

Rancang Bangun Website Wisata Kabupaten Pasuruan dengan Pendekatan Component-Driven Development

Intan Nur Rahman¹, I Made Suartana²

^{1,2} Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

intannur.22004@mhs.unesa.ac.id

madesuartana@unesa.ac.id

Abstrak— Penelitian ini dilatarbelakangi oleh belum tersedianya media informasi digital yang terintegrasi dan terstruktur untuk menyajikan informasi destinasi wisata di Kabupaten Pasuruan. Informasi wisata yang tersedia masih tersebar di berbagai sumber dan belum didukung oleh antarmuka yang konsisten serta mudah digunakan, sehingga menyulitkan pengguna dalam memperoleh informasi secara efektif. Penelitian ini bertujuan untuk merancang dan membangun website wisata Kabupaten Pasuruan menggunakan pendekatan Component-Driven Development (CDD). Pendekatan CDD dipilih karena mampu menghasilkan antarmuka berbasis komponen yang modular, reusable, dan mudah dipelihara. Sistem dikembangkan menggunakan React JS sebagai frontend dan Laravel sebagai backend yang diintegrasikan melalui API. Pengujian sistem dilakukan melalui pengujian komponen dan pengujian pengguna. Pengujian komponen dilakukan menggunakan Storybook untuk pengujian visual dan React Testing Library (RTL) untuk pengujian fungsional. Sementara itu, pengujian pengguna dilakukan menggunakan metode System Usability Scale (SUS) dan Single Ease Question (SEQ). Hasil penelitian menunjukkan bahwa website berhasil menyajikan informasi wisata secara terstruktur, interaktif, dan mudah digunakan. Hasil pengujian SUS memperoleh nilai rata-rata sebesar 84,1 yang termasuk kategori acceptable, sedangkan hasil SEQ menunjukkan nilai rata-rata di atas 5,5 sehingga sistem dinilai mudah digunakan oleh pengguna. Selain itu, hasil evaluasi metrik kualitas komponen menunjukkan nilai Reusability Index (RI) sebesar 80%, Test Coverage Percentage (TCP) sebesar 100%, Component Cohesion (CpCoh) sebesar $\pm 90\%$, serta Execution Time Success Rate (ETSR) sebesar 100% yang menandakan bahwa komponen sistem memiliki tingkat penggunaan ulang, cakupan pengujian, koherensi, dan stabilitas yang sangat baik. Dengan demikian, penerapan Component-Driven Development terbukti mampu menghasilkan website wisata dengan antarmuka yang konsisten, modular, reusable, serta memiliki usability dan performa sistem yang baik.

Kata Kunci— Website Wisata, Component Driven-Development, React JS, Laravel, Usability, SUS, SEQ

I. PENDAHULUAN

Perkembangan teknologi informasi saat ini telah memberikan pengaruh besar terhadap berbagai sektor, termasuk sektor pariwisata. Pemanfaatan teknologi berbasis web menjadi salah satu solusi digital yang banyak digunakan untuk membantu Masyarakat memperoleh informasi secara cepat dan mudah diakses melalui internet. Kabupaten Pasuruan merupakan salah satu daerah di Jawa Timur yang memiliki berbagai destinasi wisata, seperti Gunung Bromo, Taman Safari Prigen, Kebun Raya Purwodadi, dan wisata alam lainnya. Namun, informasi mengenai destinasi wisata tersebut masih

belum tersaji secara menyeluruh dan terstruktur dalam media digital sehingga menyulitkan wisatawan dalam memperoleh informasi secara efektif.

Selain permasalahan penyediaan informasi, pengembangan website wisata juga sering menghadapi kendala pada aspek antarmuka pengguna (User Interface) dan pengalaman pengguna (User experience). Banyak website wisata dibangun dengan tampilan yang tidak konsisten, tata letak yang kurang rapi, serta navigasi yang kurang intuitif sehingga menyulitkan pengguna dalam memahami alur penggunaan sistem. Permasalahan tersebut berdampak pada menurunnya efektivitas media promosi digital serta rendahnya kenyamanan pengguna saat mengakses website wisata. Oleh karena itu, dibutuhkan suatu pendekatan pengembangan antarmuka yang tidak hanya mampu menyajikan informasi secara lengkap, tetapi juga menghasilkan tampilan yang konsisten, mudah digunakan, dan mudah dipelihara dalam jangka panjang.

Permasalahan tersebut diperkuat oleh beberapa penelitian sebelumnya yang menunjukkan bahwa sebagian besar website wisata masih berfokus pada fungsi dasar sistem tanpa memperhatikan efisiensi desain antarmuka dan modularitas sistem. Penelitian oleh Mirawati dan Qoiriah merupakan metode User Centered Design (UCD) dalam pengembangan website wisata Kabupaten Tuban yang berfokus pada keterlibatan pengguna, sedangkan penelitian Beauty & Suartana lebih menekankan pada optimasi Search Engine Optimization (SEO) untuk meningkatkan visibilitas website promosi wisata [1][2]. Namun, penelitian tersebut belum membahas pengembangan antarmuka berbasis komponen yang bersifat reusable dan mudah dikembangkan kembali. Pendekatan konvensional yang tidak modular menyebabkan proses pengembangan dan pemeliharaan sistem menjadi kurang efisien karena setiap perubahan tampilan atau penambahan fitur membutuhkan penyesuaian pada banyak bagian sistem.

Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan pendekatan Component-Driven Development (CDD) dalam pengembangan website wisata Kabupaten Pasuruan. CDD merupakan metode pengembangan perangkat lunak yang membangun sistem menggunakan komponen-komponen independen yang dapat digunakan kembali (reusable) [3]. Pendekatan ini dipilih karena mampu mendukung konsistensi antarmuka, meningkatkan efisiensi pengembangan, serta mempermudah proses pemeliharaan sistem. Website dikembangkan menggunakan React JS pada sisi frontend dan Laravel pada sisi Backend yang diintegrasikan melalui API [4]. Selain itu, pengujian komponen dilakukan

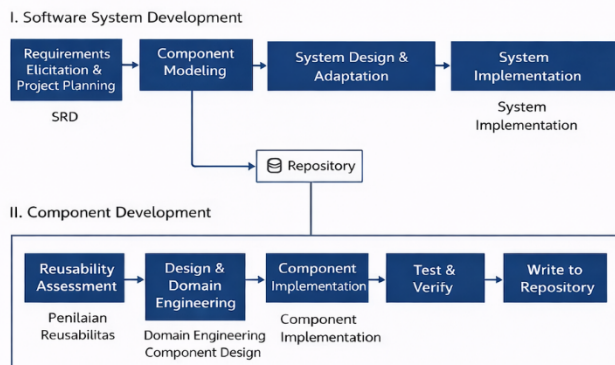
menggunakan Storybook untuk pengujian visual dan React Testing Library untuk pengujian fungsional guna memastikan setiap komponen dapat berjalan dengan baik dan konsisten pada berbagai kondisi[5]. Pengujian pengguna juga dilakukan menggunakan metode System Usability Scale (SUS) dan Single Ease Question (SEQ) untuk mengukur tingkat usability dan kemudahan penggunaan sistem.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menerapkan pendekatan Component Driven-Development (CDD) dalam pengembangan website wisata Kabupaten Pasuruan guna menghasilkan antarmuka yang modular, konsisten dan mudah digunakan[6]. Penelitian ini juga bertujuan mengintegrasikan komponen antarmuka React JS dengan backend Laravel melalui API serta melakukan pengujian terhadap komponen dan sistem untuk memastikan fungsionalitas dan keterpaduan sistem berjalan dengan baik[7]. Penelitian dibatasi pada pengembangan website wisata yang menampilkan 10 destinasi wisata di Kabupaten Pasuruan dengan fokus pada pengembangan antarmuka berbasis komponen dan pengujian fungsionalitas komponen menggunakan Storybook dan React Testing Library. Dengan demikian, hasil penelitian diharapkan mampu menjadi solusi digital yang informatif sekaligus menjadi implementasi nyata pendekatan CDD dalam pengembangan antarmuka website wisata modern[8].

II. METODE PENELITIAN

Penelitian ini melalui beberapa tahapan terstruktur yang ditunjukkan pada diagram alur penelitian di Gbr. 1

Tahapan Metode Component-Driven Development (CDD)



Gbr. 1 Alur Tahapan Metode Penelitian

A. Requirement Elicitation & Project Planning

Tahap ini merupakan tahap awal dalam perumusan kebutuhan pengguna dan kebutuhan sistem yang menjadi dasar perancangan antarmuka dengan pendekatan Component Driven-Development (CDD). Pada penelitian ini, Langkah yang dilakukan adalah menganalisis user requirement, functional requirement dan design requirement kemudian seluruh analisis kemudian dirangkum ke dalam dokumen Software Requirement Description (SRD) yang terstruktur.

B. Component Modeling

Proses pemodelan dilakukan dengan mengidentifikasi komponen antarmuka seperti navigasi, daftar destinasi, detail destinasi, pencarian, serta pemesanan tiket. Setiap komponen didefinisikan berdasarkan perannya dalam mendukung alur interaksi pengguna. Selain itu, hubungan antar komponen juga telah ditentukan untuk memastikan alur sistem berjalan secara logis dan konsisten.

1) Reusability Assessment

Prinsip utama dari Component-Driven Development (CDD) adalah memaksimalkan penggunaan Kembali (reuse) komponen yang telah ada agar proses pengembangan menjadi lebih efisien[9]. Langkah pertama yang dilakukan pada tahap ini yaitu meninjau dan melakukan pencarian komponen pada Pustaka eksternal yaitu Tailwind Components, selanjutnya memilih component yang paling sesuai untuk diintegrasikan ke dalam system. Namun jika tidak ada komponen yang sesuai, maka peneliti akan membuat komponen baru dengan mengikuti standar desain yang telah ditetapkan.

2) Design & Domain Engineering

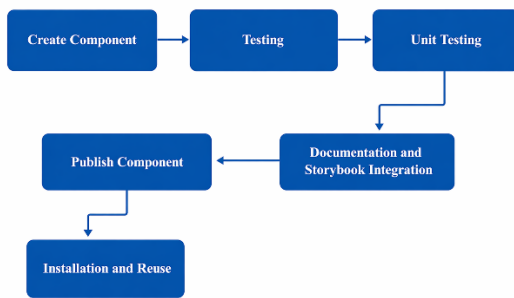
Setiap komponen dirancang dengan mempertimbangkan konteks domain pariwisata, seperti kebutuhan penyajian informasi destinasi, pemesanan tiket, serta interaksi pengguna dengan peta dan galeri visual. Pada tahap ini, peneliti menetapkan detail visual komponen, meliputi warna, tipografi, ikon, ukuran, dan jarak antar elemen, agar selaras dengan *design system* yang telah ditentukan pada tahap perancangan sistem sebelumnya. Selain aspek visual, perilaku interaksi komponen juga didefinisikan secara rinci, seperti kondisi *hover*, *active*, *disabled*, validasi input, serta respons komponen terhadap berbagai ukuran layar.

3) Component Implementation

Tahap Component Implementation merupakan proses penerapan hasil perancangan komponen ke dalam bentuk kode program secara teknis menggunakan framework React.js. Pada tahap ini, setiap komponen yang telah dirancang pada tahap Design & Domain Engineering diimplementasikan secara terpisah dengan mengacu pada spesifikasi visual, perilaku interaksi, serta aturan penggunaan yang telah ditetapkan sebelumnya. Pendekatan ini bertujuan untuk memastikan bahwa setiap komponen bersifat modular, mandiri, dan dapat digunakan kembali (reusable) sesuai dengan prinsip Component-Driven Development (CDD).

4) Test & Verify

Pada tahap ini, pengujian dan verifikasi dilakukan secara menyeluruh untuk memastikan bahwa setiap komponen berfungsi sesuai kebutuhan sebelum digabungkan ke dalam sistem yang lebih besar[10].



Gbr. 2 Alur Pengujian Komponen

Dengan melakukan pengujian seperti di Gbr. 2 dimulai dari unit testing menggunakan React Testing Library, di mana setiap komponen diuji secara terisolasi berdasarkan perilaku pengguna (behavior-driven testing), lalu dilakukan pula pengujian visual dan interaktif menggunakan Storybook, Serta dilengkapi dengan performance testing menggunakan Google Lighthouse dan juga pengukuran kuantitatif menggunakan beberapa metrik, yaitu:

1. **Reusability Index (RI)**

Metrik ini mengukur sejauh mana komponen dapat digunakan kembali pada konteks atau proyek lain. Semakin tinggi nilai RI, semakin efisien pengembangan karena komponen tidak perlu dibuat ulang.

$$RI = \frac{\text{Jumlah komponen yang dapat digunakan kembali}}{\text{Total komponen}} \times 100\%$$

2. **Test Coverage Percentage (TCP)**

Menghitung persentase baris kode atau fungsi yang tercakup dalam pengujian otomatis.

$$TCP = \frac{\text{Jumlah Komponen yang diuji}}{\text{Total Komponen}} \times 100\%$$

TCP tinggi (misalnya $\geq 80\%$) menunjukkan bahwa komponen diuji secara menyeluruh, sehingga kecil kemungkinan terdapat bug tersembunyi.

3. **Component Cohesion (CpCoh)**

Digunakan untuk menilai keterhubungan fungsi dalam satu komponen.

$$CpCoh = \frac{\text{Interaksi yang saling terkait}}{\text{Total interaksi}} \times 100\%$$

Nilai CpCoh yang tinggi menunjukkan bahwa fungsi-fungsi dalam komponen fokus pada satu tujuan tertentu, sehingga desain lebih modular, bersih, dan mudah dipelihara.

4. **Execution Time Success Rate (ETSR)**

Mengukur seberapa sering komponen dapat menjalankan fungsinya dengan sukses dalam batas waktu tertentu.

$$ETSR = \frac{\text{Jumlah eksekusi sukses tepat waktu}}{\text{Total eksekusi}} \times 100\%$$

Nilai ETSR tinggi menunjukkan bahwa komponen tidak hanya benar secara logika, tetapi juga efisien dalam performa.

5) **Write to Repository**

Pada tahap ini, seluruh komponen antarmuka yang telah diimplementasikan menggunakan React.js disimpan secara terorganisir berdasarkan fungsi dan tingkatannya. Penyimpanan ini bertujuan untuk memudahkan proses penelusuran, pemeliharaan, serta penggunaan ulang komponen pada bagian sistem yang lain. Selain kode sumber, repository juga memuat penjelasan singkat mengenai fungsi dan cara penggunaan komponen agar dapat dipahami dengan mudah pada tahap implementasi sistem berikutnya.

C. **System Design & Adaptation**

Tahap System Design & Adaptation merupakan proses perancangan visual dan struktur antarmuka sistem yang mengacu pada hasil pemodelan konseptual komponen pada tahap sebelumnya. Pada tahap ini, konsep komponen yang telah didefinisikan mulai diterjemahkan ke dalam bentuk rancangan



antarmuka yang lebih konkret dan terstruktur.

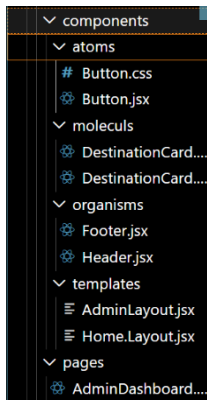
Gbr. 3 Tahap Atomic Design

Pendekatan **Atomic Design** pada Gbr.3 digunakan sebagai kerangka utama dalam menyusun komponen antarmuka secara hierarkis, mulai dari komponen terkecil hingga membentuk halaman sistem yang utuh. Pendekatan ini membagi komponen antarmuka ke dalam lima tingkatan, yaitu **atom**, **molekul**, **organisme**, **template**, dan **page**.

D. **System Implementation**

Tahap **System Implementation** merupakan proses penerapan seluruh hasil perancangan sistem ke dalam bentuk aplikasi yang dapat dijalankan. Proses implementasi diawali dengan penyusunan struktur proyek React yang terorganisasi berdasarkan prinsip *component-based architecture*. Struktur folder disusun untuk memisahkan komponen berdasarkan tingkat Atomic Design, seperti atom, molekul, organisme, template, dan halaman (*page*). Selanjutnya, peneliti mulai membangun struktur halaman dan navigasi utama sistem. Halaman-halaman utama yang dikembangkan meliputi halaman beranda, halaman daftar destinasi wisata, halaman

detail destinasi, serta halaman pemesanan tiket. Pada tahap awal implementasi, integrasi data dilakukan menggunakan *data dummy* sebagai pengganti data asli dari backend. Setelah struktur antarmuka dinyatakan stabil, proses implementasi



dilanjutkan dengan integrasi awal antara frontend React.js dan backend Laravel melalui mekanisme RESTful API. Hasil dari tahap ini adalah sebuah aplikasi website wisata berbasis React.js yang telah terstruktur dengan baik, siap untuk memasuki tahap pengujian lanjutan serta integrasi penuh dengan backend Laravel.

E. User Testing

Metode pengujian yang digunakan dalam penelitian ini mengacu pada pendekatan **Usability Testing** dengan dua instrumen utama, yaitu **System Usability Scale (SUS)** dan **Single Ease Question (SEQ)**. Pemilihan kedua instrumen ini didasarkan pada penelitian oleh (Mirnawati & Qoiriah, 2024b), di mana kombinasi antara metode SUS dan SEQ dinilai efektif untuk menilai persepsi pengguna terhadap tingkat kemudahan, kenyamanan, dan kualitas sistem secara menyeluruh.

Pengujian dilakukan setelah proses pengembangan prototipe website selesai. Responden diminta untuk mencoba berbagai fitur yang tersedia serta menyelesaikan beberapa skenario tugas yang telah ditentukan, seperti mencari destinasi wisata, melihat detail informasi lokasi, hingga melakukan simulasi pemesanan tiket. Setelah menyelesaikan tugas-tugas tersebut, responden kemudian mengisi kuesioner SUS dan SEQ untuk menilai pengalaman mereka selama menggunakan sistem.

III. HASIL DAN PEMBAHASAN

A. Hasil Implementasi Website

Website wisata Kabupaten Pasuruan berhasil di kembangkan menggunakan React JS sebagai frontend dan Laravel sebagai backend yang diintegrasikan melalui REST API. Implementasi ini dilakukan dengan menerapkan konsep Atomic Design sehingga

antarmuka website dibangun menggunakan komponen-komponen modular dan reusable.

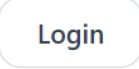
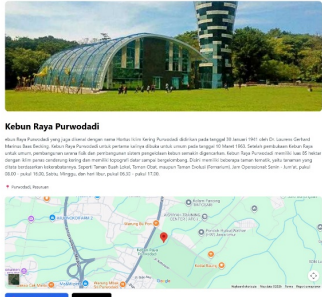
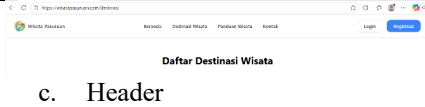
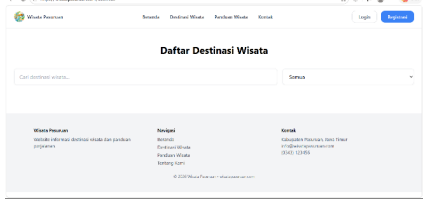
Pada tahap implementasi, komponen antarmuka berhasil dikembangkan ke dalam beberapa kategori, yaitu *atoms*, *molecules*, *organisms*, *templates*, dan *pages* seperti pada Gbr.4 berikut.

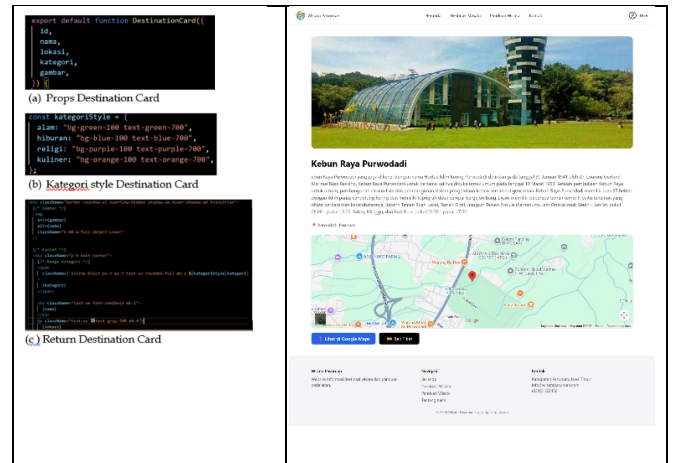
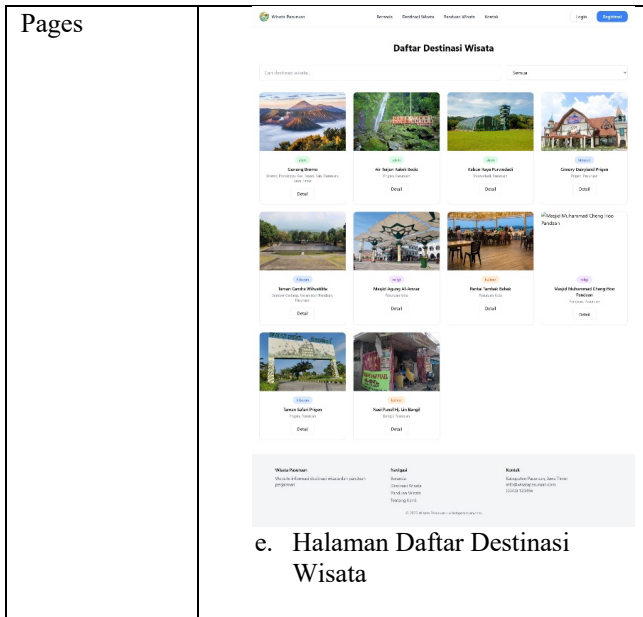
Gbr. 4 Struktur Komponen pada Proyek React.js

Komponen Button digunakan sebagai elemen dasar antarmuka untuk berbagai aksi pengguna seperti login, registrasi, pencarian wisata, dan pembelian tiket. Selanjutnya, komponen DestinationCard digunakan untuk menampilkan informasi singkat mengenai destinasi wisata berupa gambar, nama wisata, lokasi, dan kategori wisata. Komponen Header dan Footer digunakan sebagai bagian navigasi utama website yang diterapkan secara konsisten pada berbagai halaman sistem.

TABEL I

TABEL PENERAPAN ATOMIC DESIGN PADA KOMPONEN ANTARMUKA

Kategori	Hasil Antarmuka
Atoms	 <p>a. button</p>
Molecules	 <p>b. Destination Card</p>
Organisms	 <p>c. Header</p>
Templates	 <p>d. Home Layout</p>



Setelah proses implementasi komponen pada Tabel II selesai dilakukan, tahap berikutnya adalah Test & Verify untuk memastikan bahwa setiap komponen dapat berjalan sesuai fungsi dan tetap konsisten secara visual. Tahap *Test & Verify* pada penelitian ini telah dilakukan untuk memastikan bahwa komponen yang dikembangkan memiliki kualitas yang baik dari segi fungsionalitas, konsistensi tampilan, serta performa sistem. Pengujian dilakukan menggunakan beberapa pendekatan, yaitu *unit testing*, *visual testing*, dan *performance testing*.

Selanjutnya komponen – komponen yang telah dirancang pada Tabel I diimplementasikan ke dalam bentuk kode program menggunakan React.js. Pengembangan komponen dilakukan dengan memanfaatkan konsep *props* dan *state* pada React.js. Props digunakan untuk mengatur data yang bersifat dinamis, sehingga satu komponen dapat digunakan dalam berbagai konteks dengan tampilan atau fungsi yang berbeda.

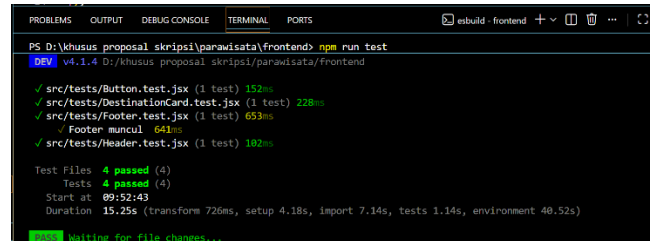
Sementara itu, *state* digunakan untuk mengelola kondisi internal komponen, seperti perubahan input pengguna, status interaksi, serta respon terhadap aksi tertentu.

Pengujian fungsional dilakukan menggunakan **React Testing Library** dengan bantuan **Vitest** sebagai *test runner*. Pada tahap ini, setiap komponen diuji secara terisolasi berdasarkan perilaku pengguna (*behavior-driven testing*). Komponen seperti Button, Header, Footer, dan DestinationCard diuji untuk memastikan bahwa elemen ditampilkan dengan benar dan mampu merespons interaksi pengguna sesuai dengan yang diharapkan.

TABEL II

TABEL IMPLEMENTASI KOMPONEN REACT.JS PADA ANTARMUKA SISTEM

Kode komponen	Antarmuka sistem
<pre>const Button = ({ children, color = "blue", outline = false, plate = false, disabled = false, href, onClick, }) => { const base = ... } (a) Props</pre>	
<pre>const colors = { blue: "bg-blue-500 text-white hover:bg-blue-600", cyan: "bg-cyan-500 text-white hover:bg-cyan-600", green: "bg-green-500 text-white hover:bg-green-600", red: "bg-red-500 text-white hover:bg-red-600", } (b) Styling</pre>	
<pre>return (<button onClick={onClick} disabled={disabled} className={className} {children} {button} />) (c) JSX</pre>	



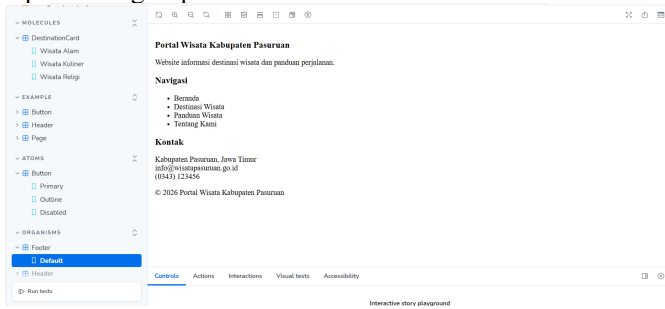
Gbr. 5 Hasil Pengujian Unit Testing Menggunakan Vitest dan React Testing Library

Pada Gbr.5 Seluruh komponen berhasil diuji dengan status passed, yang menunjukkan bahwa fungsi komponen berjalan sesuai dengan yang diharapkan

Selain pengujian fungsional, dilakukan pula pengujian visual menggunakan **Storybook**. Pada tahap ini, komponen ditampilkan secara terisolasi dalam berbagai kondisi, seperti normal, hover, dan aktif. Pengujian ini bertujuan untuk memastikan bahwa tampilan komponen konsisten dengan design system serta responsif terhadap berbagai ukuran layar.

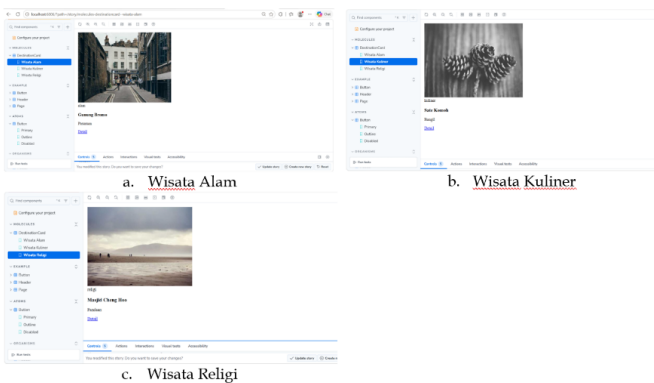
Gbr.6 Pengujian Visual Komponen Header Menggunakan Storybook

Berdasarkan Gbr.6, komponen Header diuji secara visual menggunakan Storybook untuk memastikan konsistensi tampilan navigasi pada sistem.



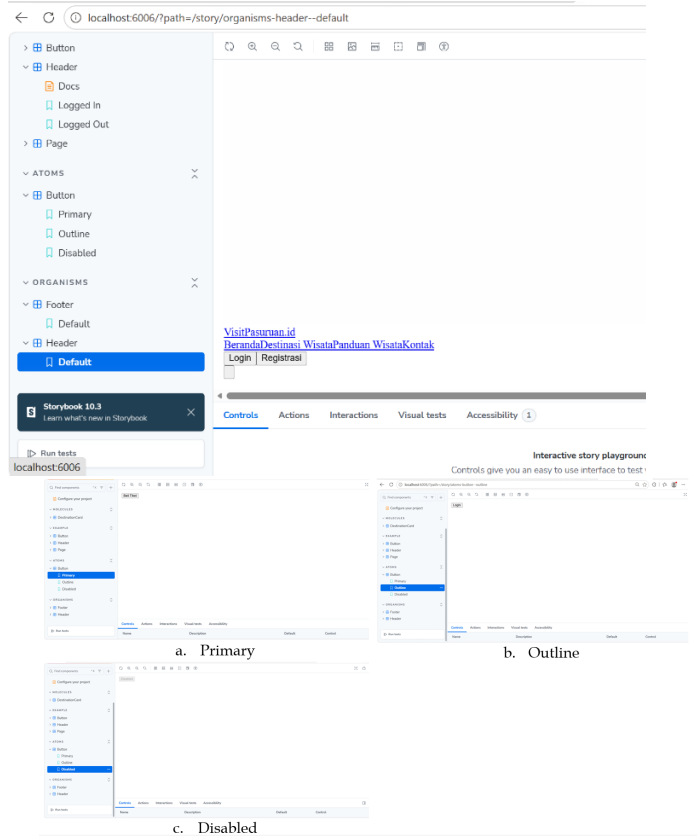
Gbr.7 Pengujian Visual Komponen Footer Menggunakan Storybook

Berdasarkan Gbr.7, komponen Footer diuji secara visual menggunakan Storybook untuk memastikan bahwa tampilan bagian bawah website tetap konsisten pada berbagai halaman sistem.



Gbr.8 Pengujian Visual Komponen Destination Card Menggunakan Storybook

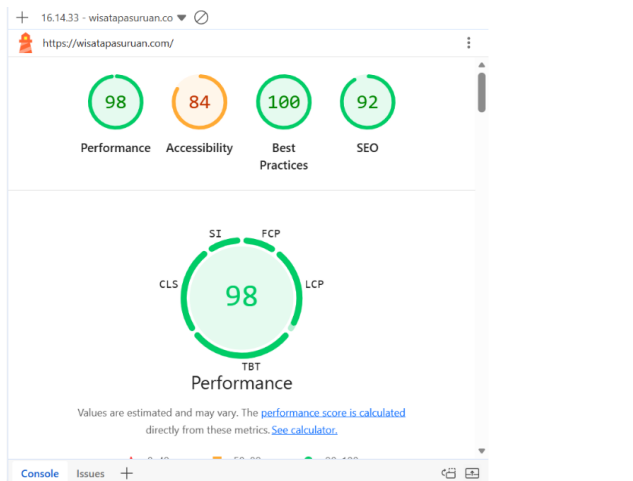
Berdasarkan Gbr.8, komponen DestinationCard diuji menggunakan beberapa variasi data destinasi wisata untuk memastikan bahwa komponen dapat digunakan kembali secara konsisten.



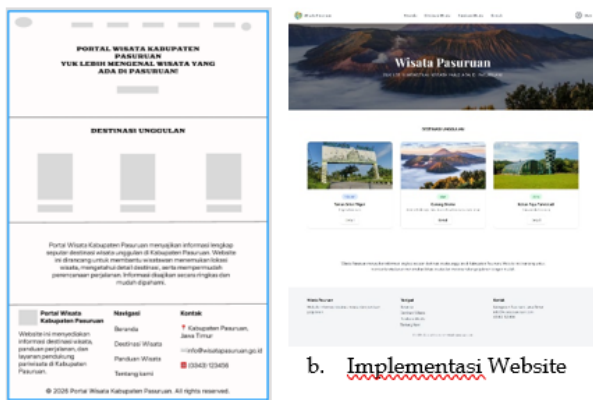
Gbr.9 Pengujian Visual Komponen Button Menggunakan Storybook

Berdasarkan Gbr.9, komponen Button diuji menggunakan beberapa variasi tampilan seperti Primary, Outline, dan Disabled untuk memastikan konsistensi visual komponen pada berbagai kondisi interaksi pengguna. Berdasarkan hasil pengujian visual menggunakan Storybook, seluruh komponen berhasil mempertahankan konsistensi tampilan pada berbagai kondisi dan variasi data.

Untuk mengevaluasi performa sistem, dilakukan pengujian menggunakan **Google Lighthouse**. Pengujian ini menilai aspek performa, aksesibilitas, best practices, dan SEO dari website yang dikembangkan.



Gbr.10 Hasil Pengujian Performa Menggunakan Google Lighthouse



a. Wireframe

b. Implementasi Website

Berdasarkan hasil pengujian pada Gbr.10 menunjukkan bahwa website memiliki performa yang baik dan memenuhi standar pengembangan web modern.

Selain itu, untuk memberikan evaluasi yang lebih objektif terhadap kualitas komponen, tahap verifikasi dilengkapi dengan pengukuran kuantitatif menggunakan beberapa metrik.

Berdasarkan hasil perhitungan, diperoleh nilai sebagai berikut:

TABEL III

TABEL HASIL PENGUKURAN METRIK KUALITAS KOMPONEN

Metrik	Nilai
Reusability Index (RI)	80%
Test Coverage Percentage (TCP)	100%
Component Cohesion (CpCoh)	±90%
Execution Time Success Rate (ETSR)	100%

Hasil pada Tabel III menunjukkan bahwa komponen yang dikembangkan memiliki tingkat reusability yang tinggi, telah diuji secara menyeluruh, memiliki struktur yang modular, serta

mampu berjalan dengan baik tanpa kegagalan selama proses pengujian.

Komponen yang telah berhasil melewati tahap pengujian kemudian disimpan dan dikelola dalam repository proyek untuk mendukung penggunaan kembali dan pemeliharaan sistem. Tahap *Write to Repository* pada penelitian ini telah dilakukan sebagai proses penyimpanan dan pengelolaan komponen yang telah dikembangkan dan diuji. Repository yang dimaksud dalam penelitian ini tidak terbatas pada platform publik seperti GitHub, tetapi mengacu pada penyimpanan kode secara terstruktur dalam lingkungan pengembangan proyek.

Dengan demikian, tahap *Write to Repository* memastikan bahwa seluruh komponen tersimpan secara sistematis, mudah diakses, serta siap digunakan kembali. Hal ini mendukung konsistensi antarmuka, efisiensi pengembangan, serta keberlanjutan sistem sesuai dengan pendekatan Component-Driven Development (CDD).

Setelah seluruh komponen berhasil dikembangkan, diuji, dan disimpan pada repository proyek pada tahap sebelumnya, proses selanjutnya adalah System Design & Adaptation. Tahap ini berfokus pada penerapan komponen-komponen yang telah dikembangkan ke dalam struktur halaman sistem secara utuh. Tahap *System Design & Adaptation* pada bab ini merupakan implementasi dari proses perancangan antarmuka yang telah dilakukan pada bab sebelumnya. Pada tahap ini, desain yang telah dibuat menggunakan Figma dalam bentuk *wireframe* dan *mockup* diimplementasikan ke dalam bentuk aplikasi berbasis web menggunakan React.js.

Gbr. 11 Implementasi Wireframe

Berdasarkan Gbr.11 hasil implementasi, wireframe membantu dalam menentukan tata letak elemen seperti header, konten utama, dan navigasi sehingga struktur halaman menjadi lebih terarah. Selanjutnya wireframe yang ada dijadikan mockup untuk menentukan dengan elemen visual seperti warna, tipografi, dan ikon.

TABEL IV

TABEL IMPLEMENTASI MOCKUP

Implementasi Mockup

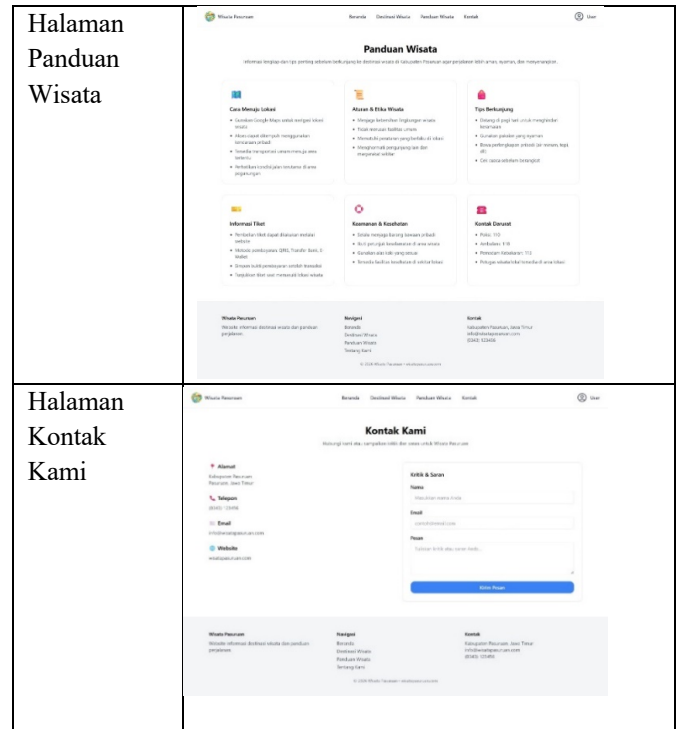
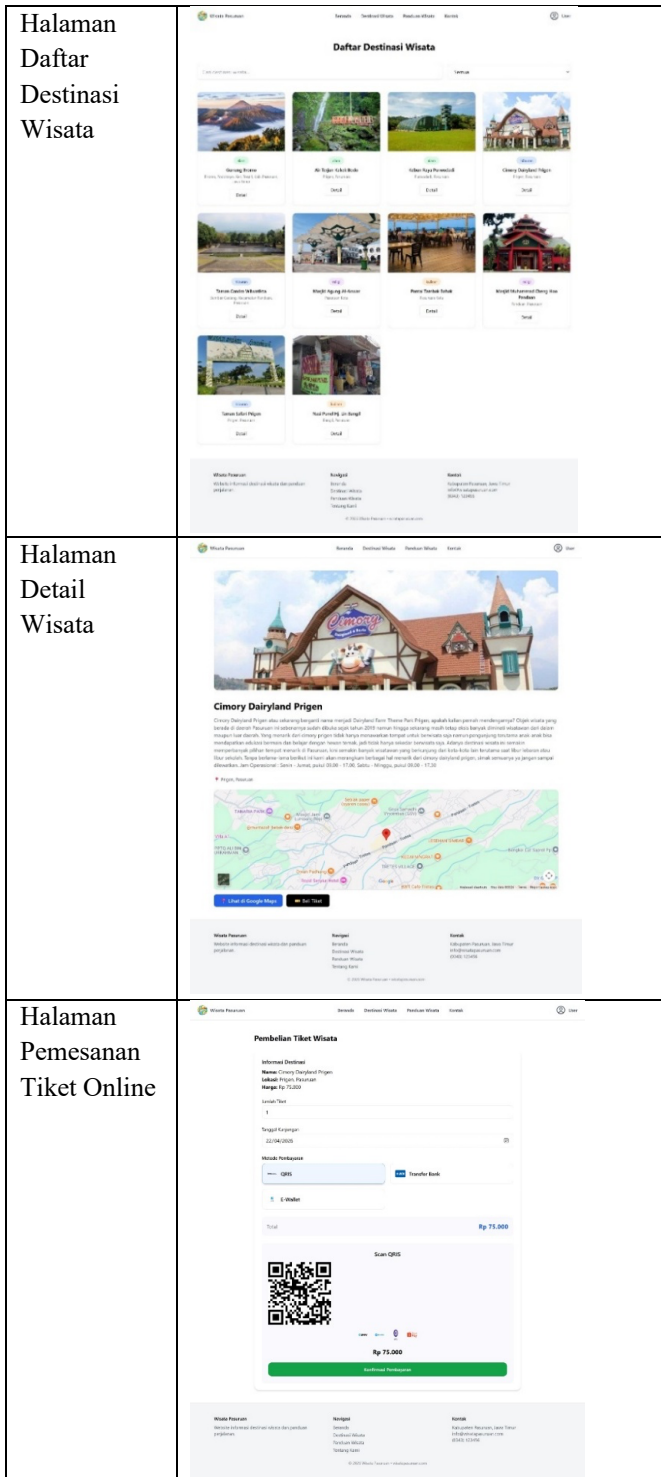
<p>Halaman Beranda</p>	
<p>Halaman Daftar Destinasi Wisata</p>	
<p>Halaman Detail Wisata</p>	
<p>Halaman Pemesanan Tiket Online</p>	

<p>Halaman Panduan Wisata</p>	
<p>Halaman Kontak Kami</p>	

Selanjutnya, setelah design mockup pada Tabel IV sudah ditentukan lalu diimplementasi menjadi halaman user seperti pada tabel berikut :

TABEL V
TABEL IMPLEMENTASI HALAMAN USER

<p>Hasil Implementasi Halaman user</p>	
<p>Halaman Beranda</p>	



Berdasarkan Tabel V menunjukkan bahwa halaman user pada website wisata Kabupaten Pasuruan telah berhasil diimplementasikan dan terintegrasi dengan sistem frontend React.js serta backend Laravel.

B. User Testing

Pengujian pengguna dilakukan menggunakan metode *Single Ease Question (SEQ)* dan *System Usability Scale (SUS)* dengan melibatkan 25 responden untuk mengukur tingkat kemudahan penggunaan dan kepuasan pengguna terhadap website wisata Kabupaten Pasuruan.

Pengujian SEQ dilakukan dengan memberikan beberapa tugas kepada responden, seperti registrasi, login, mencari destinasi wisata, melihat detail wisata, menggunakan Google Maps, membeli tiket online, melihat riwayat pembelian tiket, dan logout.

TABEL VI
TABEL NILAI RATA-RATA SEQ

Task	F1	F2	F3	F4	F5	F6	F7	F8
Rata-rata	6,4	6,4	6,36	6,24	6,12	6,12	6	6,6
Median	6,3							

Hasil pengujian pada Tabel VI menunjukkan bahwa seluruh task memperoleh nilai rata-rata di atas standar SEQ yaitu 5,5 sehingga sistem dinilai mudah digunakan oleh pengguna. Task dengan nilai tertinggi terdapat pada fitur logout dengan nilai rata-rata sebesar 6,6, sedangkan task dengan nilai terendah

terdapat pada fitur riwayat pembelian tiket dengan nilai rata-rata sebesar 6. Mayoritas responden memberikan nilai pada rentang 6 hingga 7 yang menunjukkan bahwa antarmuka website, navigasi sistem, serta fitur-fitur utama yang dikembangkan telah membantu pengguna dalam menyelesaikan tugas dengan baik. Implementasi fitur seperti pencarian wisata, Google Maps, dan pembelian tiket online juga dinilai sesuai dengan kebutuhan pengguna terhadap sistem wisata yang praktis dan mudah digunakan.

Selain pengujian SEQ, dilakukan pula pengujian menggunakan metode SUS untuk mengukur tingkat *usability* sistem secara keseluruhan. Hasil pengujian SUS memperoleh nilai rata-rata sebesar 84,1 yang termasuk ke dalam kategori Grade B dengan tingkat *acceptability* "Acceptable". Mayoritas responden memberikan nilai tinggi pada pertanyaan positif seperti kemudahan penggunaan sistem, kesesuaian fungsi fitur, serta kemudahan memahami navigasi website. Selain itu, sebagian besar responden juga memberikan nilai rendah pada pertanyaan negatif, seperti sistem dianggap rumit atau membingungkan. Hasil tersebut menunjukkan bahwa website wisata Kabupaten Pasuruan memiliki tingkat *usability* yang baik dan dapat diterima oleh pengguna sebagai media informasi dan pemesanan wisata berbasis web.

IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian website wisata Kabupaten Pasuruan menggunakan pendekatan *Component-Driven Development* (CDD), diperoleh kesimpulan bahwa penerapan pendekatan CDD berhasil dilakukan melalui penggunaan komponen berbasis *Atomic Design* yang terdiri dari *atoms*, *molecules*, *organisms*, *templates*, dan *pages*. Pendekatan tersebut menghasilkan antarmuka yang modular, terstruktur, konsisten, serta mendukung *reusability* komponen pada berbagai halaman sistem sehingga proses pengembangan dan pemeliharaan sistem menjadi lebih efisien. Integrasi antara *frontend* React JS dan *backend* Laravel melalui API berhasil diterapkan dengan baik sehingga sistem mampu menampilkan informasi destinasi wisata secara dinamis serta mendukung fitur utama seperti pencarian destinasi, melihat detail wisata, dan simulasi pemesanan tiket secara terintegrasi. Hasil pengujian menggunakan React Testing Library, Storybook, Google Lighthouse, *System Usability Scale* (SUS), dan *Single Ease Question* (SEQ) juga menunjukkan bahwa sistem dapat berjalan dengan baik, memiliki tampilan yang konsisten, serta mudah digunakan oleh pengguna. Pengujian SUS memperoleh nilai sebesar 84,1 yang termasuk kategori *Acceptable*, sedangkan hasil SEQ memperoleh nilai rata-rata sebesar 6,3 yang menunjukkan bahwa website memiliki tingkat kemudahan penggunaan yang baik.. Selain itu, hasil evaluasi

metrik kualitas komponen menunjukkan nilai *Reusability Index* (RI) sebesar 80% yang menandakan sebagian besar komponen berhasil digunakan kembali pada berbagai halaman sistem. Nilai *Test Coverage Percentage* (TCP) sebesar 100% menunjukkan seluruh fungsi komponen telah tercakup dalam pengujian. *Component Cohesion* (CpCoh) memperoleh nilai $\pm 90\%$ yang mengindikasikan bahwa setiap komponen memiliki fokus fungsi yang baik dan terstruktur. Sementara itu, nilai *Execution Time Success Rate* (ETSR) sebesar 100% menunjukkan bahwa seluruh komponen mampu berjalan dengan sukses dan stabil sesuai batas waktu eksekusi yang ditentukan. Dengan demikian, penerapan pendekatan *Component-Driven Development* (CDD) terbukti mampu menghasilkan website wisata yang modular, reusable, stabil, dan memiliki *usability* yang baik.

REFERENSI

- [1] Beauty, S. I., & Suartana, I. M. (2024). Implementasi Website Promosi Desa Wisata dengan Static Site Generator untuk Search Engine Optimization (Studi Kasus Desa Wisata Sendangduwur). *Journal of Informatics and Computer Science*, 06.
- [2] Mirmawati, N., & Qoiriah, A. (2024a). Rancang Bangun Website Obyek Wisata Kabupaten Tuban "Tuban Explore" Menggunakan Metode User Centered Design (UCD). *Journal of Informatics and Computer Science*, 06.
- [3] Abdulhakeem, A., Saeed, S., Mustafa, N., Ibrahim, M. A. M., & Al-Tayar, B. (2024). Enhancing Frontend Efficiency: Reusability and Testability in Component-Driven Development. *4th International Conference on Emerging Smart Technologies and Applications, ESmarTA 2024*. <https://doi.org/10.1109/eSmarTA62850.2024.10638895>
- [4] Nursaid, F. F., Hendra Brata, A., & Kharisma, A. P. (2020b). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri) (Vol. 4, Issue 1). <http://j-ptiik.ub.ac.id>.
- [5] Foster, C. (n.d.). *Component Driven Development with Storybook*.
- [6] LadiTech. (2025, July 27). *Component-Driven Development: A Comprehensive Guide for Modern Web Development*. <https://www.laditech.com/component-driven-development-a-comprehensive-guide-for-modern-web-development>.
- [7] Putra, F. P. E., Efendi, R. W., Tamam, A. B., & Pramadi, W. A. (2025). Tren dan Praktik Terbaik dalam Pengembangan Web Berbasis API : Kajian Literatur terhadap Framework Laravel dan React. *Infomatek*, 27(1), 165–178. <https://doi.org/10.23969/infomatek.v27i1.25122>
- [8] Alrubae, A. U., Cetinkaya, D., Liebchen, G., & Dogan, H. (2020). A process model for component-based model-driven software development. *Information (Switzerland)*, 11(6). <https://doi.org/10.3390/info11060302>
- [9] Khan, F., Tahir, M., Arif, F., Babar, M., & Khan, S. (2016). Framework for Better Reusability in Component Based Software Engineering. *J. Appl. Environ. Biol. Sci*, 6(4S), 77–81. www.textroad.com.
- [10] Homepage, J., Federick, J., & Pakereng, M. A. I. (2020). IJCIT (Indonesian Journal on Computer and Information Technology) Test-Driven Development pada Pengembangan Aplikasi Android untuk Memantau COVID-19. In *IJCIT (Indonesian Journal on Computer and Information Technology)* (Vol. 6, Issue 1). <https://creativecommons.org/licenses/by-sa/4.0/>