

Rancang Bangun Aplikasi Menyedia Dan Pengguna *API* Destinasi Wisata Pulau Jawa Berbasis Website

Moch Dani Ferdian Saputra¹, Muhammad Mega Nugraha², Danendra Alvyn Ashari³, Muhammad Muharrom Al Haromainy^{4*}

^{1,2,3,4*} Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur

122081010147@student.upnjatim.ac.id

222081010213@student.upnjatim.ac.id

322081010301@student.upnjatim.ac.id

*Corresponding author email: muhammad.muharrom.if@upnjatim.ac.id

Abstrak : Pulau Jawa memiliki jumlah destinasi wisata terbanyak di Indonesia, namun informasi mengenai objek-objek wisata tersebut seringkali tidak dikelola secara optimal dan sulit diakses secara digital. Website pariwisata yang ada cenderung statis dan kurang interaktif, sehingga tidak efektif dalam menarik minat pengunjung. Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi berbasis website yang berfungsi ganda, yaitu sebagai penyedia (*provider*) dan pengguna (*user*) *API* untuk destinasi wisata di Pulau Jawa. Metode pengembangan aplikasi ini meliputi tahap analisis kebutuhan, perancangan sistem menggunakan *CDM*, *PDM*, dan *UML*, serta implementasi dengan arsitektur terpisah antara *backend* dan *frontend*. Layanan *API Gateway* dibangun menggunakan *framework Laravel* untuk menyediakan data yang terstruktur dan dinamis, sementara antarmuka pengguna (portal wisata) dikembangkan dengan *library React JS* untuk menciptakan pengalaman pengguna yang interaktif dan responsif. Pengujian sistem dilakukan dengan metode *blackbox testing*. Hasil penelitian ini adalah sebuah sistem fungsional yang terdiri dari *API Gateway* yang memungkinkan pengguna untuk mendapatkan kunci *API* dan dokumentasi, serta Portal Wisata yang dapat menampilkan, menambah, dan menghapus data destinasi wisata. Hasil pengujian menunjukkan bahwa seluruh fungsionalitas sistem berjalan sesuai dengan yang diharapkan, menyediakan solusi modern untuk sentralisasi dan distribusi informasi pariwisata di Pulau Jawa secara efisien dan terbuka.

Kata Kunci : *API*, Destinasi Wisata, Pulau Jawa, *Laravel*, *React JS*

I. PENDAHULUAN

Pulau Jawa merupakan salah satu wilayah dengan destinasi wisata terbanyak di Indonesia. Pulau ini memiliki potensi pariwisata yang sangat besar, mulai dari keindahan alam pegunungan, kawasan pantai, hingga situs-situs sejarah dan budaya yang menjadi daya tarik wisatawan lokal maupun mancanegara. Meski demikian, masih banyak destinasi wisata yang belum terdokumentasi dengan baik, dan belum tersedia dalam bentuk informasi digital yang mudah diakses oleh masyarakat [1].

Pariwisata sebagai salah satu sektor strategis dalam pembangunan ekonomi nasional memerlukan dukungan teknologi informasi untuk mendukung penyebaran informasi destinasi secara cepat dan akurat. Berbagai situs resmi pariwisata daerah masih bersifat statis, kurang interaktif, dan tidak menyediakan fitur pencarian yang komprehensif, sehingga belum mampu menjawab kebutuhan masyarakat dalam menemukan informasi wisata secara tepat [2], [3].

Studi sebelumnya telah membuktikan bahwa sistem informasi geografis (*WebGIS*) dapat membantu penyampaian informasi wisata secara visual dan dinamis. Penelitian di Kabupaten Kudus dan Lombok Timur menunjukkan bahwa integrasi data spasial dengan *Google Maps API* dapat mempermudah akses lokasi dan menampilkan informasi destinasi secara lebih menarik [4], [5]. Namun, sebagian besar sistem yang telah dikembangkan masih terfokus pada visualisasi data, belum menyediakan fitur pencarian berbasis data terstruktur, dan tidak memungkinkan pengguna untuk mengelola atau memperbarui informasi wisata secara langsung.

Penelitian ini bertujuan untuk mengembangkan sistem *WebGIS* berbasis *API* yang memungkinkan pengguna untuk mencari dan mengelola data destinasi wisata di Pulau Jawa. Sistem ini menyediakan informasi penting seperti nama wisata, alamat, tautan *Google Maps*, dan rating destinasi secara langsung, guna memudahkan pencarian dan pengambilan keputusan bagi pengguna. Disamping itu, sistem juga menyediakan antarmuka pengelolaan data bagi pihak terkait, sehingga informasi yang tersedia dapat diperbarui secara berkala dan tetap relevan.

Perbedaan utama sistem yang dikembangkan dengan sistem sejenis terletak pada kemampuannya untuk menyediakan data wisata dalam format terstruktur, terbuka, dan dapat dicari secara spesifik, serta fitur pengelolaan langsung oleh pengguna atau admin. Sistem ini diharapkan tidak hanya menjadi sumber informasi, tetapi juga alat bantu kolaboratif dalam pengembangan dan pengelolaan data wisata yang akurat, terkini, dan dapat digunakan lintas platform.

II. KAJIAN PUSTAKA

A. Website

Website merupakan sekumpulan halaman yang saling terhubung dan dapat diakses melalui jaringan internet menggunakan domain tertentu. Halaman-halaman ini berisi informasi, gambar, video, dan elemen interaktif lainnya yang ditampilkan melalui browser. Sebuah website dirancang untuk menyediakan informasi, layanan, atau fungsi tertentu, baik untuk individu, kelompok, perusahaan, maupun institusi pemerintah. Kualitas suatu website tidak hanya ditentukan oleh kelengkapan informasi yang disajikan, tetapi juga oleh tampilan visual, struktur navigasi, kecepatan akses, dan

kemudahan penggunaan yang disesuaikan dengan kebutuhan penggunanya [6].

Dalam konteks pengembangan sistem informasi pariwisata, website memegang peranan penting sebagai media utama dalam menyebarkan informasi kepada masyarakat luas. Website pariwisata idealnya tidak hanya menampilkan informasi dasar seperti nama objek wisata dan lokasi, tetapi juga dilengkapi dengan deskripsi lengkap, galeri foto, ulasan pengunjung, informasi fasilitas, serta petunjuk arah berbasis peta interaktif. Website yang didesain dengan pendekatan user-centered design mampu meningkatkan keterlibatan pengguna dan mendorong peningkatan kunjungan wisata, baik secara fisik maupun virtual.

Peran website menjadi semakin krusial ketika dikombinasikan dengan teknologi berbasis *Application Programming Interface (API)*, yang memungkinkan penyediaan data secara dinamis dan terstruktur kepada pengguna eksternal, termasuk pengembang aplikasi *mobile*, *platform* perjalanan, dan instansi terkait. Dengan adanya integrasi ini, sebuah website dapat berkembang menjadi pusat distribusi data wisata yang tidak hanya informatif, tetapi juga adaptif terhadap kebutuhan pengguna modern yang mengandalkan layanan digital secara real time.

B. API (Application Programming Interface)

API adalah antarmuka pemrograman yang memungkinkan dua sistem perangkat lunak untuk saling berkomunikasi dan bertukar data secara terstruktur. Dalam pengembangan pariwisata digital, *RESTful API* sangat penting karena memungkinkan penyedia dan konsumen data destinasi wisata saling terintegrasi secara dinamis dan real-time. Misalnya, pada sistem reservasi *tour & travel* di Salatiga, implementasi *Web Service RESTful* menggunakan *Laravel* terbukti meningkatkan interoperabilitas, efisiensi respons, serta kemudahan pengelolaan data paket wisata dan reservasi [7]. Studi lain di lingkungan pemerintah menunjukkan bahwa *REST API* yang dibangun dengan *Laravel 9* untuk sistem pendataan kapal dengan *frontend NextJS* menghasilkan pertukaran data yang cepat, aman, serta handal lewat format *JSON* [8].

Dengan demikian, penerapan API berbasis *Laravel* pada aplikasi destinasi wisata Jawa Timur memungkinkan distribusi data objek dan fasilitas wisata secara fleksibel baik untuk aplikasi *mobile*, website pihak ketiga, maupun sistem pemetaan serta memastikan konsistensi dan kualitas layanan data dalam ekosistem digital pariwisata.

C. Laravel

Laravel merupakan *framework* PHP berbasis arsitektur *Model-View-Controller (MVC)* yang dirancang untuk mendukung proses pengembangan aplikasi web dengan sintaks yang elegan, struktur kode yang bersih, dan kemudahan integrasi berbagai layanan modern seperti autentikasi, validasi data, dan pembuatan API. Dalam pengembangan sistem informasi, *Laravel* mempermudah pengelolaan basis data, pengaturan *routing*, serta integrasi dengan *front-end modern* secara efisien. Sebuah penelitian mengenai pembangunan sistem informasi *Tour & Travel* di Bali membuktikan bahwa

Laravel mampu mempercepat proses pengembangan sekaligus menjamin keamanan dan keandalan sistem karena didukung fitur-fitur bawaan seperti *Eloquent ORM*, *middleware*, dan *migration* [9].

Penelitian lain juga menunjukkan efektivitas *Laravel* pada pengembangan aplikasi pendataan peminjaman aset, di mana *framework* ini digunakan untuk membangun *REST API* yang stabil dan modular, serta mudah dalam proses dokumentasi dan pemeliharaan [10]. Berdasarkan studi-studi tersebut, *Laravel* menjadi pilihan yang tepat dalam pembangunan aplikasi penyedia dan pengguna API destinasi wisata karena mampu membangun sistem *backend* yang skalabel, terstruktur, dan siap untuk integrasi layanan spasial maupun non-spasial.

D. React

React merupakan *library JavaScript* yang dikembangkan oleh *Meta (Facebook)* dan dirancang untuk membangun antarmuka pengguna berbasis komponen. *React* memanfaatkan konsep *Virtual DOM* untuk mempercepat proses rendering halaman, sehingga pengalaman pengguna menjadi lebih responsif dan dinamis. Dalam konteks pengembangan aplikasi destinasi wisata, *React* sangat mendukung pembuatan antarmuka interaktif yang mampu menampilkan peta, galeri, serta informasi destinasi secara *real-time*. Sebuah studi dalam pengembangan aplikasi sistem informasi desa menunjukkan bahwa penerapan *React* menghasilkan performa antarmuka yang cepat, modular, dan mudah dikelola [11].

Penelitian lain menyatakan bahwa *React* mampu mengurangi kompleksitas pada pengembangan aplikasi *front-end* skala besar karena didukung fitur seperti *one-way data binding*, *component reuse*, dan *JSX (JavaScript XML)* [12]. Dalam pengembangan aplikasi pariwisata berbasis web, integrasi *React* dengan *REST API* memungkinkan sinkronisasi data destinasi secara asinkron serta meningkatkan efisiensi pengguna dalam menelusuri informasi secara spasial maupun tematik.

III. METODE PENELITIAN

A. Analisis Kebutuhan

Tahap Analisis Kebutuhan merupakan fondasi penting dalam proses pengembangan aplikasi karena menjadi dasar dalam merancang struktur data, cakupan fitur, serta kebutuhan pengguna sistem. Studi oleh Tjahyanti dan Utama menunjukkan bahwa analisis kebutuhan yang dilakukan secara menyeluruh mampu menghasilkan sistem informasi yang responsif dan berkelanjutan [13]. Hal ini dapat mengurangi risiko kesalahan dalam pengembangan, mempercepat waktu implementasi, serta meningkatkan kepuasan pengguna akhir. Selain itu, penelitian oleh Burkin menegaskan bahwa rekayasa kebutuhan (*requirements engineering*) yang efektif berperan krusial dalam meminimalkan ambiguitas, kesalahan komunikasi, dan kebutuhan yang tidak terdokumentasi sejak tahap awal [14].

Berdasarkan prinsip tersebut, penelitian ini memulai tahapan dengan analisis kebutuhan data dan sistem untuk merancang aplikasi penyedia dan pengguna API destinasi wisata

berbasis website. Fokus wilayah sistem adalah Pulau Jawa, yang terdiri dari enam provinsi: Banten, DKI Jakarta, Jawa Barat, Jawa Tengah, DI Yogyakarta, dan Jawa Timur. Pada tahap ini, dilakukan identifikasi wilayah administratif, yaitu seluruh kabupaten dan kota dari masing-masing provinsi.

Setelah wilayah teridentifikasi, dilakukan pengumpulan data objek wisata. Setiap kabupaten atau kota diwakili oleh minimal 5 objek wisata yang dipilih secara purposive berdasarkan kelengkapan informasi spasial (*koordinat*), deskriptif, serta media pendukung seperti rating. Sumber data diperoleh dari situs web pariwisata resmi, agregator wisata yang tersedia secara daring.

Selanjutnya, seluruh data divalidasi dan diklasifikasikan agar dapat disimpan dalam basis data sistem yang mendukung *API*. Tujuannya agar data yang tersedia dapat dimanfaatkan baik oleh pengguna antarmuka website maupun oleh pengembang pihak ketiga yang mengakses melalui *endpoint API*.

Melalui tahapan ini, sistem yang dibangun diharapkan mampu menyediakan informasi destinasi wisata secara akurat, terstruktur, dan mudah diakses, serta mendorong pemanfaatan teknologi digital dalam promosi wisata di Pulau Jawa, khususnya di wilayah Jawa Timur yang menjadi fokus implementasi utama.

B. Perancangan Aplikasi dan Sistem

Perancangan sistem dalam penelitian ini mencakup perencanaan struktur data dan alur interaksi pengguna guna menghasilkan sistem penyedia dan pengguna *API* destinasi wisata Pulau Jawa berbasis website yang terstruktur dan mudah dikembangkan. Perancangan dimulai dengan *Conceptual Data Model (CDM)* yang digunakan untuk menggambarkan hubungan antarentitas utama seperti provinsi, kabupaten/kota, destinasi wisata, dan kategori wisata. *CDM* ini merepresentasikan struktur konseptual data yang akan menjadi dasar pengembangan basis data dan memastikan bahwa setiap entitas memiliki atribut yang relevan serta saling terhubung.

Model ini kemudian diubah menjadi *Physical Data Model (PDM)* sebagai representasi fisik dalam basis data *MySQL*, di mana setiap entitas diterjemahkan menjadi tabel yang lengkap dengan atribut, tipe data, relasi antar tabel, kunci primer, serta indeks untuk mendukung efisiensi pengambilan data. *PDM* dirancang agar mendukung kebutuhan akses *API* yang cepat, valid, dan terstruktur baik dari sisi spasial (*koordinat*) maupun *non-spasial (informasi deskriptif)*. Menurut penelitian yang dilakukan oleh Adiguna, penyusunan model data yang tepat berdampak langsung terhadap kestabilan dan performa sistem informasi berbasis web, terutama dalam pemrosesan data spasial yang kompleks [15].

Selanjutnya, sistem juga dirancang menggunakan diagram *Use Case* yang menggambarkan bagaimana aktor seperti admin dan pengguna berinteraksi dengan sistem. *Use case* utama meliputi pengelolaan data wilayah dan destinasi oleh admin, serta akses informasi wisata oleh pengguna melalui website atau *API*. Model ini memberikan gambaran umum mengenai fungsionalitas sistem dari sudut pandang pengguna. Untuk menggambarkan alur proses dalam sistem, digunakan *Activity Diagram* yang menjelaskan tahapan logis dari proses

permintaan data hingga penyajian informasi. Diagram ini merepresentasikan bagaimana data dikelola oleh sistem dan bagaimana pengguna berinteraksi dengan antarmuka secara runtut. Sejalan dengan studi oleh Andriani, penggunaan pemodelan seperti *use case* dan *activity diagram* secara sistematis mampu memperjelas batasan sistem dan mempercepat proses validasi kebutuhan sistem oleh pemangku kepentingan [16].

Melalui proses perancangan ini, sistem yang dikembangkan diharapkan mampu memberikan layanan informasi wisata yang dinamis, akurat, dan dapat dimanfaatkan oleh pengembang aplikasi pihak ketiga secara fleksibel melalui *endpoint API* yang tersedia.

C. Implementasi Aplikasi

Tahap implementasi merupakan proses merealisasikan rancangan sistem menjadi aplikasi yang berjalan secara fungsional. Pada penelitian ini, sistem dibagi menjadi dua komponen utama, yaitu layanan *API Gateway* yang dibangun menggunakan framework *Laravel*, serta antarmuka pengguna (*client-side*) yang dikembangkan menggunakan *React*. Pendekatan ini dipilih untuk memisahkan tanggung jawab antara penyedia layanan data dan pengguna layanan, sekaligus memungkinkan skalabilitas dan integrasi lintas platform.

Laravel digunakan sebagai backend karena kemampuannya dalam membangun *RESTful API* secara efisien serta dukungan ekosistemnya terhadap pengelolaan routing, autentikasi, validasi data, dan manajemen basis data. *API* yang dibangun mencakup berbagai endpoint publik untuk menyajikan informasi destinasi wisata, kategori wisata, dan detail lokasi berdasarkan wilayah provinsi maupun kabupaten/kota di Pulau Jawa. *Endpoint* tersebut mengembalikan data dalam format *JSON* agar mudah dikonsumsi oleh berbagai klien, termasuk sistem frontend yang dibangun terpisah. *Laravel* juga menyediakan fitur seperti *Eloquent ORM* untuk mempermudah *query* basis data, serta *middleware* untuk menangani kontrol akses dan kecepatan respons.

Di sisi frontend, *React* digunakan karena kemampuannya dalam membangun antarmuka pengguna yang interaktif, dinamis, dan responsif. Aplikasi frontend memanfaatkan *fetch API* untuk berkomunikasi dengan server *Laravel* melalui *endpoint* yang telah disediakan. Data yang diperoleh dari *API* kemudian ditampilkan dalam bentuk antarmuka visual yang menarik dan ramah pengguna, termasuk tampilan daftar destinasi wisata, peta lokasi dengan integrasi *Google Maps API*, dan fitur pencarian berdasarkan kategori atau lokasi geografis. Pemisahan frontend dan backend ini juga memudahkan proses pengembangan dan pemeliharaan sistem, karena memungkinkan pengujian dan pengembangan dilakukan secara paralel tanpa saling mengganggu.

D. Uji Coba Aplikasi

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur yang telah diimplementasikan berjalan sesuai dengan kebutuhan fungsional yang telah ditentukan. Metode pengujian yang digunakan pada penelitian ini adalah *blackbox testing*, yaitu metode pengujian yang berfokus pada pengujian

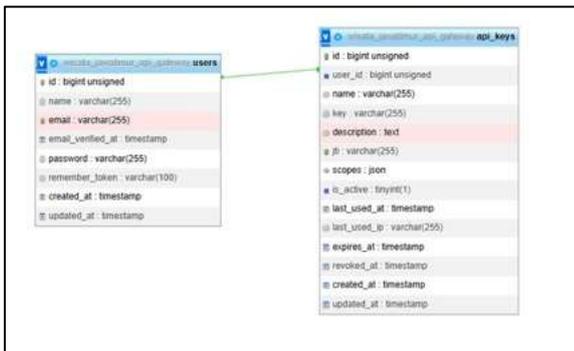
fungsionalitas sistem tanpa memperhatikan struktur internal kode program. Pengujian dilakukan dengan memberikan input pada sistem dan mengamati output yang dihasilkan, apakah sudah sesuai dengan yang diharapkan atau belum.

Pengujian dilakukan pada dua bagian utama sistem, yaitu *API Gateway* yang dibangun menggunakan Laravel dan antarmuka pengguna yang dibangun dengan React. Pada bagian *API*, pengujian mencakup validitas endpoint seperti permintaan data daftar destinasi wisata, pencarian berdasarkan wilayah dan kategori, serta akses detail informasi tiap lokasi. Pengujian dilakukan dengan mengakses endpoint menggunakan tools seperti *Postman* dan browser, lalu memeriksa apakah sistem memberikan respon data dalam format *JSON* yang valid, serta memverifikasi status kode *HTTP* (seperti 200 OK, 404 *Not Found*, atau 500 *Internal Server Error*) untuk setiap permintaan.

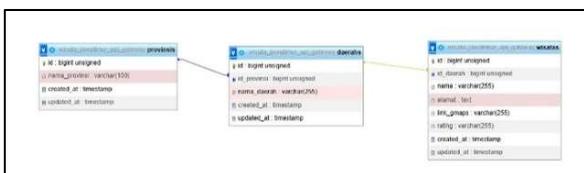
Sementara itu, pada bagian antarmuka pengguna, pengujian dilakukan terhadap setiap komponen utama yang terhubung langsung ke *API*. Hal ini meliputi pengujian tampilan halaman daftar destinasi, fitur pencarian, dan integrasi peta interaktif. Setiap aksi pengguna, seperti memilih provinsi atau mengetik kata kunci pencarian, diuji untuk memastikan hasilnya sesuai dan terhubung dengan data yang diberikan oleh *API*. Jika terdapat input tidak valid atau tidak tersedia, sistem harus tetap dapat menanganinya dengan menampilkan pesan kesalahan yang informatif kepada pengguna.

IV. HASIL DAN PEMBAHASAN

A. CDM & PDM



Gambar 1 CDM & PDM User dan *API Key*



Gambar 2 CDM & PDM Data *API*

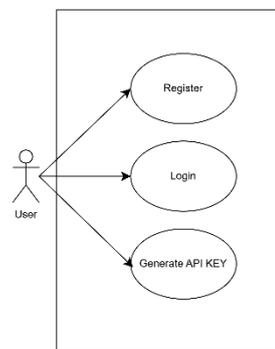
Gambar 1 dan 2 atas menunjukkan bahwa sistem ini terdiri dari beberapa tabel yang saling terkait untuk mengelola data pengguna, kunci *API*, informasi provinsi, daerah, dan tempat wisata. Tabel *users* menyimpan data pengguna seperti ID unik, nama, email, status verifikasi email, kata sandi, token "remember me", serta timestamp untuk kapan data dibuat dan terakhir diperbarui. Tabel *API_keys* berfungsi untuk mengelola kunci *API* yang terkait dengan pengguna. Setiap kunci *API* memiliki ID unik, ID pengguna yang membuat kunci tersebut

(*user_id*), nama kunci, nilai kunci (*key*), deskripsi, daftar izin (*scopes*), status aktif (*is_active*), informasi penggunaan terakhir (IP dan timestamp), tanggal kadaluarsa, serta timestamp untuk pembatalan, pembuatan, dan pembaruan data. Untuk data geografis dan wisata, terdapat tabel provinsi yang menyimpan ID unik dan nama provinsi. Tabel daerah menyimpan ID unik, nama daerah, dan memiliki hubungan (*id_provinsi*) dengan tabel provinsi, menunjukkan bahwa setiap daerah berada dalam satu provinsi. Terakhir, tabel wisata menyimpan informasi tentang tempat wisata, termasuk ID unik, nama wisata, alamat, tautan *Google Maps* (*link_gmaps*), rating, serta timestamp pembuatan dan pembaruan data. Tabel wisata ini memiliki hubungan (*id_daerah*) dengan tabel daerah, yang berarti setiap tempat wisata terasosiasi dengan sebuah daerah tertentu. Secara keseluruhan, struktur ini memungkinkan pengelolaan data yang terorganisir untuk aplikasi *Gateway API* yang berfokus pada informasi wisata di Jawa Timur.

B. Perancangan UML

Dalam perancangan UML ini kami membuat dua diagram yaitu use case diagram dan activity diagram.

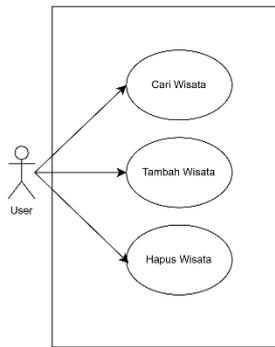
1. Use Case Diagram



Gambar 3 Use Case *API Gateway*

Diagram ini menjelaskan interaksi antara pengguna (*User*) dengan sistem *API Gateway*.

- Aktor: Terdapat satu aktor yaitu *User*, yang merupakan pengembang atau pihak ketiga yang akan menggunakan data wisata.
- Use Case: Aktor *User* dapat melakukan tiga tindakan utama pada sistem:
 1. Register: Pengguna dapat mendaftarkan akun baru ke dalam sistem.
 2. Login: Setelah memiliki akun, pengguna dapat masuk (login) ke sistem.
 3. Generate *API KEY*: Setelah berhasil login, pengguna dapat membuat sebuah kunci *API* (*API Key*) yang akan digunakan untuk mengakses data melalui endpoint yang disediakan.

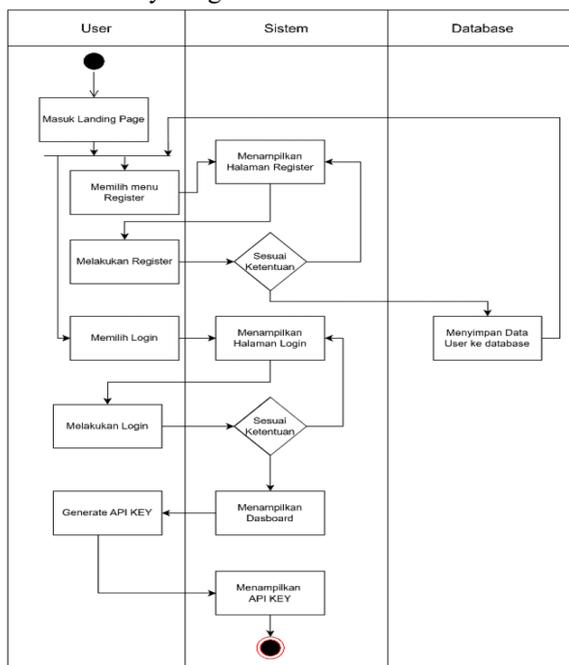


Gambar 4 Use Case Portal Wisata

Diagram ini menggambarkan fungsionalitas utama yang dapat diakses oleh pengguna (*User*) pada sistem Portal Wisata.

- **Aktor:** Sama seperti sebelumnya, aktor utamanya adalah *User*. Dalam konteks ini, *User* bisa berarti admin atau masyarakat umum yang berinteraksi dengan antarmuka website portal.
- **Use Case:** Aktor *User* memiliki tiga fungsi utama dalam mengelola data destinasi wisata:
 1. **Cari Wisata:** Pengguna dapat mencari informasi destinasi wisata yang ada di dalam sistem.
 2. **Tambah Wisata:** Pengguna dapat menambahkan data destinasi wisata baru ke dalam database.
 3. **Hapus Wisata:** Pengguna dapat menghapus data destinasi wisata yang sudah tidak relevan atau salah.

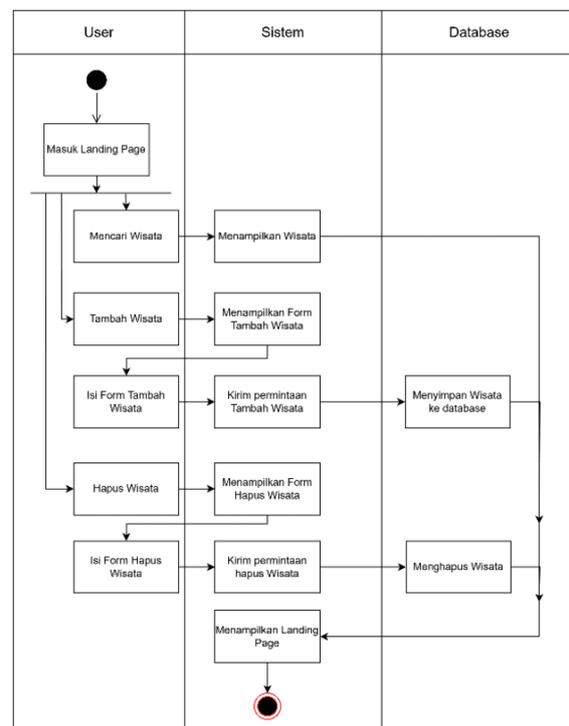
2. Activity Diagram



Gambar 5 Activity Diagram API Gateway

Diagram ini merinci alur kerja atau aktivitas pengguna saat berinteraksi dengan sistem *API Gateway*, mulai dari pendaftaran hingga mendapatkan *API Key*.

1. **Mulai:** Alur dimulai saat pengguna Masuk Landing Page.
2. **Pilihan Aksi:** Dari halaman utama, pengguna bisa memilih menu Register atau Login.
3. **Alur Registrasi:** Jika memilih Register, sistem akan menampilkan halaman registrasi. Pengguna melakukan registrasi, dan sistem akan memvalidasi data. Jika sesuai, data pengguna akan disimpan ke database, dan proses kembali ke halaman login.
4. **Alur Login:** Jika memilih Login, sistem menampilkan halaman login. Pengguna memasukkan kredensial, dan sistem akan memvalidasinya.
5. **Akses Dashboard:** Setelah login berhasil, sistem akan menampilkan Dashboard.
6. **Generate API Key:** Pengguna dapat memulai proses Generate *API KEY*. Sistem kemudian akan memproses dan menampilkan *API KEY* yang baru dibuat untuk pengguna.
7. **Selesai:** Alur berakhir setelah kunci *API* berhasil ditampilkan.



Gambar 6 Activity Diagram Portal Wisata

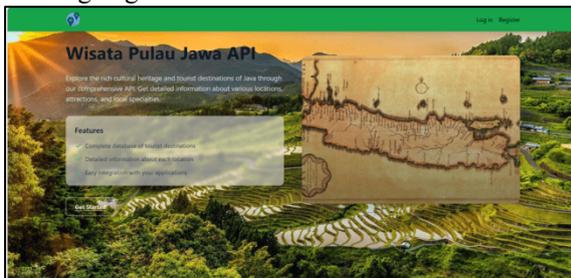
Diagram ini menjelaskan alur aktivitas pengguna dalam mengelola data destinasi pada Portal Wisata.

1. **Mulai:** Alur diawali saat pengguna Masuk Landing Page portal.

2. Pilihan Aksi: Dari halaman tersebut, pengguna dapat melakukan tiga aktivitas utama secara paralel: Mencari, Menambah, atau Menghapus data wisata.
3. Alur Mencari Wisata: Pengguna melakukan pencarian, kemudian sistem akan Menampilkan Wisata yang relevan dengan mengambil data dari database.
4. Alur Tambah Wisata: Pengguna memilih untuk menambah wisata. Sistem akan menampilkan form tambah wisata. Setelah pengguna mengisi form dan mengirimkannya, sistem akan menyimpan data wisata baru ke database.
5. Alur Hapus Wisata: Pengguna memilih untuk menghapus wisata. Sistem akan menampilkan form atau daftar untuk memilih data yang akan dihapus. Setelah pengguna mengirim permintaan, sistem akan menghapus data wisata dari database.
6. Selesai: Setelah setiap aktivitas selesai, alur dapat kembali ke halaman utama (*Landing Page*), dan proses berakhir.

C. Sistem API Gateway

1. Landing Page



Gambar 7 Landing Page API Gateway

Halaman utama menampilkan informasi umum dan akses menuju fitur utama API Gateway.

2. Register

Gambar 8 Register API Gateway

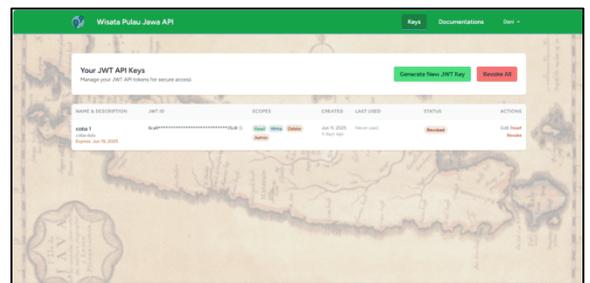
Pengguna perlu daftar terlebih dahulu dengan mengisi form register.

3. Login

Gambar 9 Login API Gateway

Setelah pengguna daftar maka pengguna perlu memasukkan email dan password agar bisa login dan mengakses Dashboard API

4. Dashboard



Gambar 10 Dashboard API Gateway

Menampilkan ringkasan aktivitas pengguna serta akses cepat ke fitur-fitur API Gateway.

5. Generate API KEY

Gambar 11 Generate API KEY

Pengguna dapat membuat API Key untuk mengakses layanan API secara aman.

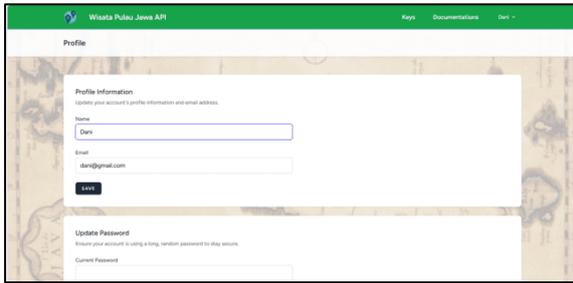
6. Documentation



Gambar 12 Documentation

Documentation berisi panduan penggunaan API lengkap dengan endpoint, metode, dan contoh respons.

7. Profil

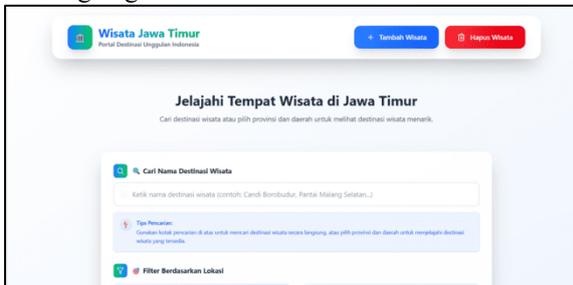


Gambar 13 Profil API Gateway

Menampilkan untuk pengguna mengelola informasi akun mereka.

D. Sistem Portal Wisata

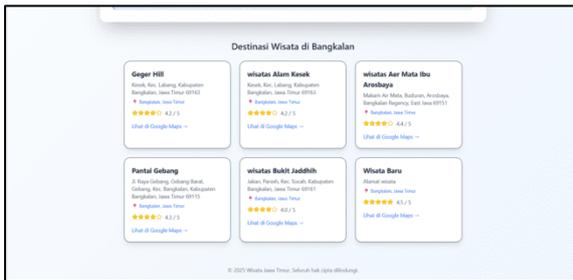
1. Landing Page



Gambar 14 Landing Page Portal Wisata

Halaman utama memungkinkan pengguna mencari tempat wisata berdasarkan nama, provinsi, dan kota/kabupaten di Pulau Jawa.

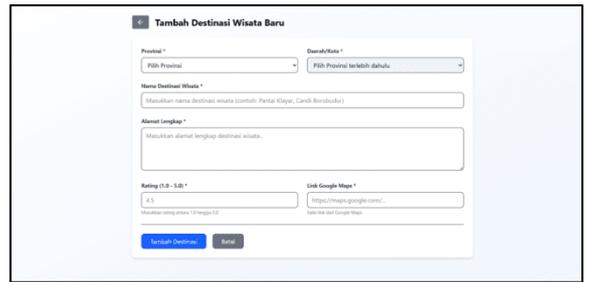
2. Daftar Wisata



Gambar 15 Daftar Wisata

Menampilkan nama wisata lengkap dengan alamat, rating, dan tautan Google Maps setelah pencarian dilakukan.

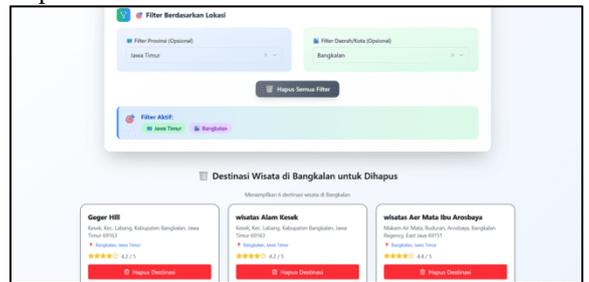
3. Tambah Wisata



Gambar 16 Tambah Wisata

Pengguna dapat menambahkan destinasi wisata baru dengan mengisi formulir berdasarkan lokasi dan informasi lengkap.

4. Hapus Wisata



Gambar 17 Hapus Wisata

Fitur ini memungkinkan pengguna menghapus wisata yang tidak valid dengan klik tombol "Hapus Destinasi".

E. Uji Coba Aplikasi

Dilakukan menggunakan metode Blackbox Testing untuk memastikan setiap fitur berjalan sesuai fungsinya. Pengujian difokuskan pada respons sistem terhadap input pengguna tanpa melihat kode program.

TABEL I
BLACKBOX TESTING LARAVEL API

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Status
	Registrasi Akun Baru	Pengguna mengisi form registrasi data valid dan menekan tombol "Register"	Sistem berhasil membuat akun baru dan menampilkan notifikasi pendaftaran berhasil	PASS
	Login	Pengguna login menggunakan akun yang telah terdaftar	Pengguna berhasil masuk ke sistem dan diarahkan ke halaman dashboard	PASS

Generate JWT API Key	Pengguna menekan tombol "Generate New API Key" di halaman manajemen token	Sistem berhasil menghasilkan dan menampilkan JWT API Key yang baru	PASS
Edit JWT API Key	Pengguna melakukan pengeditan pada nama API Key lalu menyimpan perubahan	Perubahan berhasil disimpan dan data API Key diperbarui	PASS
Reset JWT API Key	Pengguna menekan tombol "Reset" pada API Key yang aktif	Sistem menggantikan API Key lama dengan yang baru, dan menampilkan API Key terbaru	PASS
Revoke JWT API Key	Pengguna menekan tombol "Revoke" pada API Key tertentu	API Key tersebut dinonaktifkan dan tidak dapat digunakan lagi untuk autentikasi	PASS
Akses Halaman Dokumentasi	Pengguna menekan menu navigasi "Documentation"	Sistem menampilkan halaman dokumentasi API	PASS
Akses Halaman Profil	Pengguna menekan menu navigasi "Profile"	Sistem menampilkan halaman profil pengguna	PASS
Edit Informasi Profil	Pengguna mengubah nama dan email, lalu menyimpan perubahan	Informasi profil berhasil diperbarui dan sistem menampilkan notifikasi sukses	PASS
Perbarui Password	Pengguna mengisi password lama, password baru, dan konfirmasi lalu menyimpan	Password berhasil diperbarui, dan dapat digunakan untuk login berikutnya	PASS
Hapus Akun	Pengguna menekan tombol "Delete Account" dan mengonfirmasi penghapusan	Akun pengguna dihapus dari sistem, dan tidak dapat login kembali	PASS
Logout	Pengguna menekan tombol	Pengguna berhasil keluar dari sistem dan diarahkan	PASS

	"Logout" pada navbar	kembali ke halaman login	
--	----------------------	--------------------------	--

TABEL II
BLACKBOX TESTING PORTAL WISATA JAWA

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Status
1	Pencarian Nama Wisata	Pengguna mengetik nama wisata pada kolom pencarian lalu menekan tombol "Cari"	Sistem menampilkan daftar wisata yang sesuai dengan kata kunci	PASS
2	Filter Berdasarkan Lokasi	Pengguna memilih provinsi dan kota/daerah dari dropdown filter lokasi	Sistem menampilkan daftar wisata yang sesuai dengan lokasi yang dipilih	PASS
3	Tambah Wisata Baru	Pengguna menekan tombol "Tambah Wisata", mengisi form secara lengkap dan menyimpan	Sistem menyimpan data dan wisata baru muncul dalam daftar pencarian	PASS
4	Hapus Data Wisata	Pengguna menekan tombol "Hapus Wisata", mencari nama, memilih item, lalu menghapus	Sistem menghapus data wisata dan item tidak lagi muncul di daftar	PASS

V. KESIMPULAN

Penelitian ini berhasil merancang hingga mampu mengimplementasikan aplikasi penyedia dan pengguna API destinasi wisata Pulau Jawa berbasis website untuk mengatasi masalah informasi pariwisata yang terfragmentasi dan kurang optimal. Dengan memisahkan arsitektur sistem menjadi dua komponen utama layanan API Gateway menggunakan Laravel dan antarmuka pengguna (Portal Wisata) menggunakan React penelitian ini menghasilkan sebuah platform yang *fleksibel, skalabel, dan modern*.

Aplikasi yang dikembangkan telah memenuhi seluruh kebutuhan fungsional yang dirancang. Sistem API Gateway terbukti mampu mengelola pengguna, menyediakan autentikasi melalui kunci API, dan menyajikan data destinasi wisata secara terstruktur dalam format JSON. Di sisi lain, Portal Wisata sebagai pengguna API berhasil menyediakan antarmuka yang interaktif bagi pengguna untuk melakukan pencarian, penambahan, dan penghapusan data wisata, yang terhubung langsung dengan API yang tersedia.

Pengujian *blackbox* yang dilakukan pada kedua bagian sistem mengonfirmasi bahwa seluruh fitur berjalan sesuai dengan skenario yang diharapkan dan memberikan respons yang valid terhadap input pengguna. Dengan demikian, aplikasi ini tidak hanya berfungsi sebagai sebuah sistem informasi pariwisata yang fungsional, tetapi juga sebagai fondasi bagi ekosistem digital pariwisata yang lebih terbuka, di mana data

dapat diakses dan dimanfaatkan oleh berbagai pihak untuk mendukung promosi wisata di Pulau Jawa.

REFERENSI

- [1] Sari, A. M., & Lestari, R. (2018). Pengenalan Objek Wisata di Pulau Jawa Menggunakan Perangkat Mobile. *Jurnal Khatulistiwa Informatika*, 1(3), 491237.
- [2] Putra, F. K. K., Saepudin, P., Adriansyah, E., & Wahyu Adrian, I. (2018). Digital tourism: A content analysis of West Java tourism websites. *Journal of Indonesian Tourism and Development Studies*, 6(2), 73-84.
- [3] Hermawan, A., Awaluddin, M., & Yuwono, B. D. (2017). Pembuatan aplikasi webgis informasi pariwisata dan fasilitas pendukungnya di kabupaten kudus. *Jurnal Geodesi Undip*, 6(4), 51-59.
- [4] Basyri, H., Suprayogi, A., & Haniah, H. (2015). Aplikasi WebGIS Pariwisata Menggunakan Google Map *API* di Kabupaten Lombok Timur. *Jurnal Geodesi Undip*, 4(4), 8-15.
- [5] W. D. Nanda, I. Widianingsih, dan A. Z. Miftah, "The Linkage of Digital Transformation and Tourism Development Policies in Indonesia from 1879–2022," *Sustainability*, vol. 15, no. 13, 2023.
- [6] Rochmawati, I. (2019). Analisis user interface situs web iwearup. com. *Com. Visualita*, 7(2), 31-44.
- [7] K. Gowell dan S. Supriyadi, "Perancangan Web Service REST *API* Menggunakan PHP dan Framework Laravel di Tenta Tour Salatiga," *Jurnal JTJK*, vol. 8, no. 1, hlm. 49–57, Jan.–Mar. 2024.
- [8] L. S. Saputra dan H. F. Herdiyatomoko, "Implementasi REST *API* pada Sistem Pendataan Kapal Menggunakan Laravel 9 dan NuxtJS," *Journal Computer and Technology*, vol. 2, no. 1, hlm. 10–21, Jul. 2024.
- [9] Savira, N. R., Pramana, D., & Yuningsih, L. (2024, June). Sistem Informasi Layanan Tour Guide Pada Pelabuhan Benoa Menggunakan Framework Laravel. In *Seminar Hasil Penelitian Informatika dan Komputer (SPINTER)* Institut Teknologi dan Bisnis STIKOM Bali (pp. 568-573).
- [10] Adithama, S. P., & Maslim, M. (2019). Pembangunan Sistem Informasi Perpustakaan Sekolah Dasar Berbasis Web. *DINAMISIA-Jurnal Pengabdian Kepada Masyarakat*, 3(2), 350-360.
- [11] Rakhmat, G. A. (2020). Perancangan Sistem Informasi Reservasi Hotel. *Device*, 10(2), 13-19.
- [12] Maulana, R., Sulistyanto, A., & Rini, A. S. (2021). Perancangan sistem informasi pengajuan dan pelaporan pembayaran tunjangan kinerja pada lembaga pemasyarakatan salemba berbasis web menggunakan Skema Rest *API*. *Jurnal Manajemen Informatika Jayakarta*, 1(4), 283-294.
- [13] L. P. A. S. Tjahyanti dan G. R. Utama, "Peran Analisis Kebutuhan dalam Menciptakan Sistem Informasi yang Responsif dan Berkelanjutan," *KOMTEKS*, vol. 3, no. 2, 2024.
- [14] V. Burkin, "Mitigating Risks in Software Development through Effective Requirements Engineering," *arXiv preprint arXiv:2305.12478*, 2023.
- [15] Adiguna, A. R., Saputra, M. C., & Pradana, F. (2018). Analisis dan perancangan sistem informasi manajemen gudang pada PT Mitra Pinasthika Mulia Surabaya. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(2), 612-621.
- [16] Ardhiyani, R. P., & Mulyono, H. (2018). Analisis dan perancangan sistem informasi pariwisata berbasis web sebagai media promosi pada kabupaten tebo. *Jurnal Manajemen Sistem Informasi*, 3(1), 952-972.