

# Implementasi Backend pada Website Pendukung Keputusan pada Investasi Lahan di Jawa Barat

1<sup>st</sup> Muhammad Daffa' Saifullah

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

darkknight@student.telkomuniversity.ac.id

2<sup>nd</sup> Casi Setianingsih

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3<sup>rd</sup> Ashri Dinimaharawati

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

ashridini@telkomuniversity.ac.id

**Abstrak** — Penelitian ini mengembangkan dan mengimplementasikan sistem backend untuk website pendukung keputusan investasi lahan di Jawa Barat. Dengan memanfaatkan teknologi seperti Flask untuk server aplikasi dan RailwayApp sebagai platform deployment, sistem ini dirancang untuk menyajikan data real-time yang akurat dan mudah diakses oleh pengguna. Melalui integrasi data penting seperti harga tanah, tren pasar, dan analisis risiko, sistem ini membantu pengguna membuat keputusan investasi yang lebih bijak. Pengujian kinerja melalui stress test menunjukkan bahwa sistem backend mampu menangani beban kerja tinggi dengan efisiensi yang baik, ditandai dengan respons waktu yang konsisten dan hasil sukses 100% dalam skenario uji dengan 100 dan 500 pengguna. Hasil penelitian ini diharapkan dapat memberikan kontribusi positif dalam bidang investasi lahan dan menjadi referensi bagi pengembangan sistem serupa di masa mendatang.

**Kata Kunci** — Backend, Investasi Lahan, Flask, RailwayApp, Stress Test

## I. PENDAHULUAN

Investasi lahan merupakan salah satu bentuk investasi yang menjanjikan, terutama di daerah-daerah yang sedang berkembang seperti Jawa Barat [1]. Pertumbuhan ekonomi yang pesat di kawasan ini telah mendorong peningkatan permintaan terhadap lahan untuk berbagai keperluan, baik komersial, industri, maupun perumahan [2]. Namun, keputusan untuk berinvestasi dalam lahan bukanlah hal yang sederhana. Banyak faktor yang harus dipertimbangkan, seperti lokasi, harga pasar, potensi perkembangan, hingga regulasi pemerintah yang berlaku. Oleh karena itu, diperlukan suatu sistem yang dapat membantu calon investor dalam membuat keputusan yang tepat dan berdasarkan data yang akurat.

Seiring dengan perkembangan teknologi informasi, website pendukung keputusan telah menjadi alat yang semakin penting dalam berbagai bidang, termasuk investasi lahan [3]. Sistem backend dari website ini memiliki peran krusial dalam mengelola, mengolah, dan menyajikan data yang relevan bagi pengguna. Backend yang efisien dan dapat diandalkan memungkinkan pengambilan keputusan yang lebih cepat dan akurat, sehingga dapat meningkatkan peluang keberhasilan investasi.

Penelitian ini berfokus pada pengembangan sistem backend untuk website pendukung keputusan investasi lahan di Jawa Barat. Tujuan utama dari pengembangan ini adalah untuk menciptakan sebuah platform yang dapat menyajikan data dan informasi secara real-time, akurat, dan mudah diakses oleh pengguna. Selain itu, sistem ini juga diharapkan mampu mengintegrasikan berbagai data penting, seperti harga tanah, tren pasar, dan analisis risiko, sehingga pengguna dapat membuat keputusan investasi yang lebih bijak.

Dalam laporan ini, akan dibahas proses pengembangan sistem backend tersebut, mulai dari perancangan arsitektur, pemilihan teknologi yang digunakan, hingga implementasi dan pengujian sistem. Diharapkan hasil dari penelitian ini dapat memberikan kontribusi positif dalam bidang investasi lahan, khususnya di Jawa Barat, serta menjadi referensi bagi pengembangan sistem serupa di masa mendatang..

## II. KAJIAN TEORI

### A. Pengembangan Sistem Berbasis Web

Sistem berbasis web telah berkembang pesat dan menjadi platform penting dalam berbagai bidang, termasuk dalam investasi lahan. Pengembangan sistem backend merupakan aspek krusial dalam pengelolaan data dan penyediaan layanan yang efisien kepada pengguna [4].

### B. Flask sebagai Framework Web

Flask adalah framework web berbasis Python yang menawarkan fleksibilitas dan modularitas, memungkinkan pengembangan aplikasi web yang ringan dan efisien. Flask juga mendukung integrasi dengan berbagai teknologi lainnya yang dibutuhkan dalam pengembangan aplikasi web [5].

### C. Railway

Platform as a Service (PaaS) seperti RailwayApp menyediakan infrastruktur yang dibutuhkan untuk menjalankan aplikasi web tanpa perlu mengelola perangkat keras secara langsung. RailwayApp memfasilitasi deployment yang mudah dan mendukung proses CI/CD [6].

#### D. Github

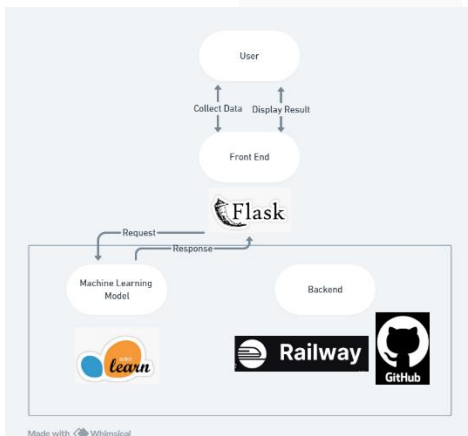
GitHub adalah platform pengelolaan kode sumber yang memfasilitasi kolaborasi pengembangan melalui kontrol versi Git. Integrasi GitHub dengan RailwayApp memungkinkan proses deployment yang otomatis, sehingga mempermudah pengelolaan aplikasi.

### III. RESEARCH METHODS

#### A. Perancangan Arsitektur Sistem

Perancangan arsitektur sistem backend dilakukan dengan mempertimbangkan kebutuhan fungsional dan non-fungsional yang diidentifikasi pada tahap awal pengembangan. Pada tahap ini, arsitektur yang dikembangkan menggunakan pendekatan berbasis komponen, di mana setiap komponen memiliki peran spesifik yang mendukung keseluruhan fungsi sistem. Komponen utama terdiri dari antarmuka pengguna (frontend), server aplikasi yang dikelola oleh Flask, model machine learning yang diimplementasikan dengan scikit-learn, dan platform deployment yang memanfaatkan Railway serta manajemen kode menggunakan GitHub. Komunikasi antara komponen frontend dan backend dilakukan melalui API (Application Programming Interface) yang dibangun dengan Flask, yang memungkinkan data dikirimkan dari frontend ke backend dan hasil diproses kemudian dikirimkan kembali untuk ditampilkan kepada pengguna..

Rancangan pada sistem backend website pendukung keputusan pada investasi lahan di Jawa Barat seperti gambar berikut:



GAMBAR 1  
Rancangan Sistem

#### B. Pengembangan dan Implementasi Sistem

Tahap pengembangan melibatkan coding dan integrasi komponen backend, mulai dari pengaturan server Flask hingga integrasi dengan model machine learning dan platform Railway. Model machine learning dilatih menggunakan dataset terkait harga lahan, tren pasar, dan faktor risiko, kemudian diintegrasikan ke dalam sistem Flask untuk memproses data real-time yang dikirimkan dari frontend.

#### C. Upload Github

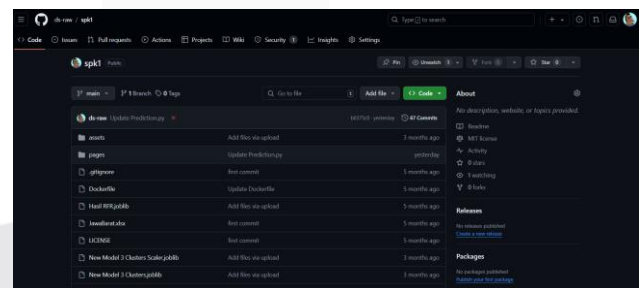
Setelah pengembangan sistem backend selesai, langkah berikutnya adalah mengunggah kode sumber ke platform GitHub. GitHub adalah platform pengelolaan versi yang berbasis Git dan digunakan untuk menyimpan, melacak

perubahan, dan mengelola kolaborasi dalam pengembangan perangkat lunak.

Proses unggah ke GitHub dimulai dengan membuat repository baru untuk proyek ini. Repository tersebut akan berfungsi sebagai wadah penyimpanan kode yang telah dikembangkan. Langkah-langkah untuk mengunggah kode ke GitHub adalah sebagai berikut:

1. Inisialisasi Repository Git: Pertama, repository lokal diinisialisasi dengan perintah `git init` di dalam direktori proyek. Ini akan membuat repository Git lokal yang melacak perubahan pada file di direktori tersebut.
2. Menambahkan File: Setelah repository diinisialisasi, semua file proyek yang ingin diunggah ke GitHub ditambahkan ke repository lokal menggunakan perintah `git add ..` Tanda titik (`.`) menunjukkan bahwa semua file dalam direktori tersebut akan ditambahkan.
3. Commit Perubahan: Setelah file ditambahkan, perubahan tersebut dicatat dalam repository dengan perintah `git commit -m "Initial commit"`. Commit ini berfungsi sebagai titik penyimpanan perubahan, yang dapat diakses kembali jika diperlukan.
4. Menghubungkan ke Repository GitHub: Selanjutnya, repository lokal dihubungkan dengan repository yang ada di GitHub menggunakan perintah `git remote add origin <URL_REPO_GITHUB>`. URL ini adalah alamat repository GitHub yang telah dibuat.
5. Push ke GitHub: Terakhir, kode yang telah di-commit diunggah ke GitHub dengan perintah `git push -u origin main`. Ini akan mengirimkan semua perubahan dari repository lokal ke repository GitHub, menjadikannya tersedia secara publik atau pribadi sesuai pengaturan repository.

Berikut adalah gambar hasil unggah ke GitHub:



GAMBAR 2  
Unggah ke Github

Proses ini memastikan bahwa kode sumber sistem backend tersimpan dengan aman di GitHub dan dapat diakses atau dikembangkan lebih lanjut di masa mendatang.

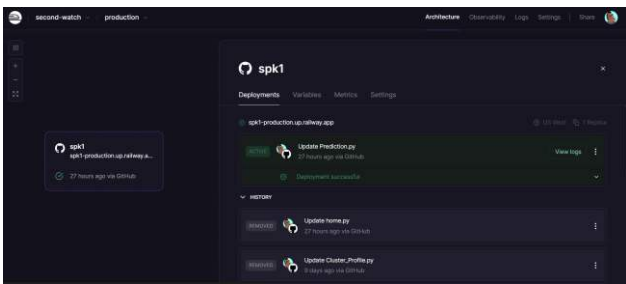
#### D. Deploy

Setelah kode sumber diunggah ke GitHub, langkah selanjutnya adalah melakukan deployment aplikasi ke platform RailwayApp. RailwayApp adalah *Platform as a Service (PaaS)* yang memungkinkan pengembang untuk menyebarkan dan menjalankan aplikasi tanpa perlu mengelola infrastruktur fisik.

Proses deployment melibatkan langkah-langkah berikut:

1. Integrasi dengan GitHub: RailwayApp memiliki fitur integrasi yang mudah dengan GitHub. Untuk memulai, pengembang hanya perlu menghubungkan akun RailwayApp mereka dengan akun GitHub, dan memilih

- repository yang berisi kode sumber dari aplikasi yang akan di-deploy.
- Membuat Proyek Baru di RailwayApp: Setelah integrasi dengan GitHub berhasil, pengembang dapat membuat proyek baru di RailwayApp. RailwayApp akan secara otomatis mendeteksi *repository* yang telah dipilih dan memulai proses *deployment*.
  - Konfigurasi *Environment Variables*: Beberapa aplikasi memerlukan variabel lingkungan (environment variables) seperti kunci API atau konfigurasi database. Variabel-variabel ini dapat dikonfigurasi di RailwayApp sebelum proses *deployment* dimulai.
  - Proses *Build* dan *Deployment*: RailwayApp akan secara otomatis membangun (*build*) aplikasi berdasarkan konfigurasi yang terdapat di *repository* GitHub, seperti *file requirements.txt* untuk dependensi Python. Setelah proses *build* selesai, aplikasi akan di-deploy dan dapat diakses melalui URL yang disediakan oleh RailwayApp.
  - Verifikasi *Deployment*: Setelah deployment selesai, penting untuk memverifikasi bahwa aplikasi berjalan dengan benar di lingkungan produksi. Pengujian dapat dilakukan dengan mengakses URL aplikasi dan memeriksa setiap fungsi yang telah diimplementasikan.
- Berikut adalah gambar hasil *deployment* menggunakan RailwayApp:



GAMBAR 3 Hasil Deployment menggunakan RailwayApp

#### IV. HASIL DAN PEMBAHASAN

##### A. Spesifikasi PaaS

Sistem *backend* yang dikembangkan dan diimplementasikan pada *platform* RailwayApp memiliki spesifikasi berikut:

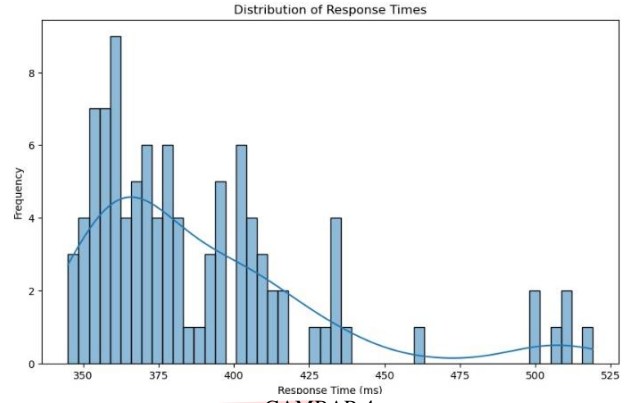
TABEL 1 Spesifikasi PaaS Railway

Spesifikasi	Informasi Spesifikasi
RAM	8 GB
vCPU	8 vCPU
Region	US-West Region
Link Service	https://spk1-production.up.railway.app/

Sistem *backend* yang dikembangkan telah diimplementasikan di *platform* RailwayApp dengan spesifikasi yang dirancang untuk memastikan performa optimal dan keandalan tinggi. Sistem ini dilengkapi dengan **8 GB RAM** dan **8 vCPU** yang memungkinkan penanganan beban kerja tinggi secara efisien, terutama dalam pemrosesan data *real-time* dan eksekusi permintaan dari pengguna.

##### B. Kinerja Sistem Backend

###### 1. Stress Test 100 User

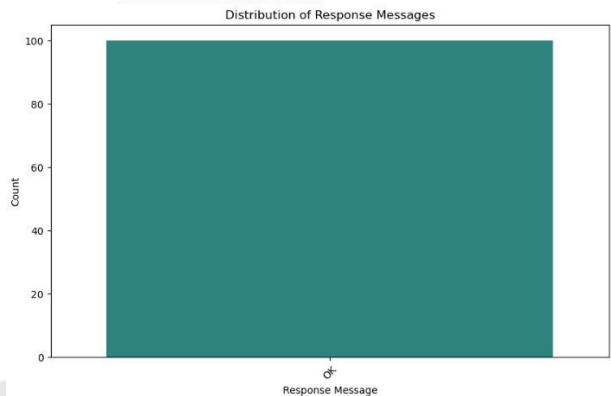


GAMBAR 4 Distribusi Waktu Respon 100 User

```
# Calculate the success rate
success_rate = df['success'].mean() * 100
print(f"Success Rate: {success_rate:.2f}%")
Success Rate: 100.00%
```

GAMBAR 5 Hasil Success Rate 100 User

Berdasarkan *Stress Test* yang telah dilakukan pada 100 user, diketahui penilaian dengan 100 user dapat dikatakan sukses 100 %.

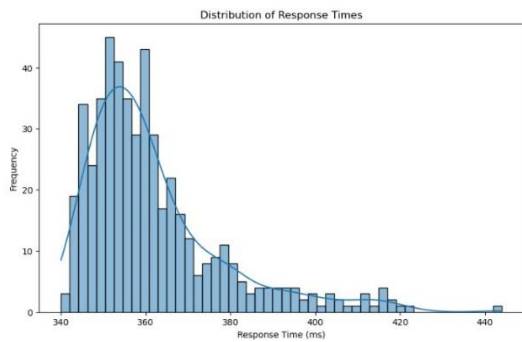


GAMBAR 6 Response Message 100 User

Dengan analisis sebagai berikut:

- Distribusi waktu respons menunjukkan bahwa sebagian besar permintaan memiliki waktu respons antara 350 ms dan 425 ms.
- Puncak histogram terletak di sekitar 350 ms hingga 375 ms, menunjukkan bahwa ini adalah rentang waktu respons yang paling sering terjadi.
- Response Message* seluruhnya OK.

###### 2. Stress Test 500 User



GAMBAR 7  
Distribusi Waktu 500 User

```
# Calculate the success rate
success_rate = df['success'].mean() * 100
print(f"Success Rate: {success_rate:.2f}%")
```

Success Rate: 100.00%

GAMBAR 8  
Hasil Success Rate 500 User

Berdasarkan *Stress Test* yang telah dilakukan pada 500 user, diketahui penilaian dengan 500 user dapat dikatakan sukses 100 %.



GAMBAR 9  
Response Message 500 User

Dengan analisis sebagai berikut :

1. Grafik ini menunjukkan bahwa sebagian besar permintaan memiliki waktu respons antara 340 ms dan 380 ms.
2. Puncak frekuensi berada di sekitar 350 ms hingga 360 ms, menunjukkan bahwa ini adalah rentang waktu respons yang paling sering terjadi.
3. *Response Message* seluruhnya OK.

## V. KESIMPULAN

Penelitian ini berhasil mengembangkan dan mengimplementasikan sistem *backend* pada platform RailwayApp yang mampu menangani permintaan pengguna secara efisien dengan spesifikasi yang telah dirancang. Hasil pengujian menunjukkan bahwa sistem backend yang dikembangkan memiliki performa yang optimal, ditunjukkan oleh hasil *stress test* yang konsisten dan memuaskan dalam mendukung investasi lahan kosong di Jawa Barat.

## REFERENSI

- [1] S. Asnawi, "ANALISIS KESESUAIAN LAHAN UNTUK PENGEMBANGAN KAWASAN INDUSTRI," *DINAMIK*, vol. 29, no. 1, 2024.
- [2] S. Alfiani and P. R. Arum, "Pemodelan Pertumbuhan Ekonomi di Jawa Barat Menggunakan Metode Geographically Weighted Panel Regression," *Jurnal Statistika*, vol. 15, no. 2, pp. 219-227, 2022.
- [3] I. Darma Paramartha and I. Kadek Arta Wiguna, "Perancangan Website Cerdas Pemilihan Kampus dengan Semantic Web dan Grabbing Data," *TIERS Information Technology Journal*, vol. 1, no. 1, pp. 12-17, 2020.
- [4] T. Sriwahyuni and e. al, "Pengembangan Sistem Informasi Manajemen Pariwisata Berbasis Web," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 12, no. 1, pp. 92-99, 2019.
- [5] S. Chikara, "pwskills," PW Skills, 14 August 2024. [Online]. Available: <https://pwskills.com/blog/flask-web-framework/>. [Accessed 14 August 2024].
- [6] heptabit, "DevOps Practices - Platform as a Service (PaaS)," Heptabit GmbH, 29 May 2020. [Online]. Available: <https://www.heptabit.at/blog/devops-practices-platform-as-a-service-paas>. [Accessed 14 August 2024].