arah yang efisien antara antarmuka visual pengguna dan struktur data di sisi backend, yang menjadi landasan teoretis utama dalam pengembangan ini.

2. Analisis Kebutuhan

Sebagai langkah awal dalam siklus Agile Iterative, tahap ini berfokus mendefinisikan kebutuhan fungsional (FR) sistem secara mendetail. Berbeda dengan tahap awal, fase ini secara spesifik memetakan fitur-fitur editor tata letak yang akan dibangun berdasarkan tujuan yang telah ditetapkan sebelumnya.

3. Perancangan Sistem

Pada tahap ini, dilakukan perancangan arsitektur fitur. Fokus utamanya adalah merancang skema struktur data JSON yang mampu merepresentasikan tata letak bersarang (nested) serta merancang desain antarmuka (UI) dan alur interaksi pengguna untuk fungsionalitas drag-and-drop.

4. Implementasi Tahap implementasi

Merupakan proses penerjemahan hasil perancangan ke dalam kode fungsional menggunakan teknologi yang telah dipilih. Fitur-fitur seperti manipulasi state dan interaksi antarmuka dibangun pada tahap ini.

5. Pengujian Fungsional dan Evaluasi Iteratif

Untuk memvalidasi fungsionalitas artefak, dilakukan pengujian Black Box Testing. Sesuai dengan metode Agile Iterative, tahap ini mencakup gerbang keputusan: jika ditemukan bug atau fitur yang belum sesuai (kondisi "Ya"), alur akan kembali ke tahap perancangan sistem untuk direvisi. Jika sistem valid (kondisi "Tidak"), proses berlanjut ke tahap berikutnya.

6. Tahap Akhir (Artefak dan Kesimpulan)

Setelah lolos pengujian, sistem ditetapkan sebagai artefak sistem yang final. Data dari pengujian dianalisis menjadi hasil pengujian untuk kemudian ditarik kesimpulan mengenai keberhasilan implementasi antarmuka drag-and-drop berbasis JSON tersebut.

7. Perhitungan Tingkat Keberhasilan

Untuk mengukur tingkat keberhasilan fungsional dari sistem yang diuji, digunakan rumus perhitungan persentase keberhasilan yang didefinisikan berdasarkan variabel a , b , dan c pada persamaan (1) sebagai berikut:

$$a = \left(\frac{b}{c}\right) \times 100 \% \quad (1)$$

dengan:

- a : Persentase keberhasilan
- b : Jumlah skenario uji lulus
- c : Total skenario uji

Tabel 1. Daftar peserta pelatihan yang sudah menyelesaikan tugas

Nama Peserta	Jumlah Tugas	Penyelesaian Tugas	Skor
R. Budiraharjo	2.850	1 Agustus 2025	756
M.M. Barmawi	3.987	5 Oktober 2025	658
N.F. Fachrudin	2.654	8 Mei 2025	812
K.R. Putra	1.876	12 September 2025	799
S. Umaroh	4.443	14 Agustus 2025	801
A.R. Nurjaman	3.768	10 Oktober 2025	832
C. Caromawati	2.512	1 November 2025	648

3. HASIL DAN PEMBAHASAN

3.1 Analisis Kebutuhan

Tahap ini berfokus pada identifikasi dan pendefinisian kebutuhan fungsional dari sistem editor halaman yang akan dibangun. Kebutuhan fungsional ini menjadi acuan untuk perancangan, implementasi, dan pengujian. Berdasarkan analisis terhadap studi kasus Coofis Verse, kebutuhan fungsional utama sebagai berikut:

- FR-01: Manajemen Halaman dan Data Sistem harus dapat mengambil, menampilkan, dan mengelola data dasbor secara asinkron. Administrator dapat membuat, menghapus, dan memilih halaman aktif dengan ID unik.
- FR-02: Manajemen Layout dan Komponen Sistem harus memungkinkan Administrator untuk menambah/menghapus kontainer layout, menambah/menghapus grid (termasuk *nested grid*), menerapkan template layout, dan mengatur komponen UI dengan *drag-and-drop* beserta penyimpanan perubahan urutan.
- FR-03: Properti dan Responsivitas Komponen Sistem harus memungkinkan Administrator untuk menambah/mengganti komponen UI dalam grid, memperbarui properti komponen (ukuran, tinggi, konten), dan otomatis menyesuaikan serta menyinkronkan tinggi kontainer berdasarkan komponen anaknya.
- FR-04: Pratinjau dan Publikasi Sistem harus dapat memuat dan menampilkan pratinjau halaman dengan simulasi tampilan *multi-device* (desktop, tablet, mobile), mempublikasikan halaman agar dapat diakses publik, dan memperbarui menu navigasi secara dinamis.
- FR-05: Notifikasi dan Navigasi Pengguna Sistem harus menampilkan pesan alert (sukses, error, warning, info), menampilkan "Page not found" untuk halaman tidak ditemukan, dan memungkinkan navigasi ke halaman profil Administrator.

3.2 Perancangan Sistem

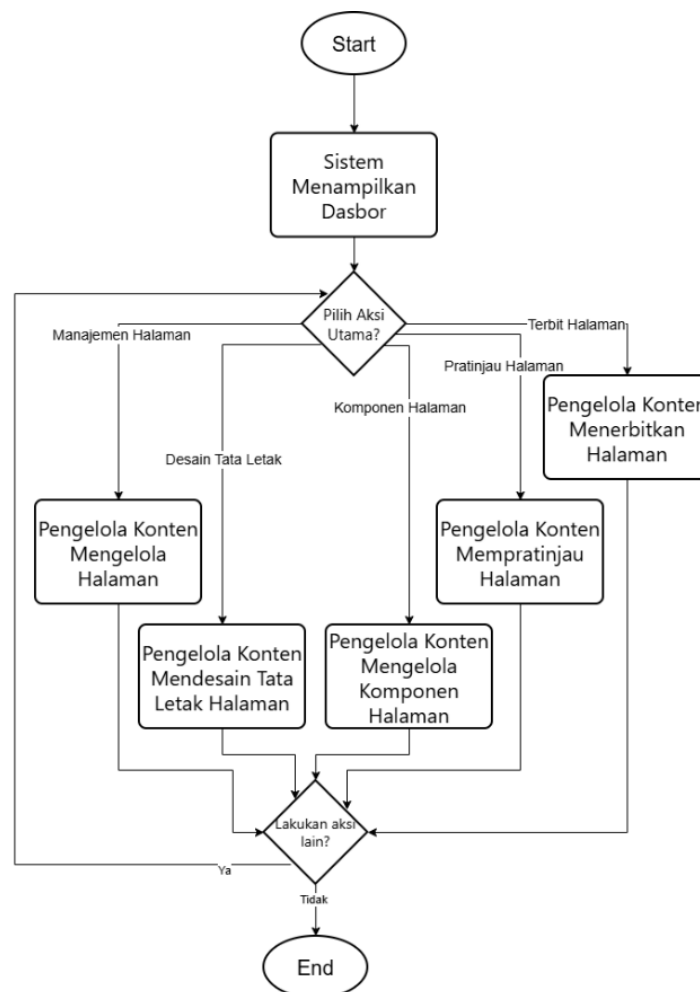
Perancangan sistem pada kegiatan ini difokuskan pada perancangan arsitektur editor visual yang mampu mendukung pengelolaan tata letak halaman secara dinamis. Tahap ini mencakup perancangan alur interaksi pengguna, antarmuka drag-and-drop, serta struktur data JSON yang digunakan untuk merepresentasikan hierarki halaman, tata letak, dan komponen secara bersarang (*nested*). Desain sistem dirancang agar setiap interaksi pengguna pada antarmuka, seperti penambahan, penghapusan, dan penyusunan ulang komponen, dapat secara langsung disinkronkan ke dalam struktur data JSON sehingga perubahan tata letak dapat dikelola secara konsisten dan responsif.

Pada tahap perancangan sistem ini juga ditetapkan pemisahan tanggung jawab antara lapisan antarmuka pengguna (*frontend*) dan representasi data tata letak, di mana antarmuka hanya berfungsi sebagai pemicu interaksi visual sementara seluruh konfigurasi halaman disimpan dalam struktur data JSON. Setiap elemen tata letak dan komponen dirancang memiliki identitas unik serta properti yang dapat dikonfigurasi untuk mendukung responsivitas multi-perangkat. Pendekatan ini memungkinkan sinkronisasi dua arah antara tampilan visual dan data, sehingga perubahan yang terjadi pada antarmuka dapat langsung direfleksikan pada struktur data tanpa ketergantungan terhadap skema basis data tertentu, serta mendukung integrasi sistem dalam arsitektur Headless CMS secara fleksibel dan terukur.

1. Flowchart Diagram

Untuk mengilustrasikan alur kerja sistem secara garis besar, disajikan diagram alur (*flowchart*) pada Gambar 2. Diagram ini menunjukkan navigasi tingkat tinggi (*high-level*) bagi pengguna dengan peran sebagai Pengelola Konten.

Dalam merancang interaksi *drag-and-drop*, sistem menerapkan logika validasi hierarkis. Ketika pengguna menyeret sebuah komponen (*drag start*), sistem mengidentifikasi tipe elemen tersebut dan membatasi area *drop* yang valid untuk mencegah kesalahan struktur (misalnya, mencegah penempatan kontainer di dalam komponen teks). Saat elemen dilepaskan (*drag end*), sistem melakukan kalkulasi posisi untuk menentukan apakah elemen tersebut harus menempati urutan baru dalam *array* tata letak atau bersarang (*nested*) di dalam komponen lain. Logika ini memastikan bahwa integritas struktur JSON tetap terjaga meskipun pengguna melakukan manipulasi visual yang kompleks secara berulang.



Gambar 2. Flowchart Diagram

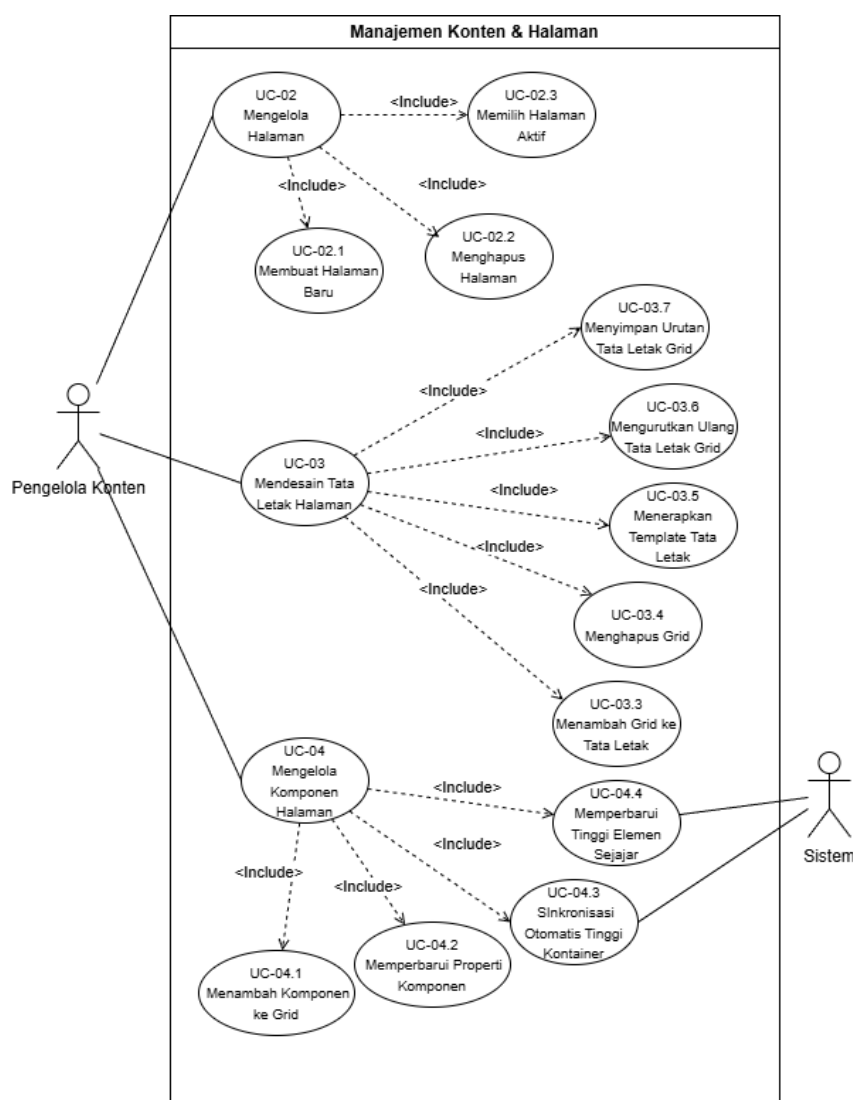
Gambar 2 mengilustrasikan alur kerja utama bagi Pengelola Konten. Proses dimulai saat Sistem Menampilkan Dasbor, yang berfungsi sebagai navigator utama.

Dari dasbor, pengguna dapat memilih salah satu modul fungsional, yaitu 'Manajemen Halaman', 'Desain Tata Letak', 'Komponen Halaman', 'Pratinjau Halaman', atau 'Terbit Halaman'. Setelah setiap proses selesai, alur menyediakan titik keputusan yang memungkinkan pengguna untuk kembali ke dasbor (menciptakan *session loop*) atau mengakhiri sesi (End). Selain menggambarkan alur navigasi utama, flowchart pada Gambar 2 juga merepresentasikan mekanisme kontrol alur sistem yang dirancang untuk mendukung proses

pengelolaan halaman secara iteratif dan terstruktur. Setiap modul fungsional yang dipilih oleh Pengelola Konten memicu serangkaian proses sistem yang saling terintegrasi, mulai dari pemrosesan data, pembaruan struktur tata letak, hingga penyimpanan perubahan. Titik keputusan yang disediakan setelah setiap proses memungkinkan sistem untuk menjaga kontinuitas sesi kerja pengguna, sehingga Pengelola Konten dapat melakukan serangkaian aktivitas pengelolaan halaman secara berulang tanpa harus keluar dari sistem, sekaligus memastikan alur kerja berjalan efisien dan terkontrol.

2. Use Case Diagram

Untuk memvisualisasikan interaksi fungsional antara pengguna dan sistem secara lebih terstruktur, dirancanglah sebuah Diagram Use Case yang memetakan batasan sistem serta hubungan ketergantungan antar fitur. Diagram ini memfokuskan pada interaksi aktor utama, yaitu Pengelola Konten, dengan tiga modul fungsionalitas inti sistem: manajemen halaman, manipulasi tata letak (layout), dan konfigurasi komponen. Representasi visual dari spesifikasi fungsional tersebut disajikan pada Gambar 3 berikut.



Gambar 3 Use Case Diagram

Berdasarkan Gambar 3, dapat diuraikan bahwa sistem membagi kewenangan aktor pengelola konten ke dalam tiga klaster aktivitas utama. Pertama, pada klaster mengelola halaman, aktor memiliki kendali penuh untuk membuat, menghapus, dan memilih halaman aktif yang akan disunting. Kedua, klaster mendesain tata letak halaman merepresentasikan inti dari fitur interaktif, di mana relasi *include* menunjukkan bahwa setiap aksi penambahan atau penghapusan kontainer dan *grid* secara inheren akan melibatkan proses penyimpanan urutan tata letak baru ke dalam basis data. Ketiga, klaster mengelola komponen halaman menangani detail

elemen visual, di mana sistem dirancang untuk merespons secara otomatis (sisi *System*) terhadap perubahan properti komponen, seperti melakukan sinkronisasi tinggi kontainer (*auto-height*) agar tampilan tetap responsif sesuai konten yang dimasukkan.

3. Struktur Data (JSON)

Untuk mendukung fungsionalitas tata letak dinamis yang dijelaskan pada flowchart, diperlukan struktur data yang fleksibel. Sesuai dengan fokus kegiatan, dapat dilihat pada Tabel 1 sistem ini dirancang menggunakan format JSON untuk merepresentasikan seluruh hierarki halaman, tata letak, dan komponen. Struktur ini memungkinkan skema yang dinamis di mana properti dan susunan elemen dapat diubah tanpa mengubah struktur basis data.

Tabel 1. Struktur JSON Page

Level	Key (Atribut)	Nama	Deskripsi
Page	id	UUID (String)	Identifikasi unik halaman
	Layouts	Array of Objects	Menampung daftar layout di dalam halaman
Container	properties	Object	Menyimpan konfigurasi responsif (grid system)
	children	Array of Objects	Menampung komponen bersarang (nested components)
Component	type	String	Menentukan jenis widget

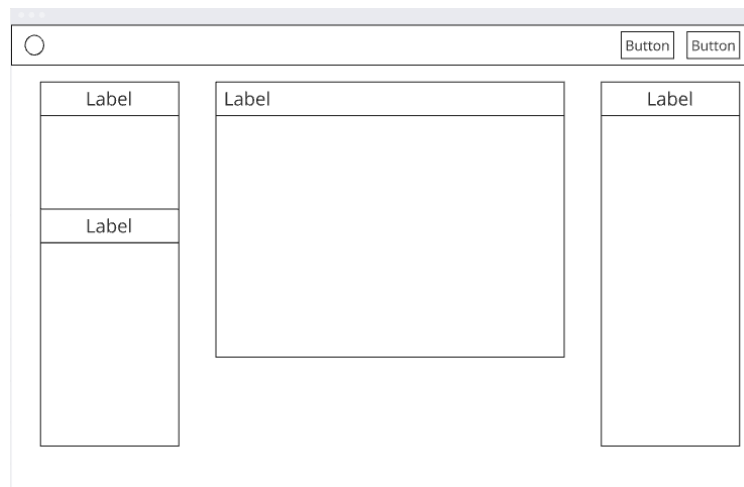
Tabel 1. merepresentasikan struktur data JSON yang digunakan untuk membangun halaman secara dinamis. Struktur ini menerapkan konsep hierarki bersarang (nested hierarchy) dimulai dari level teratas 'Page' yang memiliki atribut identitas unik. Di dalamnya, terdapat objek 'Layout' dan 'Container' yang berfungsi sebagai pembungkus struktur tata letak, dilengkapi dengan properti responsif (seperti size untuk desktop, tablet, dan mobile) serta pengaturan gaya (seperti height). Level terdalam berisi objek 'Komponen' (misalnya Navbar) yang menyimpan properti spesifik elemen antarmuka, seperti judul atau konten teks. Fleksibilitas struktur ini memungkinkan sistem untuk merender susunan halaman yang kompleks dan adaptif tanpa memerlukan perubahan skema basis data. Tabel 2. merupakan Snippet JSON page yang akan ditambah jika menambah layout atau grid.

Tabel 2 Snippet JSON

Snippet JSON
<pre>{ "id": "UUID_GENERATED", "type": "container", "properties": { "responsive": { "desktop": "Integer (1-12)", "mobile": "Integer (1-12)" } }, "children": [{ "type": "component_name", "data": "Dynamic Content Object" }] }</pre>

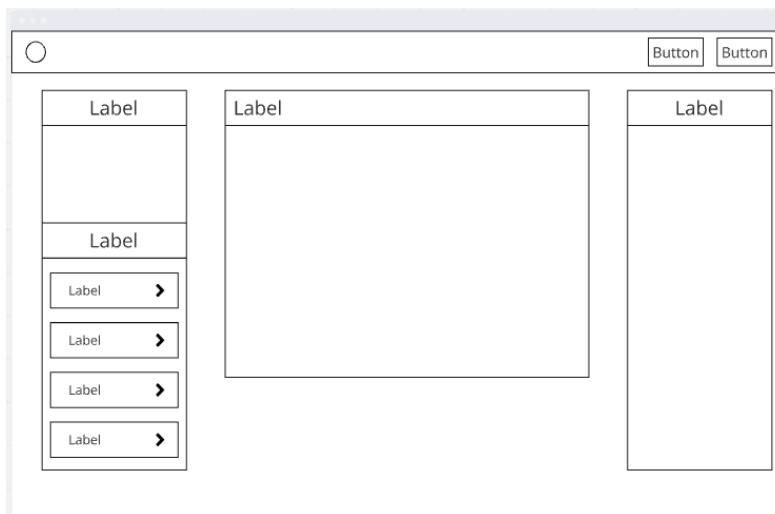
4. Perancangan Antarmuka Pengguna (*User Interface Design*)

Perancangan antarmuka pengguna (User Interface) difokuskan pada kemudahan interaksi drag-and-drop untuk memanipulasi tata letak halaman. Desain antarmuka utama sistem, yaitu halaman Editor, ditunjukkan pada Gambar 4.



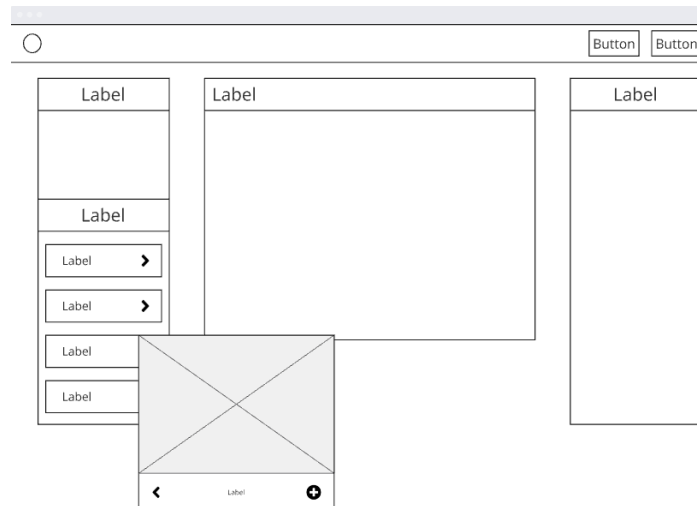
Gambar 4. Wireframe Editor

Terlihat pada Gambar 4., antarmuka editor dibagi menjadi tiga area utama yang merefleksikan struktur data sistem panel kiri (Draggable Area): Menyediakan daftar komponen dan layout yang dapat diseret pengguna. Area ini juga menampilkan struktur hierarki (tree view) dari elemen yang sedang aktif. Kanvas utama (Droppable Area): Area kerja visual tempat pengguna melepaskan komponen dan menyusun tata letak secara real-time. Panel kanan (Properties Editor): Formulir dinamis untuk mengubah atribut komponen (seperti ukuran grid, teks, atau warna) yang akan langsung memperbarui nilai properti pada objek JSON.



Gambar 5 Wireframe Component Editor

Gambar 5. memperlihatkan antarmuka component editor yang memungkinkan pengguna mengonfigurasi properti spesifik dari sebuah komponen secara mendalam. Panel sebelah kanan secara dinamis menampilkan formulir pengaturan yang relevan dengan jenis komponen yang dipilih, di mana setiap perubahan properti akan langsung direfleksikan pada tampilan pratinjau di tengah layar.



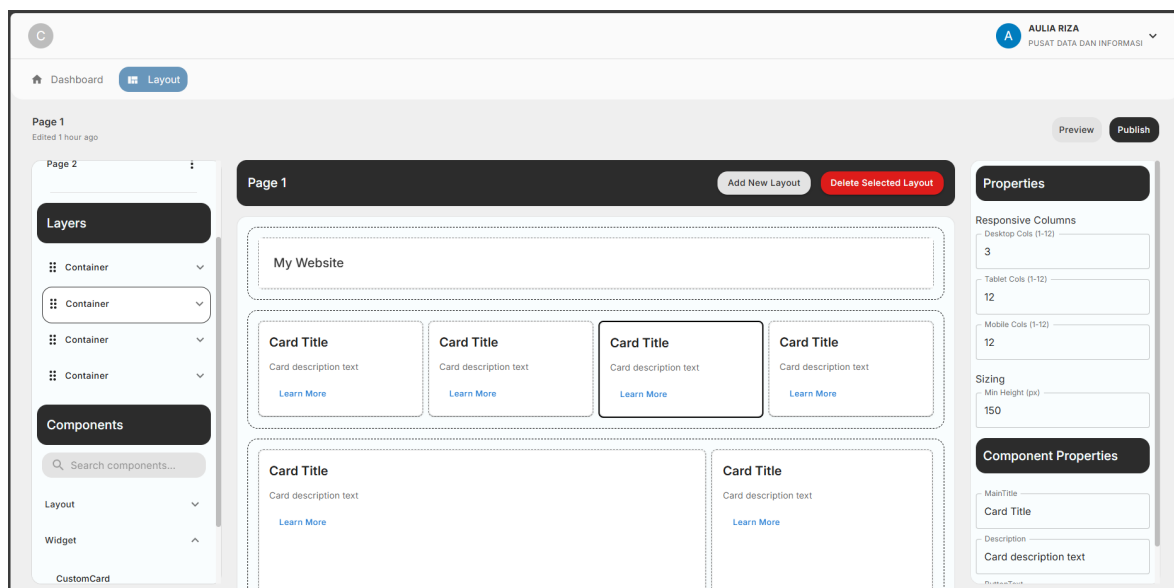
Gambar 6 Wireframe Add Widget

Gambar 6. menunjukkan antarmuka *Add Widget* yang dapat diakses melalui panel kiri. Pengguna disajikan dengan daftar kategori *widget* yang tersedia. Antarmuka ini dirancang untuk memudahkan pengguna mencari dan memilih *widget* yang diinginkan, untuk kemudian ditempatkan ke dalam area kerja menggunakan interaksi *drag-and-drop*.

3.3 Hasil Implementasi

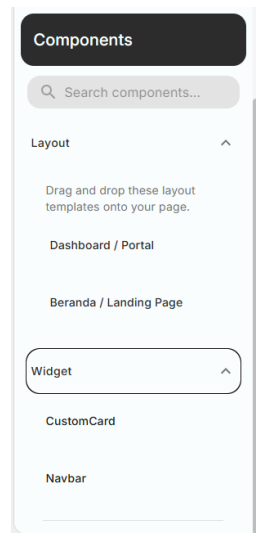
1. Halaman Editor (Workspace)

Halaman Editor merupakan orkestrator utama yang mengintegrasikan drag-and-drop dan manajemen state global. Seperti terlihat pada Gambar 7., halaman ini dibagi menjadi tiga area kerja utama: panel komponen (draggable), kanvas kerja (droppable), dan panel properti.



Gambar 7. Halaman Editor

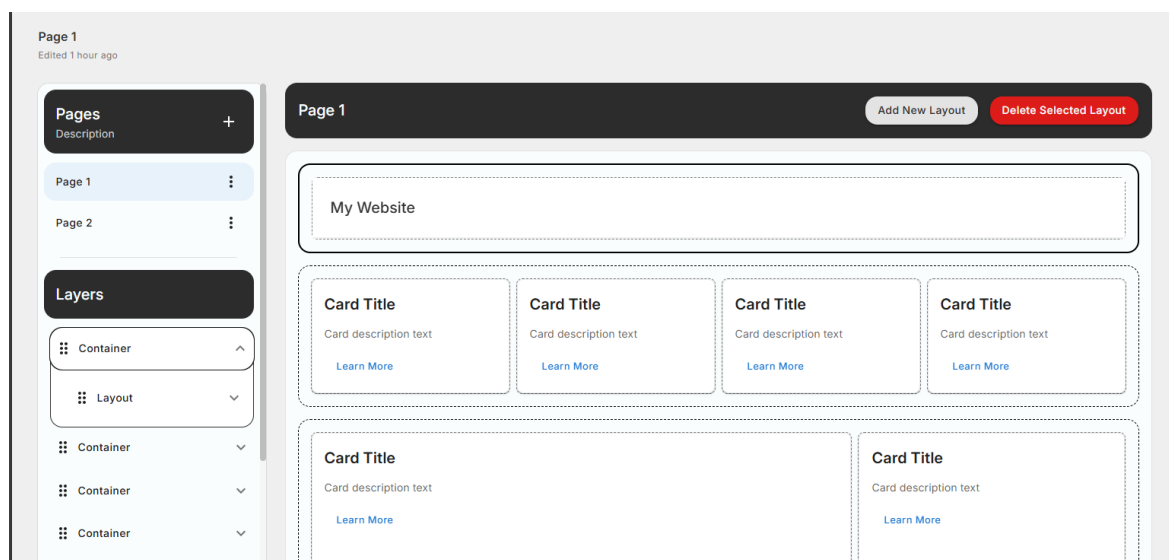
Pada Gambar 7., pengguna dapat menarik komponen dari menu kiri dan meletakkannya di area kanvas. Implementasi ini menggunakan pembungkus `DndContext` yang mengelola kejadian *drag start* dan *drag end* untuk memperbarui struktur JSON di *Redux store* secara *real-time*.



Gambar 8. Component Section Left Menu

Pada Gambar 8, pengguna dapat menarik item dari panel komponen di menu kiri dan meletakkannya di area kanvas, panel ini mengkategorikan item yang dapat ditarik menjadi dua jenis utama yaitu Layout yang Berisi *template* tata letak yang sudah jadi (misalnya "Dashboard / Portal" atau "Beranda / Landing Page") yang dapat langsung diterapkan ke halaman. Lalu Widget yang berisi komponen UI individual (seperti "CustomCard" atau "Navbar") yang dapat dimasukkan ke dalam sebuah *layout* yang sudah ada di kanvas.

Tepat di bawah panel Pages adalah panel 'Layers', yang esensial untuk manajemen struktur halaman. Seperti yang ditunjukkan Gambar 9., panel ini secara dinamis merender hierarki (*tree view*) dari semua elemen (misalnya 'Container' dan 'Layout') yang telah ditempatkan pengguna di kanvas utama.



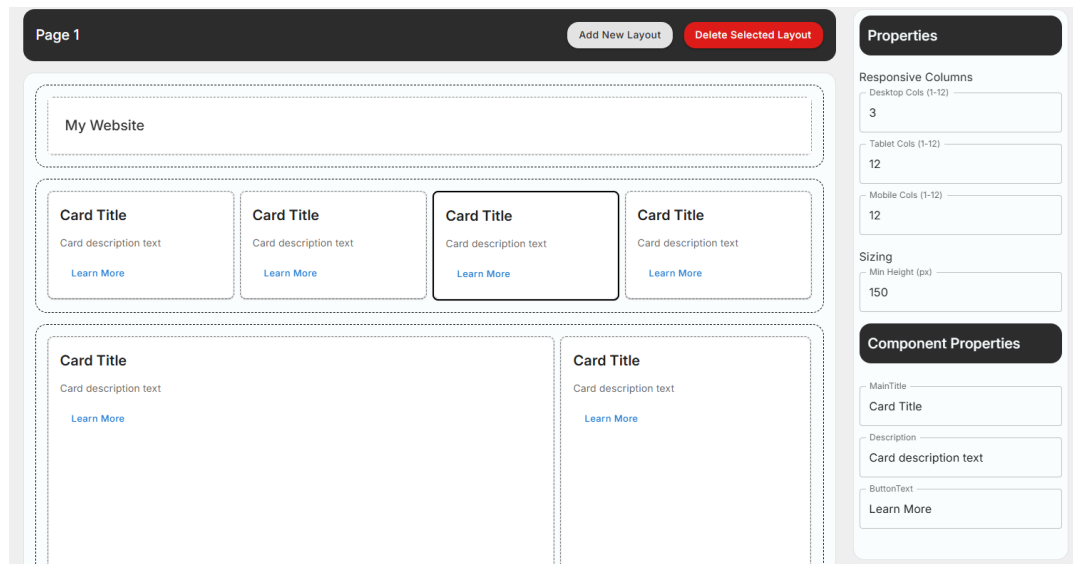
Gambar 9. Layers Section Left Menu

Panel 'Layers' ini menyediakan interaksi dua arah. Pengguna dapat mengklik sebuah elemen di panel 'Layers' untuk secara otomatis memilih dan menyorot (*select*) elemen tersebut di kanvas. Selain itu, panel ini juga

mendukung fungsionalitas *drag-and-drop* internal, memungkinkan pengguna untuk mengatur ulang (*reorder*) posisi tata letak langsung dari *tree view* tersebut.

2. Manajemen Properti Komponen

Untuk mendukung dinamisasi konten, setiap komponen yang dipilih di kanvas dapat dikonfigurasi melalui panel kanan (Right Menu).

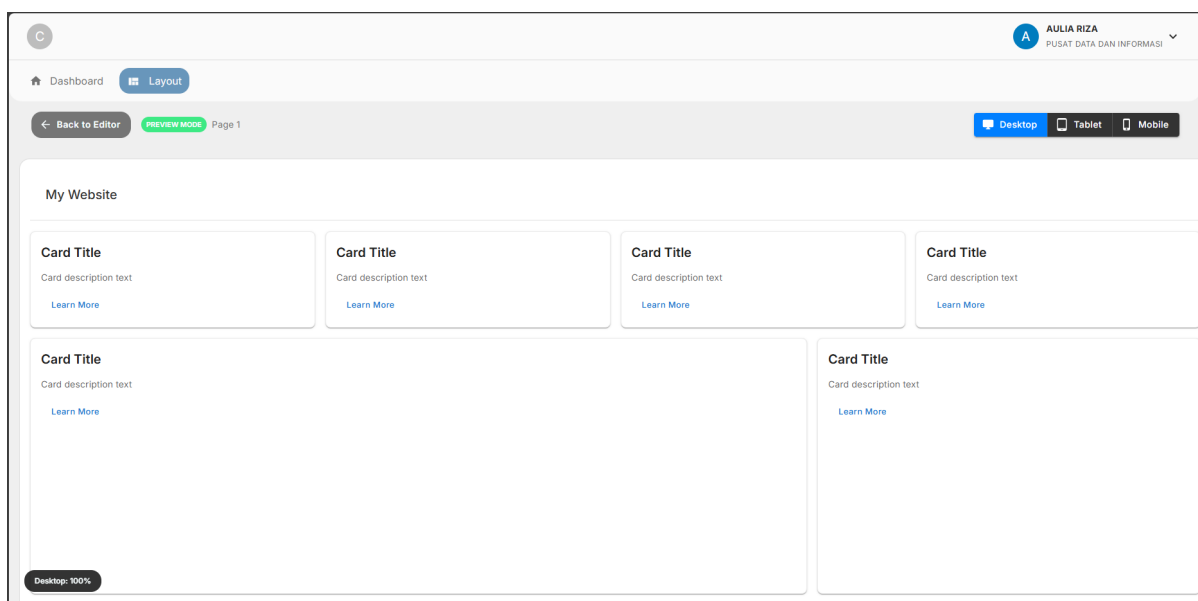


Gambar 10 Component Editor Page

Gambar 10 menunjukkan formulir dinamis yang memungkinkan pengguna mengubah atribut seperti ukuran grid (untuk desktop, tablet, mobile), tinggi kontainer, dan konten teks. Setiap perubahan pada *input* ini akan memicu pembaruan langsung pada objek JSON yang merepresentasikan halaman tersebut.

3. Fitur Pratinjau (Preview)

Sebelum diterbitkan, halaman dapat dipratinjau untuk memastikan responsivitasnya.

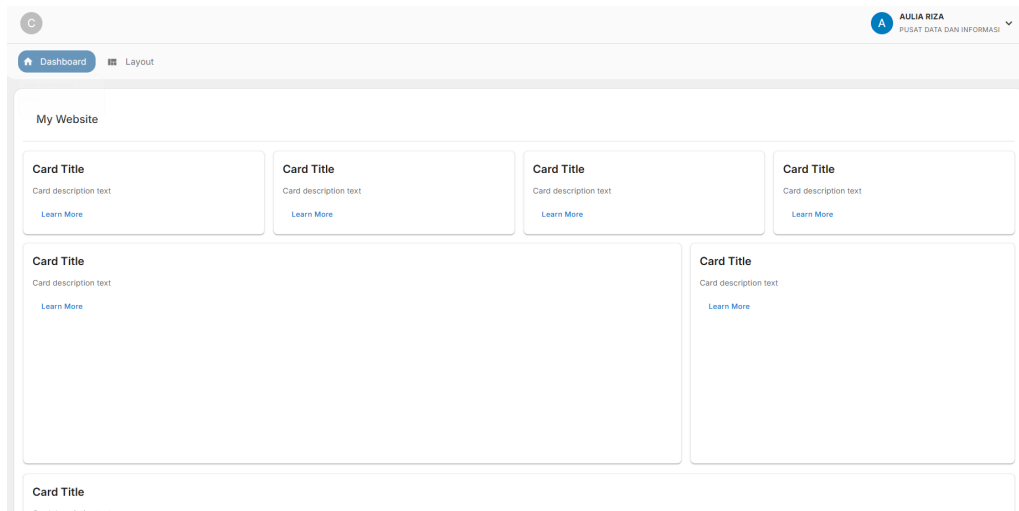


Gambar 11 Preview Page

Berikut pada Gambar 11 diatas memperlihatkan fitur pratinjau yang merender ulang struktur JSON tanpa elemen editor. Sistem menyediakan simulasi tampilan perangkat (Desktop, Tablet, Mobile) untuk memvalidasi bahwa tata letak grid 1-12 kolom berfungsi sesuai dengan desain responsif yang diinginkan.

4. Halaman Dasbor

Halaman dasbor berfungsi sebagai pusat informasi bagi pengelola konten untuk melihat ringkasan proyek.



Gambar 12 Dashboard Page

Berikut pada Gambar 12 diatas menampilkan implementasi dasbor yang memuat data secara asinkron. Komponen kartu (*card*) dirender secara dinamis berdasarkan data yang diterima dari API, memberikan gambaran status sistem kepada pengguna.

3.4 Pengujian Sistem

Untuk memvalidasi fungsionalitas sistem yang telah diimplementasikan, dilakukan pengujian fungsional menggunakan metode *Black Box Testing*. Pendekatan ini dipilih untuk memverifikasi kesesuaian *input* dan *output* sistem berdasarkan kebutuhan fungsional (FR) yang telah didefinisikan, tanpa melihat struktur kode internal.

Pengujian difokuskan pada skenario uji yang mencakup fungsionalitas inti sistem, terutama yang berkaitan dengan fokus kegiatan. Skenario ini meliputi manajemen halaman, implementasi *drag-and-drop* untuk tata letak, dan proses pratinjau serta publikasi halaman.

1. Hasil Pengujian Fungsional

Validasi fungsionalitas sistem dilakukan melalui serangkaian pengujian menggunakan metode Black Box Testing. Pengujian ini difokuskan pada verifikasi input dan output untuk memastikan bahwa setiap fitur berjalan sesuai dengan spesifikasi kebutuhan fungsional yang telah dirancang. Dari total 19 skenario uji yang dieksekusi, Tabel 3. berikut menyajikan sampel representatif yang mencakup fitur-fitur krusial sistem, mulai dari manajemen tata letak, manipulasi komponen, hingga proses publikasi konten.

Tabel 3. Sampel Hasil Pengujian Black Box

ID	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Hasil Aktual
TC-004	Menambah tata letak baru	Tata letak baru muncul di area kerja	Tata letak berhasil ditambahkan	Lulus

ID	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Hasil Aktual
TC-010	Menambah komponen ke grid (Drag & Drop)	Komponen baru muncul di dalam grid	Komponen berhasil ditambahkan	Lulus
TC-015	Mempublikasikan halaman	Halaman dapat diakses melalui URL publik	Halaman berhasil dipublikasikan	Lulus
TC-001	Menampilkan data dashboard (Login)	Data ringkasan dashboard ditampilkan setelah login	Gagal login karena halaman Login belum tersedia	Gagal

Dari total 19 skenario uji (Test Case) yang dirancang dan dieksekusi, hasil pengujian fungsional dapat diringkas sebagai berikut total skenario Uji: 19 Skenario Berhasil (Lulus): 18, Skenario Gagal: 1

2. Perhitungan Tingkat Keberhasilan

Persamaan (2) menunjukkan hasil dari tingkat keberhasilan fungsional dari sistem yang diuji, digunakan rumus perhitungan persentase keberhasilan sebagai berikut:

$$persentase = \frac{18}{19} \times 100 \% = 94,7 \% \quad (2)$$

Sistem mencapai tingkat keberhasilan fungsional sebesar 94,7%.

4. KESIMPULAN

4.1 Kesimpulan

Implementasi antarmuka drag-and-drop pada studi kasus studi kasus platform Web Content Management System (WCMS) berbasis headless berhasil dilakukan dengan mengintegrasikan pustaka DND Kit dan struktur data JSON sebagai fondasi utama. Pendekatan ini terbukti efektif dalam menyediakan interaktivitas yang responsif serta mekanisme penyimpanan data yang mampu menangani hierarki halaman yang kompleks dan bersarang (nested). Dengan memanfaatkan fleksibilitas JSON, sistem memungkinkan representasi tata letak yang dinamis tanpa mengubah skema basis data, sehingga pengguna dapat menyusun elemen visual dan memanipulasi properti komponen secara mandiri tanpa memerlukan intervensi pemrograman langsung.

Validasi kinerja sistem dilakukan melalui pengujian Black Box Testing terhadap 19 skenario uji yang mencakup manajemen halaman dan manipulasi tata letak. Hasil pengujian menunjukkan tingkat keberhasilan fungsional sebesar 94,7%, dengan sistem mampu menjalankan 18 skenario uji dengan baik. Analisis ini mengonfirmasi bahwa seluruh fungsionalitas inti yang menjadi fokus kegiatan termasuk operasi menambah, menghapus, dan menyusun ulang komponen, serta sinkronisasi perubahan properti ke dalam struktur JSON telah berjalan akurat dan sesuai dengan kebutuhan fungsional yang dirancang.

Implementasi selanjutnya disarankan untuk memperluas evaluasi sistem dengan menambahkan pengujian non-fungsional, seperti pengujian performa dan usability, guna mengukur responsivitas dan kenyamanan penggunaan sistem pada struktur tata letak yang lebih kompleks. Selain itu, pengembangan dapat diarahkan pada penerapan mekanisme validasi struktur data JSON serta penambahan fitur lanjutan seperti versioning tata letak atau kolaborasi multi-user untuk meningkatkan skalabilitas dan penerapan sistem editor visual berbasis drag-and-drop pada lingkungan Headless CMS yang lebih luas.

DAFTAR PUSTAKA

- [1] J. Paulsson, “Examensarbete 30 hp Maj 2024 Code Generation for Efficient Web Development in Headless Architecture Civilingenjörskursprogrammet i informations teknologi,” 2024.
- [2] V. Jain, “Headless CMS and the Decoupled Frontend Architecture,” *Article in International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, 2021, doi: 10.5281/zenodo.14752509.
- [3] P. Matilla, “Headless-toteutuksen kehitys WordPress- ja Next.js-ympäristössä,” 2022.
- [4] X. Cao, “HEADLESS CMS AND QWIK FRAMEWORK and their practicalities in the future of application development School of Technology 2023 ACKNOWLEDGEMENTS,” 2023.
- [5] M. Raposo, S. Eloy, and M. S. Dias, “Bridging the gap in customised housing design: Integrating a graphic user interface for user collaboration,” *PLoS One*, vol. 19, no. 12, Dec. 2024, doi: 10.1371/journal.pone.0313291.
- [6] M. Somi, “POLITECNICO DI TORINO User Interface Development of a Modern Web Application Supervisors Prof. Luca ARDITO Candidate Marzieh SOMI,” 2021.
- [7] L. Lehtinen, “Improving the user experience of a JSON Editor-based user management tool,” 2023.
- [8] L. Seppänen and E. Hietaranta, “Developing a content management system with drag and drop functionality Subject Developing a content management system with drag and drop functionality,” 2022.
- [9] S. Wijanarko, “Integrasi SortableJS, Drawflow, dan InteractJS untuk Membangun Antarmuka Aplikasi Interaktif Berbasis Web,” in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, 2025, pp. 1297–1302. doi: 10.1145/1979742.1979764.
- [10] Y. Jiang, W. Stuerzlinger, and C. Lutteroth, “Reverseorc: Reverse engineering of resizable user interface layouts with or-constraints,” in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, May 2021. doi: 10.1145/3411764.3445043.
- [11] A. Di Stefano, M. Fadda, C. Marini, F. Ramzan, and D. Reforgiato Recupero, “Efficient translation of HTML to JSON for enhanced web content production,” *Multimed. Tools Appl.*, vol. 85, no. 1, p. 19, Jan. 2026, doi: 10.1007/s11042-026-21232-7.
- [12] P. P. Ray, “A Review on Vibe Coding: Fundamentals, State-of-the-art, Challenges and Future Directions,” May 09, 2025. doi: 10.36227/techrxiv.174681482.27435614/v1.
- [13] S. Larsen-Disney, *Elevating React Web Development with Gatsby*. 2022.
- [14] L. dela Paz de Mesa, N. Aldrin Peñalosa Clores, M. P. Landrito Lorenzo, and M. Jho Emmanuel Cabrera Salvia, “CT Journeys: A Web-Based Workflow System for Streamlining Integrative Course Processes,” in *2025 23rd International Conference on ICT and Knowledge Engineering (ICT&KE)*, IEEE, Nov. 2025, pp. 1–6. doi: 10.1109/ICTKE67052.2025.11274444.
- [15] VersionOne Inc., “15th State of Agile Report.” Accessed: Feb. 08, 2026. [Online]. Available: <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>