

IMPLEMENTASI REACT JS DAN INTEGRASI STRAPI CMS DALAM PENGEMBANGAN WEBSITE OK OCE INDONESIA

Reza Hakim, Carudin, Yuyun Umaidah

Informatika, Universitas Singaperbangsa Karawang

Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361, Telp. (0267) 641177
2110631170100@student.unsika.ac.id

ABSTRAK

Perkembangan teknologi dan transformasi digital memberikan peluang besar bagi organisasi dan UMKM untuk meningkatkan efisiensi layanan melalui optimalisasi *platform* digital, seperti *website*. Penelitian ini bertujuan untuk mengembangkan *website* OK OCE Indonesia dengan menggunakan *framework* React JS pada *frontend* dan Strapi CMS pada *backend*, yang diintegrasikan untuk meningkatkan efisiensi manajemen konten, kualitas antarmuka pengguna (UI), serta pengalaman pengguna (UX). Penelitian ini menggunakan metode *Software Development Life Cycle* (SDLC) dengan model *Waterfall*, yang mencakup tahapan analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan. Pada tahap implementasi, React JS digunakan untuk membangun komponen antarmuka yang responsif, sementara Strapi CMS memungkinkan pengelolaan konten secara terstruktur dan efisien. Pengujian dilakukan dalam dua tahap, yaitu *alpha testing* dan *beta testing* dengan hasil pengujian menunjukkan bahwa tingkat kepuasan pengguna terhadap fungsionalitas *website* mencapai 89,5%, tampilan antarmuka 92,16%, dan kemudahan penggunaan 92,16%, yang dikategorikan sebagai “sangat memuaskan”. Hasil penelitian ini menunjukkan bahwa pengembangan *website* menggunakan React JS dan integrasi Strapi CMS mampu meningkatkan efisiensi manajemen konten dan memberikan pengalaman pengguna yang lebih baik. Implementasi ini diharapkan dapat menjadi referensi bagi organisasi lain dalam mengoptimalkan *platform* digital mereka.

Kata kunci: React JS, Strapi CMS, Website Development, SDLC Waterfall

1. PENDAHULUAN

Perkembangan teknologi di Indonesia dalam lima tahun terakhir menunjukkan kemajuan yang signifikan, mencakup berbagai sektor mulai dari telekomunikasi, industri digital, hingga kecerdasan buatan. Transformasi digital ini tidak hanya mempengaruhi cara masyarakat berinteraksi, tetapi juga berkontribusi terhadap pertumbuhan ekonomi nasional. Salah satu indikator utama dari perkembangan teknologi adalah jumlah pengguna internet. Menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), pada tahun 2024, jumlah pengguna internet di Indonesia mencapai lebih dari 221 juta orang, dengan tingkat penetrasi mencapai 79,5% dari total populasi. Hal ini menunjukkan peningkatan yang konsisten dibandingkan dari tahun-tahun sebelumnya [1].

Seiring dengan meningkatnya pengguna internet, kebutuhan akan *website* yang fungsional dan efektif menjadi semakin penting. Saat ini, masalah yang dihadapi oleh banyak organisasi dan perusahaan adalah bagaimana mengoptimalkan *website* mereka agar dapat memenuhi kebutuhan pengguna dengan baik, menyediakan informasi yang relevan dan mudah diakses.

Salah satu sektor yang terkena dampak signifikan dari masalah *website* ini adalah perdagangan daring, yang berperan penting dalam mendukung Usaha Mikro, Kecil, dan Menengah (UMKM). UMKM merupakan tulang punggung perekonomian Indonesia dan membutuhkan dukungan digital yang kuat agar dapat berkembang di era teknologi saat ini. Selain itu,

program-program pemerintah dan inisiatif swasta seperti OK OCE Indonesia turut berperan dalam mendorong digitalisasi UMKM. OK OCE (*One Kecamatan One Center for Entrepreneurship*) adalah gerakan yang bertujuan untuk menciptakan wirausaha baru dan memperkuat UMKM di seluruh Indonesia.

Tantangan yang dihadapi oleh OK OCE Indonesia sejalan dengan masalah yang dihadapi oleh banyak organisasi lain terkait optimalisasi *website*. Oleh karena itu, perlu adanya solusi yang dapat mengintegrasikan konten secara efisien, meningkatkan tampilan antarmuka, dan menyesuaikan fitur sesuai dengan kebutuhan bisnis.

Terdapat penelitian sebelumnya yang membahas topik yang mirip yaitu penelitian yang dilakukan oleh Hagi Azzam Azzikri dengan judul “Perancangan Ulang UI/UX *Website* OK OCE Indonesia Menggunakan Metode Lean UX” yang bertujuan untuk merancang dan membuat *prototype* desain antarmuka pengguna (UI) dan pengalaman pengguna (UX) *website* OK OCE Indonesia yang lebih baik dengan penerapan metode *Lean UX*.

Dengan mengacu pada masalah yang telah diuraikan sebelumnya, penelitian ini akan membahas pengembangan aplikasi dengan judul “Implementasi React JS dan Integrasi Strapi CMS Dalam Pengembangan *Website* OK OCE Indonesia” dan menggunakan metode *Software Development Life Cycle* (SDLC) *Waterfall*. Alasan menggunakan metode ini adalah karena sistem yang akan dikembangkan di OK OCE Indonesia memerlukan

proses pengembangan yang terdokumentasi berkelanjutan.

2. TINJAUAN PUSTAKA

2.1. OK OCE Indonesia

OK OCE (*One Kecamatan One Center for Entrepreneurship*) merupakan program kewirausahaan yang berbasis di tingkat kecamatan, dengan tujuan untuk menciptakan lapangan kerja baru di wilayah DKI Jakarta dan secara luas di Indonesia. Program ini didirikan oleh Sandiaga Uno, yang membawa visi ekonomi berfokus pada rakyat dengan prinsip keadilan, kesejahteraan, kualitas, dan kesadaran lingkungan. OK OCE bertujuan untuk mendorong lahirnya wirausaha-wirausaha baru melalui pembangunan pusat kewirausahaan di tingkat kabupaten, sehingga mampu memperkuat produk-produk UMKM. Program ini juga berfokus pada penciptaan sebanyak mungkin lapangan kerja dengan mengutamakan penggunaan tenaga kerja lokal [2].

2.2. Pengembangan Website

Pengembangan *website* merupakan proses pembuatan, perancangan dan pengelolaan situs web untuk memenuhi kebutuhan pengguna, baik individu, organisasi, maupun bisnis. Pengembangan *website* tidak hanya mencakup desain tampilan visual (*frontend*), tetapi juga aspek fungsional yang melibatkan penulisan kode dan manajemen data (*backend*) [3].

2.3. User Interface

User Interface (UI) merupakan tampilan sebuah produk yang berperan sebagai penghubung antara sistem dan pengguna. UI dapat berupa elemen seperti warna, bentuk, serta teks yang menarik yang ditampilkan dalam aplikasi. UI pada dasarnya adalah aspek visual dan interaktif dari suatu aplikasi atau *website* yang memudahkan pengguna berinteraksi dengan sistem [4].

2.4. Framework

Framework adalah kumpulan instruksi yang terorganisir dalam *class* dan fungsi-fungsi yang memiliki peran masing-masing, sehingga memudahkan pengembang untuk menggunakannya tanpa perlu menulis ulang sintaks program yang sama secara berulang, sekaligus menghemat waktu pengembangan [5].

2.5. React JS

React JS adalah *library* JavaScript *open-source* yang terutama digunakan untuk membangun *User Interface* (UI) secara khusus pada satu halaman dalam aplikasi. Pertama kali dikembangkan oleh Jordan Walke sekitar tahun 2013, React JS kini digunakan sebagai salah satu *framework* dalam pembuatan bagian *front-end* aplikasi. React JS berperan dalam mengelola lapisan tampilan untuk *desktop* maupun *mobile*. Dengan React JS, pengguna dapat membuat

komponen yang dapat digunakan kembali (*reusable components*) tanpa perlu membangunnya dari awal [6].

2.6. Strapi CMS

Strapi CMS memiliki antarmuka pengguna yang ramah dan mudah digunakan, alasan di balik kemudahannya adalah karena Strapi CMS mempermudah proses membuat, mengelola, dan mempublikasikan konten melalui UI sederhana. Tidak memerlukan pengetahuan teknis yang mendalam untuk mengoperasikan Strapi CMS, pengguna dapat dengan mudah menambahkan, mengedit, dan menghapus konten dengan beberapa klik dari antarmuka Strapi CMS. Strapi CMS juga menyediakan dokumentasi yang jelas dan terstruktur, sehingga memudahkan pengguna tim untuk menggunakannya [7].

2.7. Software Development Life Cycle

Software Development Life Cycle (SDLC) adalah proses perancangan sistem yang mencakup model dan metodologi yang digunakan untuk mengembangkan suatu sistem, dengan tujuan menghasilkan *output* berkualitas tinggi yang sesuai dengan ekspektasi pengguna [8].

2.8. Alpha Testing

Pengujian ini dilakukan untuk mengevaluasi kualitas dan kestabilan aplikasi yang sedang diuji. Prosesnya melibatkan pengecekan menyeluruh terhadap semua fungsi yang ada dalam sistem aplikasi tersebut. Fokus utama dari *alpha testing* adalah mendeteksi *bug* dengan pertanyaan dasar, "Apakah aplikasi ini berfungsi dengan benar?" [9].

2.9. Beta Testing

Beta Testing adalah tahap uji coba aplikasi atau perangkat lunak yang dilakukan setelah pengembangan produk mendekati tahap akhir, tetapi sebelum rilis resmi. Pada fase ini, produk diuji oleh sekelompok pengguna eksternal yang bukan bagian dari tim pengembangan untuk menemukan bug, kesalahan, atau masalah performa yang tidak terdeteksi selama pengujian internal.

2.10. Diagram UML

UML adalah bahasa yang digunakan untuk memvisualisasikan, membangun, dan mendokumentasikan artefak (bagian informasi yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak, seperti model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak, termasuk pemodelan bisnis dan sistem non-perangkat lunak lainnya. [10].

2.11. Penelitian Sebelumnya

Terdapat penelitian sebelumnya yang membahas topik yang mirip yaitu penelitian yang dilakukan oleh Hagi Azzam Azzikri dengan judul "Perancangan

Ulang UI/UX Website OK OCE Indonesia Menggunakan Metode Lean UX” yang bertujuan untuk merancang dan membuat *prototype* desain antarmuka pengguna (UI) dan pengalaman pengguna (UX) website OK OCE Indonesia yang lebih baik dengan penerapan metode *Lean UX* (Azzikri, 2024). Penelitian ini berbeda dengan penelitian sebelumnya karena penelitian ini berfokus pada pengembangan website yang lebih komprehensif untuk menjawab masalah manajemen konten dan optimalisasi fitur.

3. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah *Software Development Life Cycle* (SDLC) model *waterfall*. Model *waterfall* merupakan metode pengembangan perangkat lunak yang terstruktur dan linear, dimana setiap tahapan pengembangan dilakukan secara berurutan dan harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Dimulai dari analisis kebutuhan, desain sistem, implementasi, pengujian, hingga pemeliharaan. Pendekatan ini dipilih karena memberikan struktur yang jelas dan terdokumentasi dengan baik pada setiap tahap pengembangan, sehingga memastikan sistem yang dihasilkan sesuai dengan spesifikasi yang dibutuhkan.

3.1. Requirement

Pada tahap *requirement*, dilakukan pengumpulan dan analisis kebutuhan sistem secara menyeluruh. Kebutuhan ini mencakup fungsionalitas yang diharapkan, spesifikasi teknis, performa sistem, serta batasan yang ada. Pada tahapan ini, pengembang mengidentifikasi dan mendokumentasikan semua persyaratan fungsional dan non-fungsional yang diharapkan oleh pemangku kepentingan.

3.2. Design

Pada tahap ini, pengembang akan mendesain sistem berdasarkan kebutuhan yang telah dikumpulkan pada tahap sebelumnya. Pengembang akan memulai merancang sistem yang dimulai dari *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Selain itu, pengembang juga akan merancang konsep *visual* dan struktur situs yang dirancang untuk memastikan tampilan yang menarik dan navigasi yang mudah dimengerti.

3.3. Implementation

Pada tahap ini, pengembang mulai menerjemahkan desain yang telah dibuat ke dalam kode program. Fokus utama adalah pengembangan *frontend* menggunakan React JS yang mencakup pembuatan komponen *reusable* pada React JS, seperti navigasi, *form*, dan elemen-elemen visual lainnya, yang terintegrasi dengan Strapi CMS, dimana Strapi CMS berperan sebagai *backend* yang berfungsi untuk manajemen konten. Strapi CMS juga digunakan untuk membuat dan mengelola API yang fleksibel, memungkinkan pengembang memisahkan logika *backend* dari tampilan *frontend*.

3.4. Testing

Pada tahap ini, sistem yang telah dikembangkan akan diuji untuk memastikan bahwa semua fungsi berjalan dengan baik dan sesuai dengan kebutuhan yang telah didefinisikan pada tahap awal. Pengujian ini termasuk dalam kategori *User Acceptance Testing* (UAT), yang dilakukan dalam dua tahap yaitu *alpha testing* dan *beta testing*.

3.5. Maintenance

Tahapan *maintenance* dalam sistem ini mencakup beberapa langkah penting. Dimana *maintenance* dilibatkan dalam perbaikan *bug* untuk mengatasi masalah yang tidak terdeteksi selama pengujian awal. Selanjutnya, pembaruan fitur harus dilakukan sesuai dengan kebutuhan pengguna yang berkembang. Selain itu, peningkatan performa sistem diperlukan untuk menjaga efisiensi ketika data pengguna meningkat.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Hasil penelitian yang diperoleh pada penelitian ini yaitu sebuah aplikasi berbasis website menggunakan *framework* React JS dan terintegrasi dengan Strapi CMS yang dapat menyelesaikan permasalahan pada website sebelumnya dan diharapkan dapat meningkatkan kualitas pelayanan bagi pengguna.

4.2. Requirement

Pada tahap *requirement*, pengembang telah melakukan pengumpulan kebutuhan yang diperlukan untuk pengembangan aplikasi berbasis website ini. Proses ini meliputi analisis masalah dan analisis kebutuhan yang bertujuan untuk memastikan bahwa aplikasi dapat memenuhi tujuan utama pengembangan, yaitu meningkatkan kualitas pelayanan bagi pengguna.

4.3. Analisis Masalah

Berdasarkan hasil diskusi dengan Rahma Destira Nastiani selaku eksekutif divisi data dan riset OK OCE Indonesia dan dilakukan analisis permasalahan pada penelitian ini mengidentifikasi sejumlah kendala, yaitu:

- Keterbatasan untuk melakukan pengembangan fitur website menggunakan *platform* WordPress karena keterbatasan fleksibilitas dan kustomisasi yang lebih kompleks.
- Tampilan antarmuka pengguna (UI) pada website ini kurang optimal, ditandai dengan adanya elemen-elemen yang bertumpuk serta komponen yang ambigu dalam fungsi dan identitasnya.
- Kesulitan dalam pengelolaan manajemen website karena tidak adanya integrasi yang baik antar *platform* digital yang digunakan.

4.4. Analisis Masalah

Berdasarkan hasil analisis masalah tersebut, maka diperlukan beberapa hal untuk menyelesaikan permasalahan tersebut, di antaranya

4.5. Kebutuhan Fungsional

Kebutuhan fungsional adalah deskripsi spesifik tentang apa yang harus dilakukan sistem untuk memenuhi tujuan atau permintaan pengguna. Ini mencakup fungsi, fitur, dan layanan yang harus disediakan oleh sistem. Kebutuhan fungsional yang dibutuhkan pada pengembangan *website* ini tersedia pada tabel 1. berikut:

Tabel 1. Kebutuhan Fungsional

No.	Fitur	Deskripsi Fitur
1.	Strapi CMS OK OCE Indonesia	Admin dapat melakukan login ke dalam Strapi CMS dan mengelola data.
2.	Tentang Kami	Admin dan pengguna dapat melihat informasi tentang kami.
3.	7 TOP	Admin dan pengguna dapat melihat informasi 7 TOP OK OCE Indonesia.
4.	Etalase	Admin dan pengguna dapat melihat informasi etalase.
5.	Peluang	Admin dan pengguna dapat melihat informasi peluang.
6.	Event	Admin dan pengguna dapat melihat informasi <i>event</i> OK OCE Indonesia
7.	Informasi	Admin dan pengguna dapat melihat informasi singkat mengenai OK OCE Indonesia.

4.6. Kebutuhan Fungsional

Kebutuhan non-fungsional adalah spesifikasi yang mendefinisikan kualitas atau karakteristik dari sistem. Kebutuhan ini mencakup aspek-aspek yang bertujuan untuk memastikan bahwa sistem berjalan secara efisien sesuai dengan standar kualitas yang diharapkan. Pada pengembangan *website* ini, kebutuhan non-fungsional yang relevan telah dirangkum dalam Tabel 4.4 berikut ini:

Tabel 2. Kebutuhan Non-Fungsional

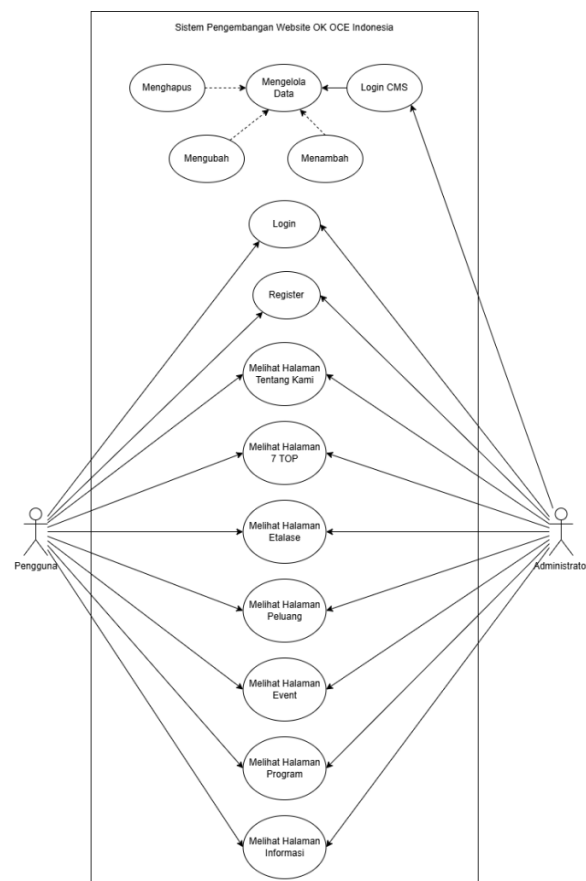
No.	Deskripsi Kebutuhan
1.	<i>Website</i> harus tersedia secara <i>online</i> setiap saat.
2.	<i>Website</i> harus terintegrasi dengan Strapi CMS melalui <i>Application Programming Interface</i> (API) dan dapat dikelola dengan baik oleh administrator.
3.	<i>Website</i> harus memiliki desain yang menarik dan nyaman saat digunakan oleh pengguna.
4.	Mampu dijalankan diberbagai jenis <i>web-browser</i> .
5.	<i>Website</i> harus dirancang untuk berkelanjutan dan dapat diperbarui secara berkala sesuai dengan perubahan kebutuhan OK OCE Indonesia.

4.7. Design

Pada tahap desain ini, dilakukan pembuatan diagram UML untuk memodelkan struktur dan alur sistem secara menyeluruh.

4.8. Use Case Diagram

Use case diagram memiliki peran untuk menggambarkan interaksi antara aktor dan sistem, Gambar 1. di bawah ini ialah gambaran *use case diagram* yang dirancang pada pengembangan *website* ini:

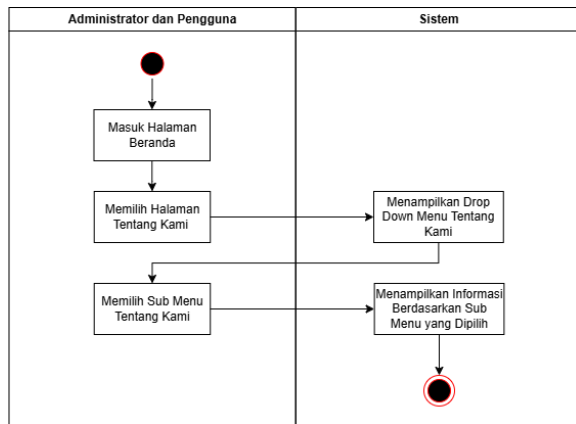


Gambar 1. Use Case Diagram

4.9. Activity Diagram

Dalam pengembangan *website* OK OCE Indonesia, *activity diagram* memiliki peran penting dalam menggambarkan alur kerja atau proses dalam sistem secara visual. Berikut ini ialah *activity diagram* yang akan dirancang:

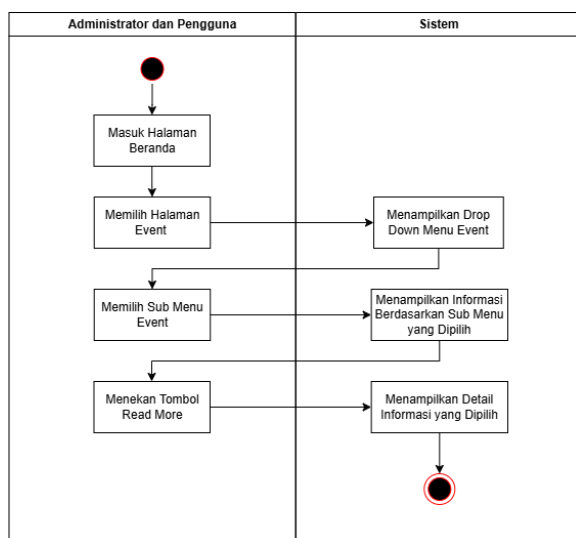
- Activity Diagram Melihat Halaman Tentang Kami**
Pada gambar 2. di bawah ini, diagram ini menggambarkan proses aktor mengakses informasi di halaman tentang kami. Dimulai dari aktor masuk ke halaman beranda, kemudian memilih menu tentang kami. Sistem menampilkan *dropdown* berisi pilihan sub-menu. Setelah aktor memilih salah satu sub-menu, sistem menampilkan informasi sesuai dengan sub-menu yang dipilih.



Gambar 2. Activity Diagram Melihat Halaman Tentang Kami

b. *Activity Diagram* Melihat Halaman *Event*

Pada gambar 3 di bawah ini, diagram ini menunjukkan alur aktor mengakses halaman *event* dari beranda, lalu aktor memilih *sub-menu* seperti daftar *event* atau berita, dan melihat informasi yang ditampilkan. Aktor dapat menekan tombol *read more* untuk melihat detail informasi.



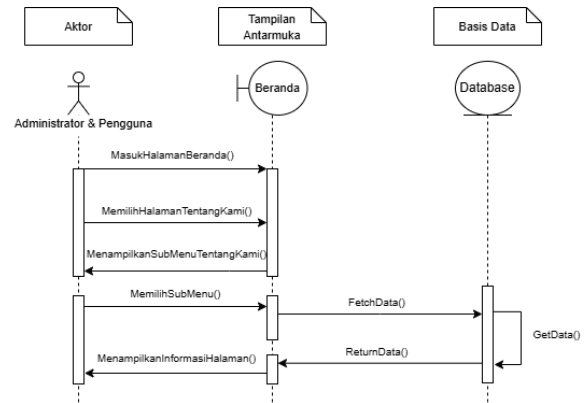
Gambar 3. *Activity Diagram* Melihat Halaman Event

4.10. Sequence Diagram

Sequence diagram adalah diagram yang menunjukkan alur pesan atau interaksi antar objek dalam sistem secara berurutan untuk memvisualisasikan proses secara rinci.

a. *Sequence Diagram* Melihat Halaman Tentang Kami

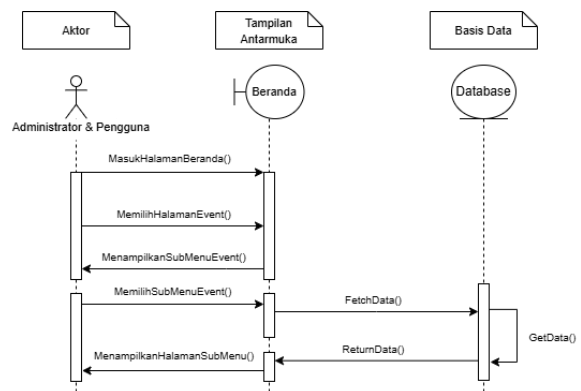
Pada gambar 4. berikut, menggambarkan alur aktor dalam mengakses halaman tentang kami. Aktor masuk ke halaman beranda, kemudian memilih menu tentang kami dan *sub-menu* yang diinginkan. Sistem akan mengambil data dari basis data, kemudian hasil informasi akan ditampilkan kepada aktor melalui antarmuka.



Gambar 4. *Sequence Diagram* Melihat Halaman Tentang Kami

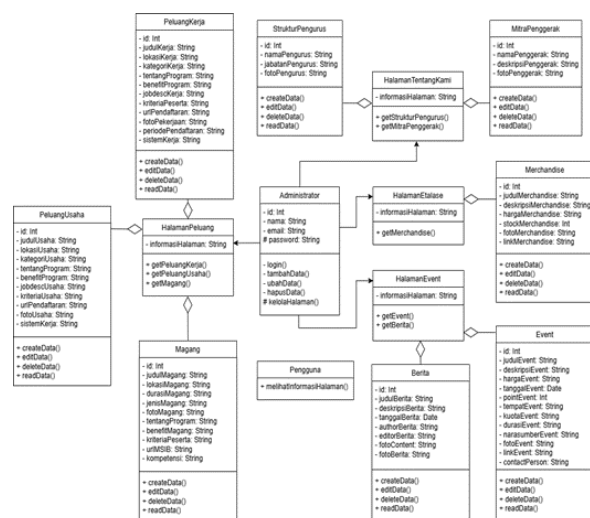
b. *Sequence Diagram* Melihat Halaman *Event*

Pada gambar 5 di atas, aktor masuk ke halaman beranda, kemudian memilih menu *event* dan *sub-menu event* yang diinginkan. Sistem akan mengambil data yang diperlukan dari basis data, kemudian halaman *event* yang dipilih akan ditampilkan kepada aktor.



Gambar 5. *Sequence Diagram* Melihat Halaman *Event*

4.11. Class Diagram



Gambar 6. *Class Diagram*

Class diagram merupakan bagian dari UML yang berfungsi untuk menggambarkan struktur dasar sistem dengan menampilkan kelas-kelas, atribut, metode, serta relasi antar kelas. Diagram ini memberikan representasi visual dari elemen-elemen inti sistem serta interaksi di antara objek-objeknya.

4.12. Implementation

Pada tahap implementasi ini, dilakukan realisasi dari rancangan sistem yang telah dirumuskan sebelumnya menjadi sebuah aplikasi berbasis *website*. Kegiatan ini mencakup pembangunan antarmuka menggunakan *framework* React JS dengan bahasa pemrograman JavaScript. Selain itu dalam proses pengembangan tampilan *frontend website*, digunakan *library* UI Flowbite berbasis Tailwind CSS agar proses pengembangan lebih efisien. Pada proses pengembangan *website ini* akan terintegrasi dengan *backend*, yaitu Strapi CMS agar pengelolaan *website* lebih mudah dikelola. Proses pengembangan pada tahap ini meliputi penulisan kode, pengaturan komponen, dan penyelarasan antara *frontend* dan *backend* agar *website* dapat berfungsi dengan maksimal.

a. Implementasi React JS (Frontend)

Pada tahap implementasi *frontend*, React JS digunakan untuk membangun antarmuka pengguna (UI) yang interaktif dan responsif untuk *website* OK OCE Indonesia. Implementasi ini dilakukan dengan mengikuti desain yang telah dirancang pada tahap desain sebelumnya.

b. Implementasi Strapi CMS (Backend)

Pada tahap ini, Strapi CMS digunakan sebagai *platform* manajemen konten untuk mengelola data yang akan ditampilkan di *frontend* (React JS). Strapi CMS adalah *headless content management system* yang memungkinkan pengelolaan konten secara fleksibel tanpa banyak memerlukan penulisan kode (minim *programming*). Dengan arsitektur berbasis API, Strapi CMS mempermudah pengembang untuk menghubungkan *backend* dengan *frontend* menggunakan REST API, sehingga *frontend* dan *backend* dapat bekerja secara independen.

c. Integrasi React JS dengan Strapi CMS

Pada tahap ini, dilakukan proses integrasi antara *frontend* (React JS) dan *backend* (Strapi CMS) menggunakan mekanisme REST API untuk menghubungkan keduanya. Integrasi ini bertujuan agar data yang dikelola melalui Strapi CMS dapat ditampilkan secara dinamis di *frontend* yang dibangun dengan React JS.

d. Implementasi User Interface

Pada tahap ini, *user interface* yang telah dirancang pada tahap sebelumnya, akan diimplementasikan dalam tahap ini, beberapa halaman diantaranya yaitu halaman beranda, struktur pengurus, mitra penggerak, *merchandise*, peluang usaha, peluang kerja, *magang*, *event* dan berita. Berikut ini *user*

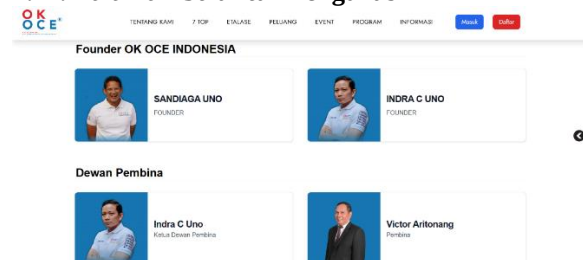
interface dalam pengembangan *website* OK OCE Indonesia yang terbagi menjadi dua antarmuka, yaitu tampilan antarmuka pengguna dan antarmuka Strapi CMS.

4.13. Halaman Beranda Website



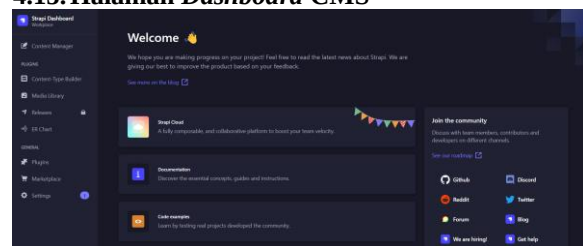
Gambar 7. Halaman Beranda

4.14. Halaman Struktur Pengurus



Gambar 8. Halaman Struktur Pengurus

4.15. Halaman Dashboard CMS



Gambar 9. Halaman Dashboard CMS

4.16. Testing

Pada tahap pengujian ini, dilakukan evaluasi terhadap sistem yang telah diimplementasikan untuk memastikan bahwa semua fitur dan fungsi berjalan sesuai dengan spesifikasi yang telah ditentukan. Dalam fase ini akan dilakukan dua tahap pengujian, yaitu *alpha testing* dan *beta testing*.

4.17. Alpha Testing

Alpha testing adalah pengujian yang dilakukan di lingkungan pengembang sebelum perangkat lunak dirilis ke pengguna. Tujuan pengujian ini adalah untuk mengidentifikasi kesalahan yang mungkin ada pada sistem, sehingga memastikan bahwa perangkat lunak telah berfungsi sesuai dengan spesifikasi yang ditentukan. Pada pengujian ini dilakukan menggunakan pendekatan *black box testing*.

Tabel 3. *Black Box Testing* React JS

No.	Skenario Pengujian	Hasil yang Diharapkan	Status
1.	Melihat Halaman Sejarah	Menampilkan halaman sejarah.	Lulus
2.	Melihat Halaman Visi dan Misi	Menampilkan halaman visi dan misi.	Lulus
3.	Melihat Halaman Nilai Inti	Menampilkan halaman nilai inti.	Lulus
4.	Melihat Halaman Struktur Pengurus	Menampilkan halaman struktur pengurus	Lulus
5.	Melihat Halaman Penggerak	Menampilkan halaman penggerak	Lulus
6.	Melihat Halaman <i>Anniversary</i>	Menampilkan halaman <i>anniversary</i>	Lulus

4.18. Beta Testing

Untuk mengukur kualitas *website* yang telah dikembangkan, dilakukan pengujian *beta testing* dengan menggunakan kuesioner yang disusun

berdasarkan tiga variabel utama, yaitu fungsionalitas *website*, tampilan antarmuka, dan kemudahan penggunaan. Kuesioner ini menggunakan skala Likert dengan rentan nilai dari skala 1 sampai 5.

Tabel 4. Indikator Variabel

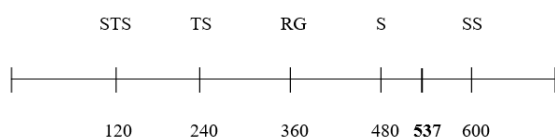
No.	Variabel	Kode	Indikator
1.	Fungsionalitas <i>Website</i>	FW1	Apakah informasi tentang program-program OK OCE ditampilkan dengan jelas di <i>website</i> ini?
		FW2	Apakah fitur navigasi menu antar halaman bekerja tanpa <i>error</i> ?
		FW3	Apakah semua tautan di <i>website</i> , termasuk tombol masuk dan daftar, berfungsi sesuai dengan tujuan masing-masing?
		FW4	Apakah gambar dan konten di setiap halaman ditampilkan dengan baik tanpa gangguan <i>loading</i> ?
2.	Tampilan Antarmuka	TA1	Apakah tampilan halaman utama <i>website</i> menarik dan sesuai dengan tema OK OCE?
		TA2	Apakah penggunaan warna dan ikon pada <i>website</i> memberikan kesan profesional dan ramah pengguna?
		TA3	Apakah teks pada <i>website</i> mudah dibaca dengan ukuran <i>font</i> yang sesuai?
		TA4	Apakah animasi atau transisi di <i>website</i> berjalan dengan baik dan tidak mengganggu pengalaman pengguna?
3.	Kemudahan Penggunaan	KP1	Apakah navigasi menu pada <i>website</i> mudah digunakan, bahkan oleh pengguna baru?
		KP2	Apakah pengguna dapat dengan mudah menemukan informasi penting seperti program, <i>event</i> , atau Pendaftaran?
		KP3	Apakah <i>website</i> responsif dan tetap terlihat baik saat diakses melalui ponsel?
		KP4	Apakah tombol-tombol seperti daftar, masuk, dan lainnya mudah ditemukan dan digunakan?

Selanjutnya dilakukan evaluasi tingkat kepuasan pengguna yang dilakukan berdasarkan data kuesioner yang diperoleh dari 30 responden. Data diolah dan dikonversikan ke dalam bentuk persentase untuk memudahkan interpretasi.

Tabel 5. Skala Penilaian Likert

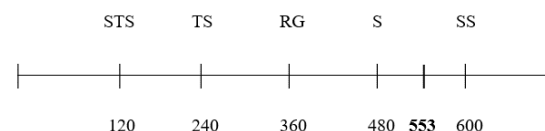
No.	Skala	Skor
1.	Sangat Setuju (SS)	5
2.	Setuju (S)	4
3.	Ragu – Ragu (RG)	3
4.	Tidak Setuju (TS)	2
5.	Sangat Tidak Setuju (STS)	1

a. Skala Kontinum Variabel Fungsionalitas *Website*

Gambar 10. Skala Kontinum Fungsionalitas *Website*

Berdasarkan data yang diperoleh dari 30 responden terhadap 4 variabel indikator dalam tampilan antarmuka, berada pada rata-rata 537 yang terletak pada daerah sangat setuju.

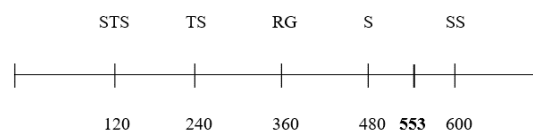
b. Skala Kontinum Variabel Tampilan Antarmuka



Gambar 11. Skala Kontinum Variabel Tampilan Antarmuka

Berdasarkan data yang diperoleh dari 30 responden terhadap 4 variabel indikator dalam tampilan antarmuka, berada pada rata-rata 553 yang terletak pada daerah sangat setuju.

c. Skala Kontinum Variabel Kemudahan Penggunaan



Gambar 12. Skala Kontinum Variabel Kemudahan Penggunaan

Berdasarkan data yang diperoleh dari 30 responden terhadap 4 variabel indikator dalam kemudahan penggunaan, berada pada rata-rata 553 yang terletak pada daerah setuju.

4.19. Maintenance

Tahap *maintenance* merupakan proses pemeliharaan sistem untuk memastikan performa dan stabilitas *website* tetap optimal. Tahap ini juga merupakan fase akhir dari pengembangan sistem ini. Dalam tahap ini, dilakukan perbaikan terhadap bug yang mungkin muncul setelah sistem digunakan, guna memastikan fungsi-fungsi utama berjalan sesuai kebutuhan. Selain itu, pengembangan lanjutan juga dilakukan dengan penyesuaian fitur-fitur berdasarkan masukan dari pengguna, serta penambahan fitur baru jika diperlukan untuk meningkatkan pengalaman pengguna.

4.20. Pembahasan

Penelitian ini berfokus pada pengembangan *website* OK OCE Indonesia berbasis React JS untuk *frontend* dan Strapi CMS untuk *backend*, yang diintegrasikan melalui REST API. Setiap tahapan dalam metode *waterfall*, mulai dari analisis kebutuhan hingga tahap *maintenance*, telah dilakukan dengan sistematis dan menghasilkan sistem yang sesuai dengan tujuan penelitian.

Pengembangan *website* ini dilakukan dengan tujuan untuk menciptakan *website* yang lebih responsif, efisien, dan mudah dikelola. Pada *website* sebelumnya, konten-konten yang diperlukan belum terintegrasi dengan baik dan tersebar di berbagai *platform* digital. Dengan dilakukan pengembangan *website* yang mengimplementasikan React JS dan terintegrasi Strapi CMS, setiap data untuk kebutuhan konten *website* tersimpan dalam satu sistem terpusat yang memudahkan pengelolaan dan memastikan konsistensi informasi yang disampaikan kepada pengguna.

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan yaitu implementasi React JS dan integrasi Strapi CMS dalam pengembangan *website* OK OCE Indonesia menggunakan metode *software development life cycle waterfall*, dapat ditarik kesimpulan bahwa penelitian ini berhasil mengembangkan ulang *website* OK OCE Indonesia dengan implementasi React JS sebagai *frontend*, untuk menciptakan antarmuka yang dinamis, modern dan responsif. Serta integrasi Strapi CMS sebagai *backend*, yang berfungsi untuk meningkatkan efisiensi pengelolaan konten, seperti menambah, mengubah, dan menghapus data program.

Hasil implementasi menunjukkan bahwa sistem baru mampu mengatasi keterbatasan fleksibilitas dan kustomisasi yang dialami pada *website* sebelumnya. Hal ini didukung dengan hasil pengujian *beta testing* melalui kuesioner yang diperoleh dengan hasil persentase skor sebesar 89,5% untuk fungsionalitas *website*, persentase skor sebesar 92,16% untuk tampilan antarmuka, dan persentase skor sebesar 92,16% untuk kemudahan penggunaan yang tergolong dalam kategori "Sangat Setuju".

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa rekomendasi yang dapat dijadikan bahan pertimbangan untuk pengembangan sistem di masa mendatang, seperti peningkatan desain dan responsivitas disarankan untuk terus diperbarui agar tetap menarik dan pemeliharaan sistem penting untuk dilakukan secara berkala, termasuk pembaruan keamanan, optimalisasi performa, dan penghapusan bug yang mungkin muncul.

DAFTAR PUSTAKA

- [1] M. Arif, "Pengguna Internet di Indonesia Meningkat di 2024," Asosiasi Penyelenggara Jasa Internet Indonesia (APJII). [Online]. Tersedia: <https://apjii.or.id/berita/d/pengguna-internet-di-indonesia-meningkat-di-2024>. [Diakses: 29 Sep. 2024].
- [2] P. Annisa, A. Azwar, dan E. Wartiningsih, "Analisis Implementasi Program OK OCE Dalam Meningkatkan Kesejahteraan Masyarakat Binaan (Studi Kasus Pada Warung Nasi Jamblang Ikhlas, Rawamangun, Jakarta Timur)," *Jurnal Administrasi Profesional*, vol. 1, no. 01, pp. 30–37, 2022. <https://doi.org/10.32722/jap.v1i01.5077>.
- [3] Y. Wahyudin and D. N. Rahayu, "Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literature Review," Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi, vol. 15, no. 3, pp. 26–40, 2020. <https://doi.org/10.35969/interkom.v15i3.74>.
- [4] W. Buana and B. N. Sari, "Analisis User Interface Meningkatkan Pengalaman Pengguna Menggunakan Usability Testing pada Aplikasi Android Course," *DoubleClick: Journal of Computer and Information Technology*, vol. 5, no. 2, p. 91, 2022. <https://doi.org/10.25273/doubleclick.v5i2.11669>.
- [5] A. F. Sallaby and I. Kanedi, "Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter," pp. 48–53, 2020.
- [6] T. Sulistyorini, E. Sova, R. Ramadhan, and U. Gunadarma, "Pemantauan Kasus Penyebaran COVID-19 Berbasis Website Menggunakan," Jurnal Informatika, vol. 1, no. 4, pp. 1–13, 2022.
- [7] A. Z. Ramadhan, "Apa itu Strapi: Kelebihan dan Kenapa Anda Harus Mencobanya," Lampung Developer Academy. [Online]. Tersedia: <https://lampungdev.com/apa-itu-strapi/>. [Diakses: 1 Okt. 2024].
- [8] C. Romindo, "Penerapan Model SDLC terhadap Sistem Penjualan Persediaan," Jurnal, vol. 7, no. 1, 2022.
- [9] P. S. Informatika, J. Studi Informasi, P. S. Arsitektur, Universitas Atma Jaya Yogyakarta, "Implementasi Pengujian Alpha dan Beta Testing pada Aplikasi Gamelan Virtual Reality," vol. 3, no. 1, pp. 48–60, 2023.
- [10] A. F. Prasetya, U. Lestari, dan D. Putri, "Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modeling Language)," Jurnal Car Rental Application Design, vol. 1, no. 1, pp. 14–18, 2022.