



IMPLEMENTASI ANTARMUKA WEBSITE RESPONSIF MENGGUNAKAN FRAMEWORK ROOTS SAGE DAN TAILWIND CSS PADA WEBSITE KAJIANMU

Muhamad Bintang Rukmana¹⁾, Adrian Reza²⁾

^{1,2} Program Studi Teknik Informatika, Universitas Nusa Putra

Corresponding Author: ¹muhamad.bintang_ti22@nusaputra.ac.id

Article Info

Article history:

Received: Jan 14, 2026

Revised: Jan 25, 2026

Accepted: Jan 31, 2026

Published: Feb 01, 2026

Keywords:

Front-End
WordPress
Roots Sage
Tailwind CSS
Responsive
Website

ABSTRACT

This research specifically discusses the stages and implementation process of the user interface (front-end) section applied to the Kajianmu website, a digital information platform. The main goal of this development activity is to be able to create a website interface that is not only visually modern, but also systematically structured, and has a high level of responsiveness to ensure content can be accessed comfortably and optimally through various types of user devices. The entire technical development process is carried out by modernizing the basic architecture of the WordPress theme by utilizing the Sage Roots framework. The use of this tool allows the implementation of the Laravel Blade template engine feature, which functions to create a much more modular, organized, and neat program code structure. In addition, Tailwind CSS technology is also intensively used as a styling framework that prioritizes utility to help speed up the interface creation process so that the results are precise and in accordance with the design that has been created. This development approach is implemented to replace traditional conventional theme development methods, with a primary focus on aspects of efficiency in writing code and precision of interface details. The final results of this study show evidence that the implementation of modern technology has succeeded in producing the Kajianmu website with a component structure that is very easy to manage (maintain), and displays a consistent visual appearance (pixel-perfect) both when opened on desktop screens and mobile devices.



This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY SA 4.0)

1. PENDAHULUAN

Perkembangan situs web saat ini menuntut peningkatan standar yang lebih tinggi, tidak hanya dalam aspek visual tetapi juga dalam efisiensi dan kemudahan pengelolaan kode. WordPress, sebagai sistem manajemen konten (CMS) yang paling banyak digunakan di seluruh dunia, sering menjadi pilihan utama karena fleksibel [7]. Namun, metode tradisional dalam mengembangkan tema WordPress seringkali menghadapi tantangan terkait struktur kode yang tidak teratur (spaghetti code) dan kendala performa akibat manajemen aset CSS/JS yang kurang optimal. Dalam konteks pengembangan platform informasi digital "Kajianmu", antarmuka pengguna memegang peranan krusial. Pengguna menuntut akses informasi yang cepat serta tampilan yang responsif baik di desktop maupun smartphone. Metode pengembangan konvensional kerap kesulitan memenuhi standar

modern ini, terutama ketika kebutuhan desain semakin kompleks dan dinamis [6].

Untuk mengatasi permasalahan struktur dan skalabilitas tersebut, diperlukan pendekatan arsitektur tema WordPress kontemporer yang memisahkan antara logika pemrograman dan tampilan. Penelitian ini menerapkan solusi tersebut menggunakan Roots Sage, sebuah starter theme yang mengadopsi prinsip Don't Repeat Yourself (DRY) dan arsitektur Wrapper. Berbeda dengan tema biasa, Sage membawa alur kerja pengembangan modern dengan mengintegrasikan alat manajemen dependensi seperti Composer, serta memisahkan logika data dari template tampilan agar struktur berkas lebih terorganisir layaknya framework aplikasi modern [1].

Sebagai bagian dari modernisasi alur kerja, Sage memanfaatkan Laravel Blade sebagai mesin templat (template engine). Blade memungkinkan pemisahan yang tegas antara logika PHP dan HTML melalui fitur

pewarisan (inheritance) dan komponen [11]. Dengan fitur ini, elemen berulang seperti header dan footer cukup didefinisikan satu kali dalam berkas induk, sehingga meningkatkan efisiensi penulisan kode dan mempermudah pemeliharaan (maintainability). Selain aspek struktur, efisiensi penataan gaya (styling) diimplementasikan menggunakan Tailwind CSS. Framework ini mengusung konsep utility-first, yang menyediakan kelas-kelas utilitas dasar (seperti flex, pt-4) untuk membangun antarmuka kustom secara cepat tanpa perlu menulis CSS tradisional yang berlebihan [2], [5]. Tailwind juga memiliki mekanisme purge yang secara otomatis menghapus kelas yang tidak terpakai di tahap produksi, menghasilkan berkas gaya yang sangat efisien. Hal ini krusial mengingat optimasi aset statis berpengaruh signifikan terhadap kecepatan muat website [13].

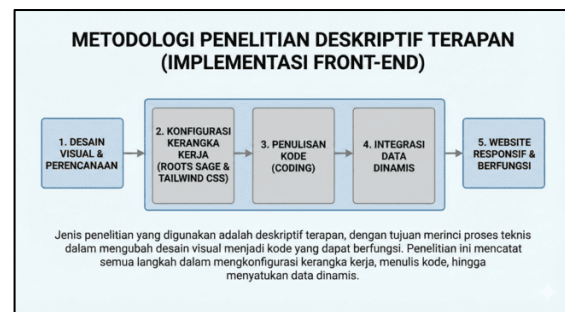
Penerapan teknologi tersebut bermuara pada strategi Desain Web Responsif dengan pendekatan Mobile-First. Pendekatan ini memprioritaskan pengaturan tampilan untuk perangkat seluler terlebih dahulu sebelum disesuaikan ke layar yang lebih besar menggunakan breakpoints yang disediakan Tailwind, menjamin pengalaman pengguna yang seragam di seluruh perangkat [12]. Tujuan utama dari penelitian ini adalah merancang dan mengimplementasikan antarmuka website Kajianmu yang tidak hanya modern dan responsif, tetapi juga dinamis melalui integrasi Advanced Custom Fields (ACF). Integrasi ini memungkinkan pengelolaan konten yang fleksibel oleh administrator tanpa mengganggu struktur kode. Penggunaan custom fields terbukti efektif dalam memperluas fungsionalitas CMS standar untuk memenuhi kebutuhan data yang spesifik [15]. Penelitian ini diharapkan dapat menjadi rujukan mengenai efektivitas peralihan dari pengembangan tema WordPress konvensional menuju metode berbasis komponen yang lebih terstruktur dan modern [3].

2. METODE PENELITIAN

Penelitian ini menerapkan metode pengembangan perangkat lunak berbasis komponen, yang merupakan strategi pelaksanaan yang menitikberatkan pada pemisahan antarmuka menjadi segmen-segmen modular yang bisa digunakan kembali. Sasaran utama yang ingin dicapai adalah menciptakan kode antarmuka situs web yang terorganisir dengan baik, terstruktur, dan memiliki standar kode yang bersih (clean code) yang sesuai dengan standar web terkini.

2.1 Pendekatan Penelitian

Jenis penelitian yang digunakan adalah deskriptif terapan, dengan tujuan merinci proses teknis dalam mengubah desain visual menjadi kode yang dapat berfungsi. Penelitian ini mencatat semua langkah dalam mengkonfigurasi kerangka kerja, menulis kode, hingga menyatukan data dinamis.



Gambar 2.1 Metodologi Penelitian

2.2. Implementasi Desain

Proses implementasi antarmuka web Kajianmu diawali dengan analisis aset desain untuk memahami struktur tata letak (layout), palet warna, dan tipografi yang diperlukan. Tahap teknis dimulai dengan Penggunaan lingkungan pengembangan lokal berbasis Docker dipilih untuk menjamin konsistensi sistem operasi dan database, yang terbukti mampu meningkatkan efisiensi alur kerja pengembangan aplikasi web [4]. Setelah lingkungan siap, dilakukan instalasi tema Roots Sage melalui Composer. Perancangan struktur kode dimulai dengan pengembangan komponen (slicing) menggunakan Blade Template Engine. Pada tahap ini, elemen antarmuka seperti header, footer, dan kartu konten dipetakan ke dalam file.blade.php yang terpisah agar sifatnya modular.

Selanjutnya, proses styling dilakukan menggunakan Tailwind CSS. Kelas utilitas diterapkan langsung pada komponen Blade untuk membentuk tampilan visual yang tepat. Pada tahap ini, penerapan responsive breakpoints juga dilakukan untuk memastikan tampilan dapat menyesuaikan dengan ukuran layar perangkat, baik desktop maupun mobile. Setelah tampilan statis selesai, langkah selanjutnya adalah mengintegrasikan data dinamis melalui Advanced Custom Fields (ACF). Kode Blade dihubungkan dengan field ACF sehingga konten gambar dan teks dapat disesuaikan melalui dasbor admin tanpa perlu mengedit kode.

2.3 Tools dan Perangkat yang Digunakan

Tools	Keterangan
Visual Studio Code	Editor kode utama yang digunakan untuk menulis sintaks PHP (Blade), JavaScript, dan konfigurasi Tailwind, didukung ekstensi untuk <i>formatting</i> kode otomatis.
Roots Sage 10 & Acorn	<i>Starter theme</i> dan pustaka pendukung yang berfungsi sebagai kerangka kerja utama, menyediakan struktur MVC dan integrasi Laravel di WordPress.
Tailwind CSS	Kerangka kerja CSS berbasis utilitas (<i>utility-first</i>) yang digunakan untuk mempercepat

	proses penataan gaya antarmuka yang responsif.
Docker	Platform kontainerisasi yang digunakan untuk menjalankan layanan <i>database</i> MySQL dan server WordPress secara lokal dan portabel.
Web Browser (Chrome)	Digunakan untuk melihat hasil <i>render</i> antarmuka dan melakukan <i>debugging</i> tampilan menggunakan fitur Developer Tools.

2.4 Tools dan Perangkat yang Digunakan

Tahapan	Tujuan
Konfigurasi Environment	Untuk menyiapkan lingkungan server dan database yang terisolasi menggunakan Docker agar proses pengembangan stabil dan konsisten.
Slicing Komponen (Blade)	Untuk memetakan struktur visual desain menjadi kode HTML yang modular, rapi, dan dapat digunakan kembali (<i>reusable</i>) menggunakan sistem <i>template</i> Blade.
Styling Antarmuka (Tailwind)	Untuk menerapkan gaya visual yang presisi sesuai desain dan memastikan tampilan responsif di berbagai ukuran layar (<i>mobile-first</i>).
Integrasi Data (ACF)	Untuk menghubungkan tampilan statis dengan database WordPress, sehingga konten teks dan gambar menjadi dinamis dan dapat dikelola admin.
Build & Optimasi	Untuk mengkompilasi aset pengembangan menjadi fail produksi yang ringan guna memaksimalkan kecepatan akses (<i>performance</i>) website.
Dokumentasi	Untuk mencatat konfigurasi teknis dan struktur kode sebagai panduan pemeliharaan sistem di masa mendatang.

3. HASIL DAN PEMBAHASAN

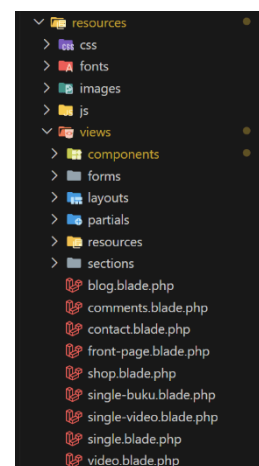
Penerapan antarmuka dalam situs "Kajianmu" telah dicapai melalui penelitian ini, dengan fokus pada pembangunan struktur kode yang terorganisir dan desain yang responsif. Sesuai dengan rencana pelaksanaan yang dijelaskan di bagian metode, langkah-langkah pengembangan dilakukan secara bertahap, dimulai dari pengaturan arsitektur tema,

pembuatan bagian-bagian (*slicing*), pengaturan gaya (*styling*), hingga penggabungan data yang dinamis.

3.1. Struktur Implementasi Tema

Tahapan implementasi diawali dengan inisiasi struktur tema menggunakan Roots Sage 10. Berbeda dengan tema WordPress konvensional, struktur direktori pada proyek ini memisahkan logika pemrograman dan tampilan antarmuka secara tegas. Pemisahan ini mengadopsi pola arsitektur Model-View-Controller (MVC) yang bertujuan untuk meningkatkan keterbacaan kode dan kemudahan pengembangan[14]. Folder *resources/views* digunakan untuk menyimpan seluruh berkas *template*.

Seperti yang terlihat pada gambar, struktur tampilan diorganisir ke dalam sub-direktori spesifik seperti *layouts* untuk tata letak utama, *components* untuk elemen UI yang dapat digunakan kembali, serta *sections* untuk bagian-bagian halaman. Hal ini memudahkan pengelolaan kode dibandingkan struktur tema tradisional yang datar (*flat*)



Gambar 3.1 Struktur Direktori *resources/views* yang Menganalisis Modularitas Blade Template

3.2. Implementasi Komponen Blade

Proses perancangan antarmuka dilakukan dengan menerapkan pendekatan desain atomik, di mana seluruh tampilan dipecah menjadi unit-unit komponen kecil yang modular menggunakan fitur Blade Template Engine. Pendekatan ini memungkinkan pengembang untuk mengisolasi elemen antarmuka yang bersifat repetitive seperti header, footer, sidebar, dan kartu konten kajian menjadi berkas komponen mandiri. Strategi ini tidak hanya memfasilitasi penggunaan ulang kode (*code reusability*) secara efisien di berbagai halaman, tetapi juga menjamin konsistensi visual di seluruh situs web serta mempermudah proses *maintenance* jika terjadi perubahan desain di kemudian hari.

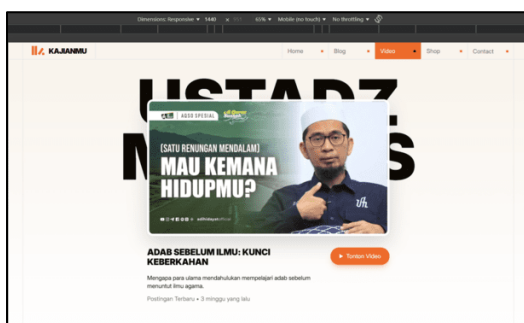


3.3. Hasil Tampilan Responsif

Setelah struktur komponen terbentuk, proses styling dilakukan menggunakan kerangka kerja Tailwind CSS. Pendekatan mobile-first terbukti efektif menjaga tata letak konten. Hal ini sejalan dengan penelitian Susanto [9] yang menyatakan bahwa penerapan Responsive Web Design (RWD) adalah solusi mutlak untuk menjamin aksesibilitas informasi di berbagai dimensi layar. Strategi ini memastikan website memiliki waktu muat yang efisien dan dapat diakses dengan nyaman melalui berbagai perangkat tanpa merusak tata letak konten.

3.3.1. Tampilan Desktop

Pada tampilan desktop dengan resolusi layar yang lebih lebar (di atas 1024 piksel), antarmuka memanfaatkan ruang yang tersedia untuk memaksimalkan penyajian informasi. Elemen navigasi utama ditampilkan secara penuh dalam format horizontal (flex-row) untuk memudahkan akses cepat ke menu utama. Bagian konten utama, seperti daftar video kajian, ditata menggunakan sistem Grid multi-kolom agar proporsi antar elemen terlihat seimbang dan memanfaatkan whitespace dengan baik.



3.3.2. Tampilan Mobile

Pada perangkat seluler, tata letak antarmuka beradaptasi secara otomatis untuk mengatasi keterbatasan lebar layar. Elemen navigasi yang kompleks diringkas menjadi tombol menu hamburger atau sidebar tersembunyi untuk menghemat ruang. Struktur konten yang sebelumnya berbentuk grid horizontal diubah menjadi satu kolom vertikal (stacking) menggunakan properti flex-col. Selain itu, ukuran tipografi dan dimensi tombol interaktif disesuaikan agar tetap terbaca dengan jelas dan mudah diakses menggunakan sentuhan jari (touch-friendly).



3.4. Integrasi Data Dinamis

Untuk memaksimalkan fungsi website sebagai platform informasi digital yang dinamis, tahap akhir pengembangan melibatkan integrasi antara antarmuka depan (front-end) dengan basis data melalui Advanced Custom Fields (ACF). Pada tahapan ini, variabel-variabel statis di dalam templat Blade digantikan dengan fungsi pemanggil data dinamis yang terhubung ke field khusus di panel admin. Integrasi ini memberikan kendali penuh kepada administrator untuk melakukan manajemen konten mulai dari teks deskripsi, unggahan gambar, hingga tautan multimedia secara real-time melalui dasbor WordPress. Pendekatan ini secara efektif memisahkan peran antara pengembang dan pengelola konten, sehingga pembaruan informasi dapat dilakukan dengan mudah dan aman tanpa risiko merusak struktur kode program yang mendasarinya. Dengan tampilan yang rapi dan konsisten, pengguna dapat fokus pada konten kajian tanpa gangguan visual. Aspek kenyamanan antarmuka (User Interface) dan pengalaman pengguna (User Experience) ini merupakan faktor kunci keberhasilan sebuah platform pembelajaran digital [10].



Gambar 3.5 Input ACF Video



Gambar 3.6 Hasil Input ACF Video

4. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa proses pengembangan antarmuka pada website Kajianmu telah berhasil menghasilkan sistem front-end yang modern, modular, dan memiliki mudah untuk dikembangkan lebih lanjut. Pendekatan pengembangan berbasis komponen terbukti efektif dalam mengatasi kendala struktur kode yang sering ditemui pada pengembangan tema WordPress tradisional.

Penggunaan Roots Sage 10 sebagai kerangka kerja utama, yang didukung oleh pustaka Roots Acorn, sangat membantu dalam menciptakan arsitektur tema yang rapi dengan memanfaatkan logika pemrograman dan tampilan. Pemanfaatan template engine Laravel Blade memungkinkan penulisan kode HTML yang lebih ringkas dan terstruktur, sehingga memudahkan proses pemeliharaan di masa mendatang.

Selain itu, penerapan Tailwind CSS dengan konsep utility-first tidak hanya mempercepat proses penataan gaya (styling), tetapi juga terbukti menghasilkan efisiensi kode yang lebih baik dibandingkan kerangka kerja CSS berbasis komponen [8]. Hasil pengujian menunjukkan bahwa desain responsif berjalan dengan konsisten, di mana tampilan website mampu beradaptasi secara presisi baik pada perangkat desktop maupun perangkat seluler (mobile).

Secara keseluruhan, implementasi stack teknologi modern ini telah memenuhi tujuan penelitian, yaitu menciptakan antarmuka website Kajianmu yang tidak

hanya menarik secara visual, tetapi juga rapi secara struktur kode dan siap digunakan sebagai platform informasi digital yang handal.

5. DAFTAR PUSTAKA

- [1] Fauzi, A., & Wibowo, S. (2022). Rancang Bangun Sistem Informasi Manajemen Proyek Menggunakan Framework Laravel dan Konsep Component-Based Development. *Jurnal Ilmiah Teknologi Informasi Asia*, 16(1), 45-54.
- [2] Hidayat, T., & Saputra, W. (2022). Perancangan Front-End Website E-Commerce Menggunakan Metode Utility-First CSS. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 9(3), 45-52.
- [3] Nugroho, A. S., & Yulianto, L. (2023). Pengembangan Tema WordPress Kustom Menggunakan Pendekatan Modular untuk Optimalisasi Kecepatan Website. *Jurnal Transformatika*, 20(2), 12-20.
- [4] Prasetyo, B., & Hartono, D. (2021). Implementasi Docker Container untuk Efisiensi Lingkungan Pengembangan Aplikasi Berbasis Web. *Jurnal Sistem Komputer dan Informatika (JSON)*, 3(2), 112-119.
- [5] Pratama, A. R., & Nugraha, F. S. (2023). Implementasi Framework Tailwind CSS pada Perancangan Antarmuka Website Sistem Informasi. *Jurnal Manajemen Informatika (JAMIKA)*, 13(1), 15-25.
- [6] Ramadhan, T., & Putra, G. (2023). Perancangan UI/UX Aplikasi Web Marketplace Menggunakan Pendekatan Design Thinking. *Jurnal Desain Komunikasi Visual dan Multimedia*, 7(1), 10-18.
- [7] Sari, I. P., & Ramadhani, F. (2021). Pemanfaatan Content Management System (CMS) WordPress untuk Pembangunan Website Profil Sekolah. *Jurnal Pengabdian Kepada Masyarakat*, 2(1), 34-40.
- [8] Setiawan, D., & Arifin, Z. (2024). Analisis Perbandingan Performa dan Efisiensi Kode Antara Framework Bootstrap dan Tailwind CSS. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 8(2), 110-118.
- [9] Susanto, R., & Andriana, A. D. (2020). Penerapan Metode Responsive Web Design pada Sistem Informasi Geografis Pariwisata. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 9(1), 55-62.
- [10] Wardani, D. K., & Purnomo, A. (2022). Analisis User Interface dan User Experience pada Website E-Learning Menggunakan Metode Heuristic Evaluation. *Jurnal Teknik Informatika dan Sistem Informasi*, 8(3), 445-456.
- [11] Wijaya, K., & Santoso, H. (2021). Penerapan Template Engine Blade pada Framework Laravel dalam Membangun Sistem Informasi Akademik. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(1), 23-30.
- [12] Purnomo, A., & Kusuma, W. A. (2023). Strategi Mobile-First Design dalam Meningkatkan User Experience pada Aplikasi Web Responsif. *Jurnal Teknologi dan Sistem Komputer*, 11(2), 88-95.
- [13] Firmansyah, R., & Haryanto, D. (2024). Analisis Optimasi Performa Website Menggunakan Minifikasi Aset CSS dan JavaScript. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 6(1), 30-38.
- [14] Rahmawati, E., & Santoso, B. (2022). Implementasi Arsitektur MVC (Model-View-Controller) pada Pengembangan Sistem Informasi Berbasis Web Modern. *Jurnal Ilmiah Komputasi*, 21(3), 215-224.
- [15] Kurniawan, D., & Lestari, S. (2023). Kustomisasi Manajemen Konten Website Menggunakan Plugin Advanced Custom Fields pada WordPress. *Jurnal Manajemen Informatika dan Sistem Informasi*, 9(1), 45-51.