

Aplikasi Pengelolaan Data Wilayah Menggunakan *RESTfull* API Pada Kementerian ESDM

Bagas Ahmad Faizal*, Budi Sudrajat, Hasta Herlan Asymar

Informatika, Universitas Bina Sarana Informatika, Jakarta

*bagasahmadfaizal123@gmail.com

Abstrak— Penelitian ini bertujuan untuk mengembangkan sistem informasi pengelolaan data wilayah administratif berbasis *RESTful* API dengan menggunakan framework *Laravel*. Sistem ini dirancang guna mendukung kebutuhan pengelolaan data wilayah yang meliputi empat entitas utama, yaitu provinsi, kabupaten/kota, kecamatan, dan kelurahan/desa. Pengelolaan data wilayah seringkali dilakukan secara manual atau belum terintegrasi dengan baik, sehingga memunculkan kendala dalam hal efisiensi, akurasi, serta aksesibilitas data. Untuk menjamin keamanan akses, sistem menerapkan autentikasi pengguna berbasis *JSON Web Token (JWT)* sehingga setiap pengguna yang mengakses sistem memiliki otorisasi yang valid dan dapat dipertanggungjawabkan. Sistem ini menyediakan implementasi fungsi *CRUD (Create, Read, Update, Delete)* pada setiap entitas, yang memungkinkan proses pengelolaan data dilakukan secara lebih mudah, cepat, dan terstruktur. Proses pengembangan dilakukan dengan pemisahan antara sisi *back-end* dan *front-end*, di mana keduanya saling berkomunikasi melalui *HTTP request*. Dengan pendekatan ini, sistem menjadi lebih fleksibel, skalabel, serta mudah diintegrasikan dengan aplikasi lain sesuai kebutuhan instansi. Pengujian sistem menggunakan metode *Black Box Testing* untuk memastikan seluruh fitur berjalan sesuai dengan spesifikasi yang ditetapkan. Hasil pengujian menunjukkan bahwa autentikasi serta fungsi *CRUD* pada setiap entitas dapat berfungsi dengan baik dan menghasilkan keluaran sesuai input. Dengan adanya sistem ini, diharapkan instansi terkait mampu mengelola data wilayah secara efisien, meningkatkan kualitas layanan publik, mendukung integrasi data, serta mempercepat proses digitalisasi pemerintahan yang berkelanjutan.

Article History:

Received: Oct 12, 2025

Revised: Jan 5, 2025

Accepted: Jan 20, 2026

Published: Jan 31, 2026

Kata Kunci—*Crud; Data Wilayah; Json Web Token; Laravel; Restful Api.*

DOI: 10.22441/jitkom.v10i1.008

I. PENDAHULUAN

Perkembangan teknologi informasi telah membawa dampak signifikan dalam berbagai sektor, termasuk sektor pemerintahan. Salah satu bentuk transformasi digital di pemerintahan adalah penerapan *electronic government (e-Government)*, yang bertujuan untuk meningkatkan efisiensi operasional, transparansi, serta kualitas pelayanan publik melalui pemanfaatan teknologi informasi dan komunikasi. [1] Untuk mewujudkan *e-Government* yang efektif, dibutuhkan sistem informasi yang terintegrasi dan mampu berbagi data secara cepat, akurat, dan aman antar unit maupun sistem yang berbeda.

Di lingkungan Kementerian Energi dan Sumber Daya Mineral (KESDM), telah tersedia basis data wilayah administratif dari tingkat provinsi hingga kelurahan. Namun, data tersebut belum dilengkapi dengan mekanisme integrasi yang memadai, sehingga pertukaran data masih dilakukan secara manual. Kondisi ini menimbulkan sejumlah kendala, seperti keterlambatan akses informasi, potensi kesalahan *input*, serta duplikasi data.

Penelitian terdahulu menunjukkan bahwa pemanfaatan teknologi integrasi data dapat mempermudah pengembangan sistem dan meningkatkan efisiensi pertukaran informasi. [2] Namun, pada konteks KESDM, belum tersedia sistem yang mampu menyediakan data wilayah secara terstruktur, *real-time*, dan mudah diintegrasikan dengan aplikasi lain. Berdasarkan hal tersebut, penelitian ini berfokus pada pengembangan sistem

pengelolaan data wilayah berbasis integrasi modern yang mendukung pertukaran data secara efisien dan aman. Tujuan penelitian ini adalah menghasilkan sistem informasi yang mampu mempercepat proses integrasi data wilayah di KESDM, sekaligus memperkuat tata kelola pemerintahan berbasis digital yang terbuka, responsif, dan berbasis data

II. LITERATURE REVIEW

Penelitian Terdahulu yang memiliki keterkaitan dengan pengembangan aplikasi pengelolaan data wilayah menggunakan *RESTful API* dalam konteks pemerintahan maupun instansi antara lain sebagai berikut. Penelitian [3] membahas pengelolaan transfer data kepegawaian pemerintah daerah. Penelitian ini memanfaatkan teknologi *web service* berupa *RESTful API* yang berfungsi untuk mempermudah proses integrasi data ke berbagai platform lain. Selanjutnya, penelitian pada PT Kayan Putra Utama Coal yang bergerak di bidang pertambangan batu bara menunjukkan bahwa penerapan *RESTful API* dapat digunakan untuk mempermudah serta mengintegrasikan pertukaran data sumber daya manusia dengan menggunakan *framework Laravel*. [4] Selain itu, penelitian yang dilakukan pada Balai Desa Widarapayung Kulon yang bertanggung jawab dalam pengelolaan aset dan barang juga menerapkan *NodeJS* dengan dukungan *RESTful API*. Pada penelitian ini, digunakan pula *JWT (JSON Web Token)* sebagai mekanisme keamanan dalam proses pertukaran data. [5]

III. METODOLOGI PENELITIAN

Penelitian ini dilakukan di Kementerian Energi dan Sumber Daya Mineral (ESDM), tepatnya di Pusat Data dan Informasi (PUSDATIN) yang berlokasi di Jakarta Pusat. Pemilihan instansi tersebut sebagai objek penelitian didasarkan pada kebutuhan integrasi data antar wilayah yang hingga saat ini masih dilakukan secara manual, serta belum tersedianya *API* yang sepenuhnya terintegrasi dengan baik untuk mendukung pengelolaan data.

Untuk melanjutkan penelitian ini digunakan beberapa metode menurut Mukhamad Fathoni, 2019 [6], yaitu:

- Studi Pustaka, dilakukan untuk mencari referensi terkait pengembangan aplikasi berbasis *RESTful API* sebagai landasan teoritis dalam penelitian.
- Observasi, dilakukan secara langsung pada aktivitas pengguna atau instansi terkait dengan tujuan memahami proses kerja yang sedang berjalan, sehingga dapat ditemukan potensi solusi yang dapat diterapkan melalui sistem informasi.
- Wawancara, dilakukan dengan staf pengelola sistem untuk menggali kebutuhan sistem, kendala yang dihadapi, serta fitur-fitur yang diperlukan dalam pengembangan aplikasi.

Selain metode pengumpulan data, penelitian ini juga menggunakan metode perancangan sistem dengan pendekatan Rekayasa Perangkat Lunak (RPL) melalui model pengembangan *Waterfall*, yang meliputi tahapan sebagai berikut menurut Fatmawati [7] :

- Analisis Kebutuhan, untuk mengidentifikasi permasalahan serta kebutuhan sistem berdasarkan hasil observasi dan wawancara dengan pihak Kementerian ESDM. Informasi yang diperoleh dijadikan dasar dalam merancang sistem yang sesuai.
- Perancangan Sistem, yaitu membuat rancangan arsitektur aplikasi yang mencakup struktur *RESTful API*, rancangan basis data relasional dengan *MySQL*, serta pemodelan sistem menggunakan pendekatan monolitik modular.
- Implementasi, yaitu pembangunan sistem sesuai desain menggunakan: *Laravel 11* sebagai *framework* utama, *MySQL* untuk basis data, *AuthJWT* sebagai mekanisme autentikasi, serta *HTML*, *CSS*, dan *Bootstrap 5* untuk antarmuka pengguna.
- Pengujian, dilakukan dengan metode *black-box* testing untuk memastikan sistem berjalan sesuai kebutuhan dan menghasilkan output yang benar berdasarkan input yang diberikan.

Pemeliharaan, mencakup penyusunan dokumentasi sistem serta penanganan terhadap permasalahan yang muncul pasca implementasi agar sistem dapat digunakan dan dikembangkan secara berkelanjutan

IV. HASIL DAN ANALISA

A. Arsitektur Sistem

Arsitektur sistem dirancang dengan pendekatan *client-server* berbasis *RESTful API*. Pada sisi *backend*, sistem dibangun menggunakan *framework Laravel* untuk menyediakan layanan pengelolaan data dan autentikasi pengguna. Sementara itu, pada sisi *frontend*, aplikasi memanfaatkan teknologi *web modern* untuk menampilkan data secara interaktif kepada pengguna. Komunikasi antara *frontend* dan *backend* dilakukan melalui protokol *HTTP* dengan format pertukaran data *JSON*.

1. Komponen Sistem

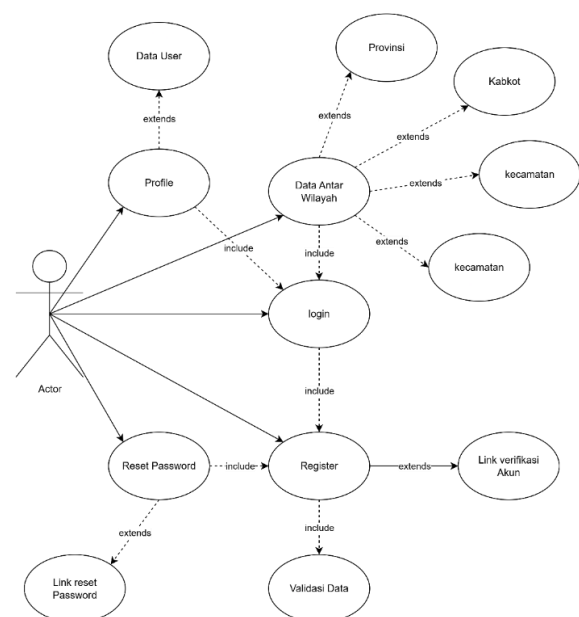
Sistem terdiri dari beberapa komponen utama, yaitu:

- Frontend (Client)*: Menyediakan antarmuka pengguna berbasis web agar pengguna dapat mengakses dan mengelola data.
- Backend (Server)*: Menggunakan *framework Laravel* untuk memproses permintaan dari pengguna, mengatur logika bisnis, dan menyediakan layanan *API*.
- Database*: Menyimpan data wilayah dan informasi terkait secara terstruktur menggunakan *MySQL*.
- API (Application Programming Interface)*: Menjadi perantara komunikasi antara *frontend* dan *backend* dengan format data *JSON*.

2. Alur Proses

Alur proses sistem dimulai ketika pengguna melakukan *login* untuk mendapatkan akses. Selanjutnya, permintaan dari pengguna dikirim melalui *frontend* ke *backend* menggunakan *HTTP request*. *Backend* kemudian memproses permintaan sesuai logika bisnis dan melakukan interaksi dengan basis data. Hasil pengolahan data dikirim kembali ke *frontend* dalam bentuk *JSON* untuk ditampilkan kepada pengguna.

a. Use Case Diagram



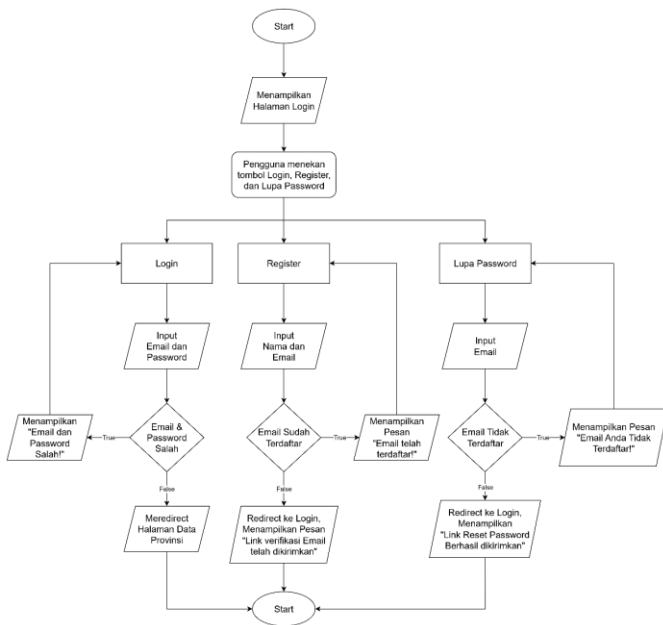
Gambar 1. Use Case Diagram

Use Case diagram menggambarkan interaksi antara aktor (pengguna) dengan sistem dalam menjalankan berbagai fungsionalitas yang tersedia. [8] Pada diagram, aktor utama adalah User yang dapat mengakses beberapa fitur utama sistem, antara lain:

- **Autentikasi:** pengguna dapat melakukan *Login*, *Register*, *Reset Password*, dan *Verifikasi Akun*. Proses ini mencakup validasi data dan pengiriman link melalui *email*.
- **Manajemen Data:** pengguna dapat mengelola data wilayah administratif berupa Provinsi, Kabupaten/Kota, Kecamatan, dan Kelurahan/Desa, termasuk melakukan operasi *CRUD*.
- **Pengelolaan Akun:** pengguna dapat mengatur *Profile* dan melihat *Data User*.
- **Data Antar Wilayah:** sistem juga menyediakan pengelolaan relasi antar data wilayah.

Relasi *include* dan *extend* pada *use case* menunjukkan keterkaitan antar proses, misalnya *login* yang membutuhkan validasi data (*include*), atau *register* yang diperluas dengan verifikasi akun (*extend*). Dengan adanya *use case* diagram ini, gambaran kebutuhan fungsional sistem menjadi lebih jelas, sehingga memudahkan proses implementasi dan pengujian.

b. *Flowchart*



Gambar 2. Flowchart

Flowchart menggambarkan alur proses autentikasi pengguna pada system. [9] Proses diawali dengan halaman *login* yang menampilkan tiga pilihan utama, yaitu *Login*, *Register*, dan *Lupa Password*.

1) *Login*:

- Pengguna memasukkan *email* dan *password*.
- Sistem memvalidasi data.
- Jika *email* atau *password* salah, sistem menampilkan pesan *error*.

d) Jika benar, sistem akan mengarahkan pengguna ke halaman *Data Provinsi*.

2) *Register*:

- Pengguna memasukkan nama dan *email*.
- Sistem memeriksa apakah *email* sudah terdaftar.
- Jika sudah, ditampilkan pesan bahwa *email* sudah digunakan.
- Jika belum, sistem mengirimkan *link* verifikasi ke *email* pengguna, lalu mengarahkan kembali ke halaman *login* dengan pesan bahwa verifikasi telah dikirimkan.

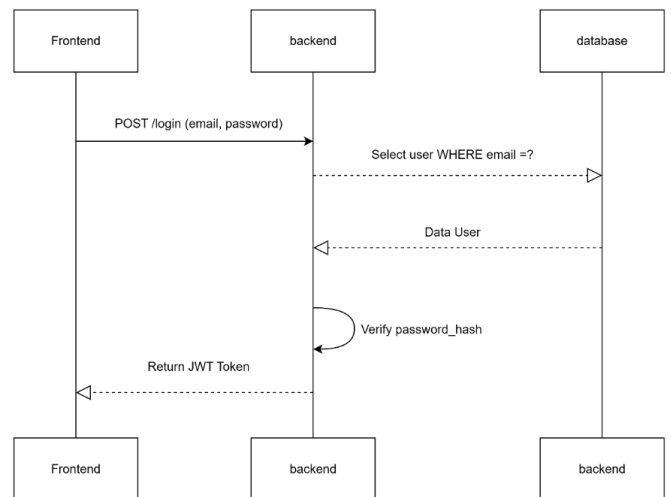
3) *Lupa Password*:

- Pengguna memasukkan *email* untuk reset *password*.
- Sistem memeriksa apakah *email* terdaftar.
- Jika tidak, ditampilkan pesan bahwa *email* belum terdaftar.
- Jika benar, sistem mengirimkan *link reset password* ke *email* dan mengarahkan kembali ke halaman *login* dengan pesan bahwa link berhasil dikirimkan.

Flowchart ini memberikan gambaran visual mengenai alur autentikasi pengguna dalam sistem, sehingga memudahkan pengembang maupun pengguna untuk memahami mekanisme *login*, *register*, dan pemulihan *password*.

c. *Sequence Diagram*

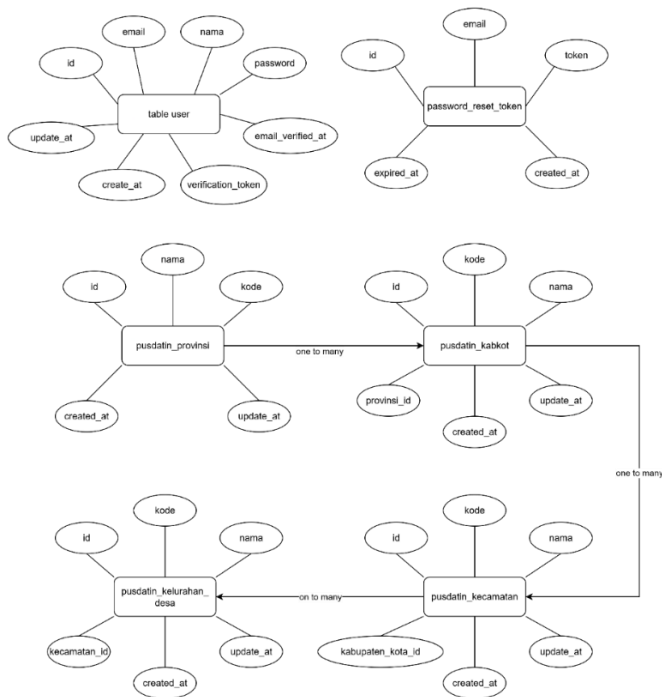
Sequence diagram menggambarkan alur komunikasi antar komponen utama, yaitu *Frontend*, *Backend*, dan *Database*. [10]



Gambar 3. Sequence Diagram

3. *Perancangan Basis Data*

Perancangan basis data menggunakan *Entity Relationship Diagram (ERD)* untuk memodelkan data wilayah administratif. ERD ini menggambarkan relasi antar entitas utama, yaitu Provinsi, Kabupaten/Kota, Kecamatan, dan Kelurahan/Desa.



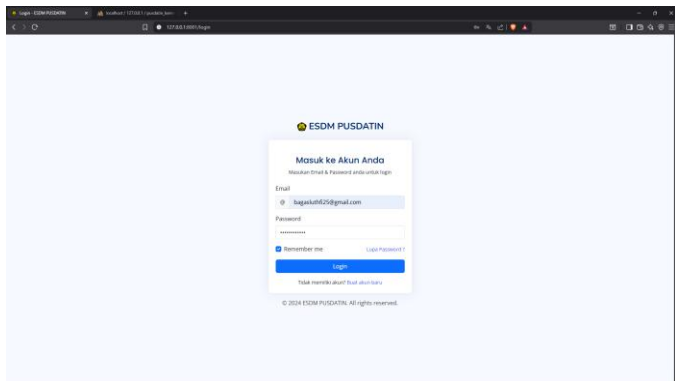
Gambar 4. Entity Relationship Diagram (ERD)

B. Implementasi Fungsi-fungsi Utama

Pada tahap implementasi, sistem dibangun menggunakan *Laravel 11* dengan konsep *RESTful API* sebagai media pertukaran data antara *back-end* dan *front-end*. Fokusnya pada pengelolaan data wilayah yang mencakup empat entitas utama: Provinsi, Kabupaten/Kota, Kecamatan, dan Kelurahan/Desa.

Keamanan akses dijamin dengan *JSON Web Token (JWT)* sehingga hanya pengguna yang terautentikasi dapat mengakses data. *API* dirancang mengikuti prinsip *separation of concerns*, di mana *back-end* menyediakan data dalam format *JSON* dan *front-end* menampilkan data kepada pengguna. Pendekatan ini membuat sistem lebih fleksibel, terstruktur, dan mudah diintegrasikan dengan aplikasi lain.

1. Login



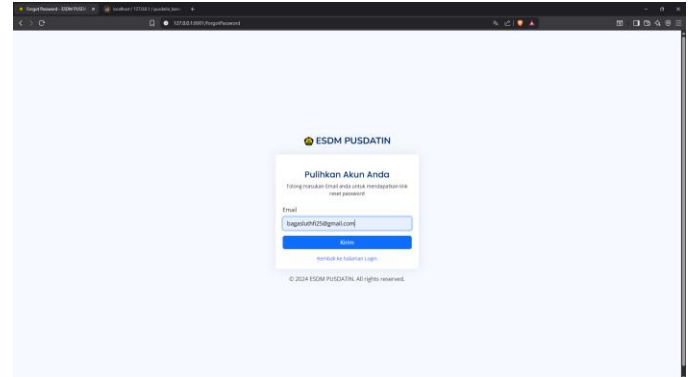
Gambar 5. Halaman Login

Fitur *login* berfungsi untuk mengautentikasi pengguna sebelum mengakses sistem. Proses dimulai dari *input email* dan *password*, kemudian sistem memverifikasi data dengan mencocokkannya ke *database*. Jika valid, sistem menghasilkan

JSON Web Token (JWT) yang dikirimkan sebagai tanda autentikasi dan mengarahkan pengguna ke halaman utama.

Endpoint login diakses melalui *method POST /api/login* dengan parameter *email* dan *password*. Jika berhasil, sistem mengembalikan response berupa token dan data pengguna, sedangkan jika gagal akan menampilkan pesan error seperti “*Email tidak terdaftar*” atau “*Password salah*”.

2. Lupa Password



Gambar 6. Halaman Lupa Password

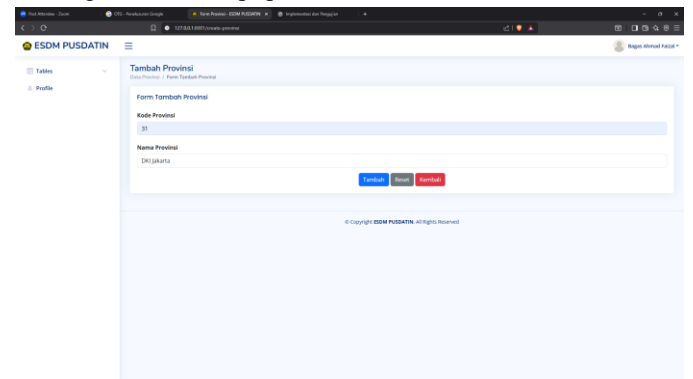
Fitur lupa *password* digunakan ketika pengguna tidak dapat mengakses akun karena melupakan kata sandi. Proses dimulai dari pengisian form dengan *email* terdaftar. Jika valid, sistem mengirimkan link atau token reset *password* ke *email* pengguna.

Pengguna kemudian membuka link tersebut untuk membuat *password* baru dan mengonfirmasikannya. Jika berhasil, sistem menyimpan *password* baru di *database* serta menampilkan pesan “*Password berhasil direset*”. Sebaliknya, jika *email* tidak terdaftar atau token tidak valid, sistem akan menampilkan pesan *error*.

3. CRUD Data Provinsi

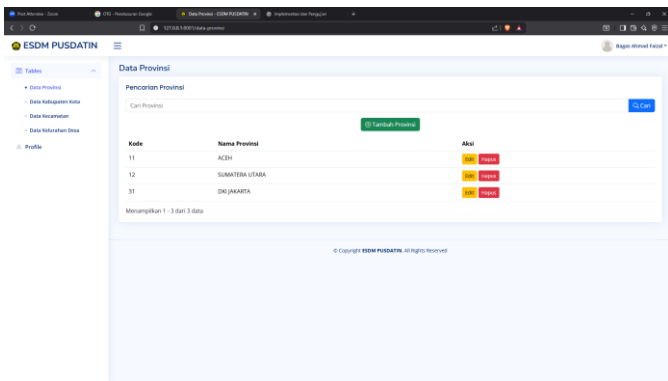
Fitur *CRUD (Create, Read, Update, Delete)* digunakan untuk mengelola data provinsi dalam sistem melalui antarmuka *front-end* dan *endpoint RESTful API*.

a. *Create*: Pengguna menambahkan data provinsi baru dengan mengisi *Kode_provinsi* dan *Nama_provinsi* melalui *endpoint POST /api/provinsi*.



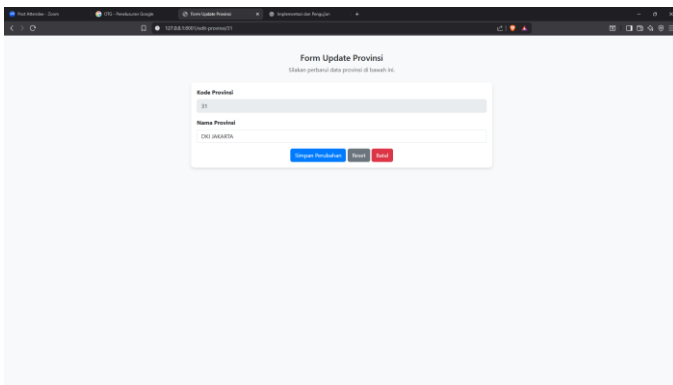
Gambar 7. Form Tambah Data Provinsi

b. *Read*: Sistem menampilkan daftar provinsi dari *database* menggunakan *endpoint GET /api/provinsi* dalam bentuk tabel.



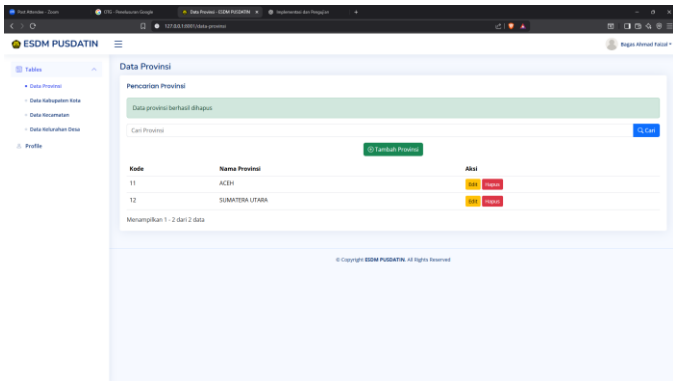
Gambar 8. Tampilan *Read Data* Provinsi

c. *Update*: Pengguna dapat memperbarui Nama provinsi melalui *endpoint* *PUT* /api/provinsi/{id}, sementara kode provinsi tidak dapat diubah.



Gambar 9. Form Update Data Provinsi

d. *Delete*: Pengguna dapat menghapus data provinsi tertentu menggunakan *endpoint* *DELETE* /api/provinsi/{id}.



Gambar 10. Delete Data Provinsi

C. Hasil Pengujian

Hasil pengujian merupakan tahap untuk memastikan bahwa sistem yang telah dikembangkan berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Pengujian dilakukan dengan metode *Black Box Testing*, yaitu dengan memberikan *input* pada sistem dan mengamati *output* yang dihasilkan tanpa melihat kode program. Fokus pengujian mencakup fungsi autentikasi (*login* dan *logout*) serta fitur pengelolaan data (*CRUD*), guna memastikan kedua komponen utama tersebut berfungsi dengan benar, aman, dan sesuai harapan pengguna.

1. Pengujian Fungsi Autentikasi

Pengujian autentikasi dilakukan menggunakan *Black Box Testing* untuk memastikan proses login dan logout berjalan sesuai spesifikasi.

Tabel 1. Hasil Pengujian Fungsi Autentikasi

No	Fungsi	Input	Output yang diharapkan	Status
1	<i>Login</i>	<i>Email & password</i>	Pengguna berhasil login dan diarahkan ke halaman dashboard	Berhasil
2	<i>Login</i>	<i>Email/Password Salah</i>	Muncul pesan error: " <i>Email</i> atau <i>Password</i> salah"	Berhasil
3	<i>Logout</i>	Klik Tombol Logout	Pengguna diarahkan ke halaman login dan sesi dihapus	Berhasil

- Login valid: Sistem berhasil mengautentikasi pengguna, memberikan akses, dan mengarahkan ke halaman utama.
- Login tidak valid: Sistem menolak akses dan menampilkan pesan error "*Email* atau *Password* salah".
- Logout: Sistem berhasil menghapus sesi pengguna dan kembali ke halaman login.

2. Pengujian *CRUD* Provinsi

Pengujian *CRUD* pada entitas Provinsi juga menggunakan *Black Box Testing* untuk memastikan fungsi pengelolaan data berjalan dengan benar.

Tabel 2. Hasil Pengujian *CRUD* Provinsi

No	Fungsi	Input	Output yang diharapkan	Status
1	<i>Create</i>	Kode dan Nama Provinsi "Jawa Barat"	Data provinsi berhasil disimpan dan muncul dalam daftar	Berhasil
2	<i>Read</i>	-	Sistem menampilkan seluruh data provinsi yang tersimpan	Berhasil
3	<i>Update</i>	Ubah nama "Jawa Barat" → "Jabar"	Data provinsi berhasil diperbarui dan nama berubah di daftar	Berhasil
4	<i>Delete</i>	Hapus data "Jabar"	Data provinsi berhasil dihapus dan tidak lagi tampil di daftar	Berhasil

- a. Create: Data provinsi baru berhasil disimpan ke database dan muncul di daftar.
- b. Read: Sistem menampilkan seluruh data provinsi dengan lengkap dan akurat.
- c. Update: Nama provinsi berhasil diperbarui dan langsung tampil di daftar.
- d. Delete: Data provinsi berhasil dihapus dan tidak ditampilkan lagi.

V. KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi pengelolaan data wilayah berbasis *RESTful API* menggunakan *framework Laravel 11* dengan dukungan autentikasi *JSON Web Token (JWT)*. Sistem yang dibangun mampu menyediakan fungsi utama berupa *CRUD* pada entitas provinsi, kabupaten/kota, kecamatan, dan kelurahan/desa secara terstruktur, efisien, serta aman. Hasil pengujian menggunakan metode *Black Box Testing* menunjukkan bahwa seluruh fitur, mulai dari autentikasi hingga pengelolaan data wilayah, berjalan sesuai spesifikasi dan menghasilkan keluaran yang benar. Dengan adanya aplikasi ini, proses integrasi dan pengelolaan data wilayah di Kementerian ESDM menjadi lebih mudah, cepat, fleksibel, serta mendukung percepatan transformasi digital pemerintahan.

DAFTAR PUSTAKA

- [1] A. Karim, B. Bangun, Kusmanto, I. Purnama, S. Z. Hararap, D. Irmayani, M. Nasution, M. H. Munandar, Rahmadani and I. R. Munrhe. "Pengantar Teknologi Informasi," *Yayasan Labuhanbatu Berbagi Gumilang*, Ed. 1, 2020
- [2] Rivan Nurdin and Rangga Sanjaya, "Perancangan REST API Untuk Pengembangan Sistem SPBE di Desa Tembok," *eProsiding Teknik Informatika*, vol. 5, no. 1, pp. 177–185, 2024.
- [3] H. A. Pangestu, D. Kurniadi, and Y. Septiana, "Aplikasi Pengelolaan Data Pegawai Berbasis REST API untuk Transfer Data Real Time dengan Framework Codeigniter," *Jurnal Algoritma*, vol. 19, no. 1, pp. 304–313, May 2022, doi: 10.33364/algoritma/v.19-1.1090.
- [4] R. Simbulan and J. Aryanto, "Implementasi REST API Web Services pada Aplikasi Sumber Daya Manusia," *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 5, no. 1, pp. 552–560, Jan. 2024, doi: 10.35870/jimik.v5i1.511.
- [5] S. B. Farchani, N. Hermanto, and B. A. Kusuma, "Implementasi Rest Api Dalam Pengembangan Backend Inventory Peminjaman," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 2, pp. 1404–1413, Mar. 2025, doi: 10.29100/jipi.v10i2.6249.
- [6] S. Sotar, A. Nolla, and A. R. Aswin, "Sistem Informasi Pelayanan Kunjungan pada Kejaksaan Negeri Solok," *remik*, vol. 9, no. 3, pp. 771–779, Aug. 2025, doi: 10.33395/remik.v9i3.14830.
- [7] S. Supiyandi, M. Zen, C. Rizal, and M. Eka, "Perancangan Sistem Informasi Desa Tomuan Holbung Menggunakan Metode Waterfall," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 2, p. 274, Apr. 2022, doi: 10.30865/jurikom.v9i2.3986.
- [8] S. Ramdany, "Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web," *Journal of Industrial and Engineering System*, vol. 5, no. 1, July 2024, doi: 10.31599/2e9afp31.
- [9] Z. Tuasamu et al., "Analisis Sistem Informasi Akuntansi Siklus Pendapatan Menggunakan DFD dan Flowchart Pada Bisnis Porobico," *Jurnal Bisnis dan Manajemen (JURBISMAN)*, vol. 1, no. 2, pp. 495–510, May 2023, doi: 10.61930/jurbisman.v1i2.181.
- [10] Siska Narulita, Ahmad Nugroho, and M. Zakki Abdillah, "Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS)," *Bridge : Jurnal publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 3, pp. 244–256, Aug. 2024, doi: 10.62951/bridge.v2i3.174.