



## PENGEMBANGAN *BACKEND* APLIKASI GEOPROPERTY DENGAN GOLANG DI PT NERDVANA SOLUSI TEKNOLOGI

Muhammad Asnur Ramdani<sup>1</sup>, Tiffany Nabarian<sup>2</sup>, Reza Maulana<sup>3</sup>

<sup>1,2,3</sup> Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640

muha21035ti@student.nurulfikri.ac.id, nabarian@nurulfikri.ac.id, rezamaulana@nurulfikri.ac.id

### Abstract

*This research discusses the backend design of the Golang-based GeoProperty application at PT Nerdvana Solusi Teknologi, which is designed to improve the property search experience with the integration of Geographic Information System (GIS) technology. This research focuses on developing the backend using the Agile Scrum method to ensure optimal functionality in managing spatial and property data. The backend supports key features such as interactive map-based property search, polygon input for search area boundaries, and Point of Interest (POI) visualization around the property. Testing was conducted using the black box testing method to evaluate API functionality. The results show that the backend system can meet user needs with reliable performance, provide accurate visualization of property locations, and comprehensive information that supports prospective buyers' decisions.*

**Keywords:** API, Backend, Design, Geographic Information System, Golang

### Abstrak

Penelitian ini membahas tentang rancang bangun *backend* aplikasi GeoProperty berbasis Golang di PT Nerdvana Solusi Teknologi, yang dirancang untuk meningkatkan pengalaman pencarian properti dengan integrasi teknologi *Geographic Information System* (GIS). Penelitian ini berfokus pada pengembangan *backend* menggunakan metode *Agile Scrum* untuk memastikan fungsionalitas optimal dalam mengelola data spasial dan properti. *Backend* ini mendukung fitur utama seperti pencarian properti berbasis peta interaktif, *input* poligon untuk batas area pencarian, dan visualisasi *Point of Interest* (POI) di sekitar properti. Pengujian dilakukan menggunakan metode *black box testing* untuk mengevaluasi fungsionalitas API. Hasilnya menunjukkan bahwa sistem *backend* dapat memenuhi kebutuhan pengguna dengan performa yang andal, memberikan visualisasi lokasi properti yang akurat, serta informasi komprehensif yang mendukung keputusan calon pembeli.

**Kata kunci:** API, Backend, Golang, Rancang Bangun, Sistem Informasi Geografis

### 1. PENDAHULUAN

Teknologi informasi memiliki peran penting dalam industri properti, terutama melalui media digital seperti situs web[1]. Promosi properti secara *online* menawarkan banyak keunggulan, termasuk jangkauan *audiens* yang lebih luas dan informasi yang lebih detail. Data menunjukkan bahwa 47% calon pembeli lebih memilih mencari iklan properti secara *online* sebelum melakukan survei langsung[2].

Namun, sebagian besar situs properti di Indonesia masih terbatas pada penyajian data dasar, seperti harga dan ukuran properti, tanpa menyajikan informasi tentang fasilitas publik, aksesibilitas, atau potensi pengembangan lokasi. Hal

ini menyulitkan calon pembeli untuk memahami kondisi properti secara menyeluruh. Penelitian menunjukkan bahwa penggunaan peta interaktif dapat meningkatkan minat pembeli hingga 18% dibandingkan dengan situs tanpa fitur ini[3]. Sebagai solusi, PT. Nerdvana Solusi Teknologi mengembangkan aplikasi GeoProperty, yang memanfaatkan teknologi *Geographic Information System* (GIS) untuk menghadirkan visualisasi lokasi yang lebih lengkap dan interaktif. GIS memiliki potensi untuk meningkatkan konversi penjualan hingga 25%, mengurangi biaya pemasaran hingga 40%, dan meningkatkan kepuasan pelanggan hingga 30%[4].

*Backend* aplikasi ini dikembangkan menggunakan Golang, yang dipilih karena kemampuannya dalam menangani data spasial yang kompleks secara efisien. Dengan pendekatan pengembangan berbasis *Agile Scrum*, GeoProperty diharapkan dapat meningkatkan pengalaman pengguna dalam pencarian properti.

Pengembangan *backend* aplikasi GeoProperty bertujuan untuk mengintegrasikan teknologi *Geographic Information System (GIS)* sebagai fitur unggulan dalam menyediakan visualisasi lokasi properti yang interaktif dan informatif. Dengan memanfaatkan teknologi GIS, aplikasi ini dapat menampilkan data spasial yang detail, seperti batas wilayah, akses ke fasilitas publik, dan *Point of Interest (POI)* di sekitar lokasi properti. Fitur ini dirancang untuk membantu calon pembeli memahami kondisi lokasi secara lebih komprehensif, sehingga dapat mendukung mereka dalam membuat keputusan pembelian yang lebih tepat.

Penelitian ini difokuskan pada pengembangan *backend* GeoProperty menggunakan Golang, dengan ruang lingkup yang dibatasi pada pengolahan data spasial berbasis *dummy* data dari Kota Depok. Fitur unggulan yang dikembangkan meliputi pencarian properti berbasis peta interaktif dan visualisasi POI. *Backend* ini diharapkan dapat memberikan pengalaman pencarian properti yang lebih efektif dan inovatif, sekaligus membedakan GeoProperty dari platform serupa di pasar properti *online*.

#### Website Properti

*Website* properti merupakan platform digital yang menyediakan berbagai informasi, layanan, dan transaksi seputar properti, seperti rumah, apartemen, tanah, hingga bangunan komersial. Platform ini sering dilengkapi dengan sistem transaksi *online* yang mencakup fitur penawaran harga, negosiasi, hingga pembayaran, sehingga mempermudah proses jual beli tanpa memerlukan pertemuan langsung[5].

#### Data Spasial

Data spasial adalah jenis data yang berisi informasi tentang lokasi di permukaan bumi, dengan setiap atributnya memiliki referensi geografis (*georeference*). Artinya, data ini mengaitkan berbagai informasi atribut dengan unit spasial tertentu. Data spasial dimanfaatkan di berbagai bidang, seperti perencanaan tata kota, pengelolaan sumber daya alam, pemetaan risiko bencana, hingga analisis pasar. Misalnya, dalam analisis penjualan, data spasial dapat membantu menentukan lokasi dengan potensi pasar terbaik berdasarkan informasi geografis[6].

#### GeoJSON

GeoJSON adalah format standar berbasis JSON (*JavaScript Object Notation*) yang dirancang untuk merepresentasikan data *geospasial* secara terstruktur dan mudah diproses. Format ini sering digunakan dalam aplikasi web untuk visualisasi peta interaktif, analisis data spasial, dan integrasi

data geografis. Selain menyimpan informasi geometri, GeoJSON juga mendukung atribut tambahan seperti nama, ukuran, atau jenis properti, sehingga cocok untuk berbagai kebutuhan data spasial dinamis[7].

#### PostGIS

PostGIS adalah sebuah ekstensi untuk PostgreSQL yang memungkinkan pengguna untuk mengelola dan menganalisis data spasial. Dengan PostGIS, pengguna dapat menyimpan, mengelola, dan melakukan berbagai analisis terhadap data spasial di dalam *database* PostgreSQL. Beberapa fitur yang disediakan mencakup operasi geometri seperti perhitungan jarak, pemeriksaan batas, serta pemrosesan data spasial lainnya[8].

#### Point of Interest

*Point of Interest (POI)* merujuk pada lokasi-lokasi spesifik yang memiliki nilai menarik atau berguna bagi seseorang. Istilah ini mencakup berbagai jenis tempat, seperti restoran, rumah sakit, toko, taman, objek wisata, stasiun transportasi, dan fasilitas publik lainnya. POI tidak hanya penting untuk aktivitas sehari-hari, tetapi juga sering digunakan dalam aplikasi navigasi, peta interaktif, dan analisis data spasial[9].

#### Agile Scrum

*Agile Scrum* adalah sebuah kerangka kerja yang digunakan untuk mengelola proyek, khususnya dalam pengembangan perangkat lunak, dengan pendekatan yang adaptif dan responsif. Dalam *Agile Scrum*, proyek dibagi menjadi periode kerja singkat yang disebut *sprint*, di mana tim bekerja bersama untuk mencapai tujuan yang telah ditentukan. Pendekatan ini tidak hanya menekankan pada pencapaian tujuan di setiap *sprint*, tetapi juga pada perbaikan berkelanjutan dalam proses dan hasil yang dihasilkan[10].

#### Black Box Testing

*Black box testing* adalah suatu metode pengujian perangkat lunak yang tidak melibatkan pengetahuan tentang struktur internal atau kode sumber perangkat lunak tersebut. Dalam *black box testing*, pengujian dilakukan dengan memasukkan *input* dan memeriksa hasil *output* tanpa memahami bagaimana perangkat lunak tersebut bekerja di dalamnya[11].

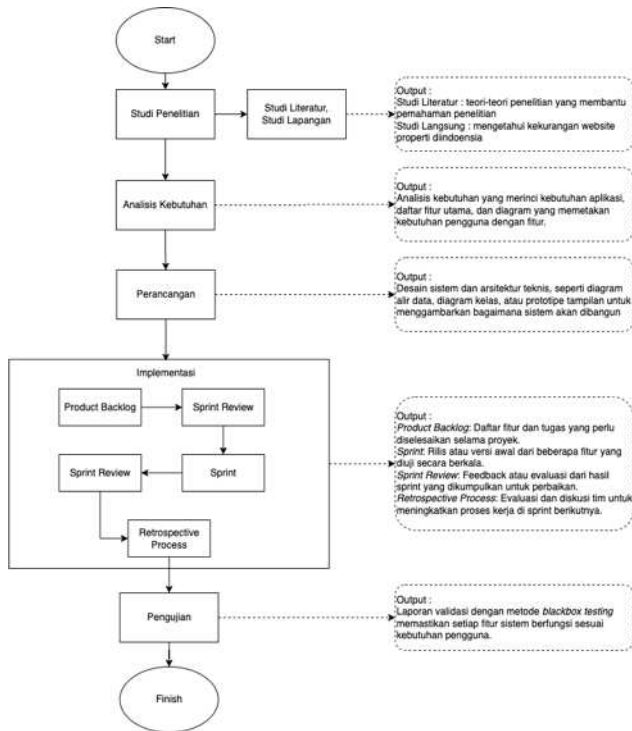
## 2. METODE PENELITIAN

Jenis penelitian yang digunakan dalam penelitian ini adalah *Research and Development (R&D)* yang dimana R&D bertujuan untuk mengembangkan serta menghasilkan produk baru yang dapat memecahkan masalah spesifik dalam suatu bidang[12].

### 2.1. Tahapan Penelitian

Pada Gambar 1 ini menunjukkan tahapan pengembangan sistem GeoProperty, mulai dari studi awal hingga pengujian

akhir. Setiap tahap dirancang secara sistematis untuk memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan pengguna, dimulai dari studi literatur dan lapangan, analisis kebutuhan, perancangan, implementasi dengan pendekatan iteratif, hingga pengujian untuk validasi akhir.



Gambar 1. Tahapan Penelitian

### 2.1.1. Studi Penelitian

Tahap ini melibatkan studi literatur untuk memahami teori, teknologi, dan metode yang mendukung pengembangan *backend* GeoProperty berbasis Golang. Studi mencakup GIS, arsitektur *backend*, dan penerapan *framework Agile Scrum*, menghasilkan laporan yang merangkum konsep, arsitektur, dan metode relevan sebagai dasar pengembangan aplikasi.

### 2.1.2. Analisis Kebutuhan

Tahap ini mengidentifikasi kebutuhan aplikasi, termasuk fitur, performa, dan integrasi data *geospasial*. *Output*-nya berupa dokumen kebutuhan, daftar fitur utama, dan diagram yang menghubungkan kebutuhan pengguna dengan fitur.

### 2.1.3. Perancangan

Tahap ini merancang arsitektur *backend*, mencakup struktur *database*, API, dan modul. *Output*-nya berupa diagram arsitektur yang menggambarkan interaksi *server*, *database*, dan klien, skema *database geospasial*, spesifikasi API dengan detail metode dan format JSON, serta diagram alir data dari *input* hingga *output*.

### 2.1.4. Implementasi (Agile Scrum)

Implementasi dilakukan melalui pendekatan *Agile Scrum* yang melibatkan beberapa aktivitas utama:

#### a. Product Backlog

*Product backlog* berisi daftar fitur, tugas, dan perubahan yang diusulkan untuk pengembangan *backend* GeoProperty. Setiap *item*, berupa *user story*, mendeskripsikan kebutuhan atau perbaikan, diprioritaskan berdasarkan urgensi proyek. *Output* tahap ini adalah daftar *user stories* yang mencakup fitur-fitur utama aplikasi GeoProperty.

#### b. Sprint Planning

*Sprint planning* adalah proses merencanakan siklus kerja dua minggu, di mana tim memilih *item* dari *product backlog* berdasarkan prioritas dan estimasi waktu. Tim menentukan kapasitas kerja untuk memastikan tugas dapat diselesaikan tepat waktu. *Output*-nya adalah dokumen berisi daftar *user stories* terpilih dan alokasi tugas.

#### c. Sprint

*Sprint* adalah periode kerja intensif untuk menyelesaikan tugas dari *sprint planning*, dengan 6 *sprint* dilaksanakan dari September hingga November. Tim mengembangkan API, mengelola data spasial, mengintegrasikan *database PostgreSQL*, dan menguji setiap komponen. Hasilnya berupa progres mingguan, fitur yang selesai, serta modul API dan integrasi sesuai spesifikasi.

#### d. Sprint Review

*Sprint review* dilakukan di akhir setiap *sprint* untuk mempresentasikan hasil kerja kepada pemangku kepentingan. Tim mendemonstrasikan fitur yang selesai, mengevaluasi kesesuaian dengan harapan, dan menerima umpan balik untuk perbaikan atau *sprint* berikutnya. *Output*-nya adalah laporan *sprint review* yang mencakup fitur selesai, umpan balik, dan demo fitur.

#### e. Retrospective Process

Setelah *sprint review*, tim melakukan retrospektif untuk mengevaluasi proses, mengidentifikasi perbaikan, dan meningkatkan efisiensi. Hasilnya berupa dokumentasi evaluasi yang mencakup hal positif, kendala, dan perbaikan.

### 2.1.5 Pengujian

Pengujian dilakukan dengan metode *black box testing*, yang fokus pada fungsionalitas sistem tanpa melihat kode. Tujuannya untuk memastikan fitur yang dikembangkan berfungsi sesuai kebutuhan. Setiap modul diuji berdasarkan *input* dan *output* yang diharapkan. Hasilnya berupa laporan yang merangkum skenario pengujian, *bug* yang ditemukan, status perbaikan, dan validasi pemenuhan kebutuhan fungsional sesuai spesifikasi.

## 2.2. Metode Pengumpulan Data

Penelitian ini menggunakan tiga metode pengumpulan data: observasi langsung, dokumentasi, dan umpan balik dari pengguna serta tim pengembang.

### 2.2.1. Observasi Langsung

Peneliti mengamati tim pengembang selama proses pengembangan, mencatat interaksi tim, penerapan *Agile Scrum*, dan penggunaan alat terkait data *geospasial*. Selain itu, peneliti juga mengamati *website* properti di Indonesia untuk membandingkan fitur aplikasi dengan standar industri.

### 2.2.2. Dokumentasi

Data dikumpulkan melalui dokumen proyek seperti analisis kebutuhan, spesifikasi desain, dan catatan pengujian untuk mengevaluasi kesesuaian antara rencana dan hasil.

### 2.2.3. Umpan Balik

Umpan balik dari pengguna dan pengembang diperoleh melalui wawancara atau survei untuk mengidentifikasi kebutuhan pengguna dan mengevaluasi fitur aplikasi.

## 2.3. Metode Pengujian

Penelitian ini menggunakan *Black Box Testing* untuk menguji fungsionalitas *backend* aplikasi GeoProperty, dengan tujuan memastikan sistem berfungsi sesuai ekspektasi pengguna. Pengujian dilakukan dengan memberikan sampel data pada fitur yang tersedia, tanpa memerlukan pemahaman mendalam tentang implementasi teknis. Hasil pengujian akan disajikan dalam bentuk tabel untuk mempermudah evaluasi kinerja sistem dan mengidentifikasi area yang perlu diperbaiki, sehingga aplikasi dapat beroperasi lebih efisien dan memenuhi harapan pengguna.

## 3. HASIL DAN PEMBAHASAN

### 3.1. Analisis dan Perancangan

Analisis dan perancangan ini bertujuan untuk mengidentifikasi kebutuhan fungsional dan non fungsional

dari observasi sistem yang ada, khususnya kompetitor di pasar properti *online*. Berdasarkan analisis, dirancang fitur utama aplikasi GeoProperty dan strategi implementasi yang terstruktur untuk memastikan aplikasi memenuhi kebutuhan pengguna dan memberikan keunggulan dibandingkan kompetitor.

#### 3.1.1. Kompetitor Analisis

Analisis terhadap kompetitor utama seperti Rumah123 dan Lamudi mengungkapkan keunggulan dan kekurangan fitur yang mereka tawarkan. Perbandingan ini, yang dapat dilihat pada Tabel 1, menunjukkan bagaimana GeoProperty dapat unggul dalam pencarian properti dengan memanfaatkan teknologi GIS untuk memberikan pengalaman yang lebih akurat dan interaktif, memungkinkan pengguna memilih properti berdasarkan lokasi geografis yang tepat dan informasi kontekstual terkait.

**Tabel 1.** Hasil Kompetitor Analisis

Fitur	Rumah123	Lamudi	GeoProperty
Pencarian Berdasarkan Lokasi	Tersedia	Tersedia	Tersedia
Pencarian Berdasarkan Gambar <i>Polygon</i> Peta	Tidak Tersedia	Tidak Tersedia	Tersedia (Fitur Unggulan)
Peta Interaktif Properti	Tidak Tersedia	Tidak Tersedia	Tersedia (Fitur Unggulan)
Menampilkan <i>Point of Interest</i> (POI)	Tidak Tersedia	Tidak Tersedia	Tersedia (Fitur Unggulan)

#### 3.1.2. User Stories

Berdasarkan analisis kompetitor, GeoProperty unggul dalam fitur seperti pencarian berdasarkan gambar *polygon* peta, peta interaktif properti, dan tampilan *Point of Interest* (POI) di sekitar properti. Fitur-fitur ini dirancang untuk memenuhi kebutuhan fungsional dan memberikan pengalaman lebih akurat dan interaktif bagi pengguna, baik individu, agen, maupun pengunjung. Tabel 2 menyajikan *user stories* yang menjadi dasar pengembangan aplikasi, menggambarkan apa yang ingin dicapai oleh masing-masing pengguna aplikasi GeoProperty.

**Tabel 2.** User Stories

No	Sebagai	Saya ingin	Sehingga
1	Pengguna Individu, Agen Properti	Mengiklankan properti saya di aplikasi GeoProperty	Saya dapat menjual atau menyewakan properti dengan mudah.
2	Pengguna Individu, Agen Properti	Mengelola beberapa iklan properti sekaligus	Saya dapat menawarkan berbagai properti kepada calon pembeli.
3	Pengunjung ( <i>visitor</i> )	Menghubungi penjual properti secara langsung	Saya dapat memperoleh informasi lebih lanjut tanpa perantara.
4	Pengunjung ( <i>visitor</i> )	Menggunakan peta untuk menggambar area pencarian	Saya dapat menemukan properti di lokasi yang saya tentukan.
5	Pengunjung ( <i>visitor</i> )	Melihat properti beserta titik-titik POI di sekitarnya	Saya dapat memahami lingkungan properti dengan lebih baik.
6	Pengunjung ( <i>visitor</i> )	Mencari properti berdasarkan kategori dan harga	Saya dapat menemukan properti sesuai kebutuhan saya.

No	Sebagai	Saya ingin	Sehingga
7	Pengguna Individu, Agen Properti	Mendaftarkan akun saya dengan cepat	Saya dapat mulai menggunakan aplikasi tanpa hambatan.
8	Pengguna Individu, Agen Properti	Mengelola daftar properti yang saya iklankan	Saya bisa mengubah atau menghapus data iklan saya.
9	Pengunjung ( <i>visitor</i> )	Menggunakan filter untuk mempermudah pencarian properti	Saya dapat menemukan properti yang paling relevan dengan kebutuhan saya.
10	Pengunjung ( <i>visitor</i> )	Melihat peta interaktif properti yang dilengkapi dengan akurasi tinggi	Saya dapat mempercayai informasi lokasi yang disediakan.

### 3.1.3. Sprint Planing

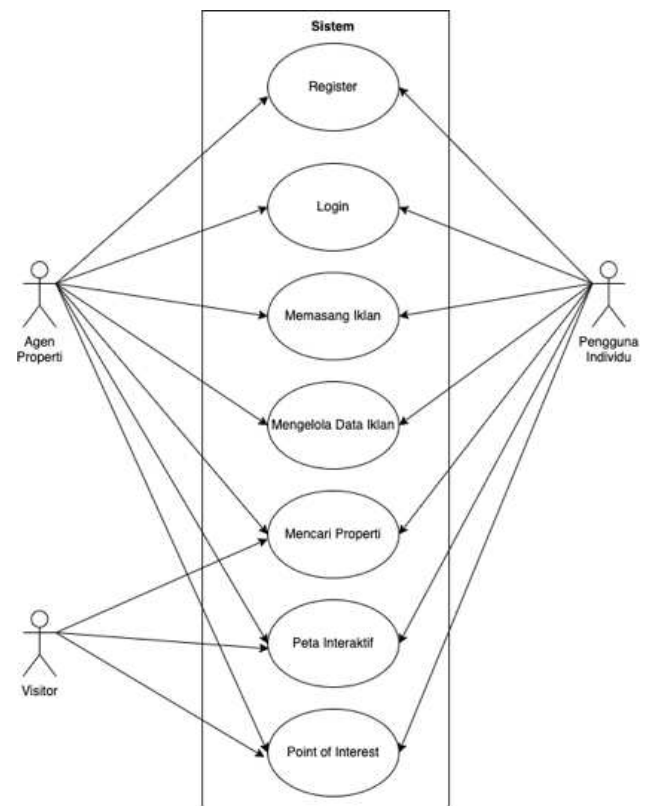
*Sprint Planning* dalam proyek ini disusun berdasarkan *product backlog* yang telah diidentifikasi, dengan fokus pada pengembangan dan pengujian fitur utama *Geoproperty*. Fitur yang dikembangkan meliputi iklan properti, pencarian properti menggunakan peta interaktif, dan penerapan teknologi GIS untuk meningkatkan pengalaman pencarian. Setiap sprint akan mencakup tugas dengan estimasi *story point* sesuai prioritas untuk memastikan pengembangan fitur berjalan sesuai rencana. Tabel 3 menunjukkan rincian tugas dan estimasi *story point* untuk setiap *sprint*.

Tabel 3. *Sprint Planning*

No	Sprint	Product Backlog	Waktu
1	Sprint 1	Analisis kebutuhan sistem <i>backend</i> , Merancang arsitektur <i>backend</i> , Merancang ERD	14 Hari
2	Sprint 2	Membuat <i>boilerplate code</i> untuk <i>backend</i> , Setup server VPS untuk <i>deployment</i> , Implementasi fitur Login	14 Hari
3	Sprint 3	Implementasi fitur <i>Register</i> , Implementasi fitur <i>Refresh Token</i> , Implementasi fitur <i>Input Ads Property</i>	14 Hari
4	Sprint 4	Implementasi <i>Input Polygon</i> , Implementasi validasi <i>overlaps polygon</i> , Implementasi fitur <i>Get All Property</i>	14 Hari
5	Sprint 5	Implementasi fitur <i>Get DetailProperty</i> , Implementasi fitur <i>Update Data Property</i> , Implementasi fitur <i>Delete Data Property</i>	14 Hari
6	Sprint 6	Implementasi fitur pencarian data berdasarkan nama daerah, Implementasi fitur pencarian data properti dengan <i>polygon</i> , Membuat visualisasi pada data GeoJSON	14 Hari

### 3.1.4. Use Case Diagram

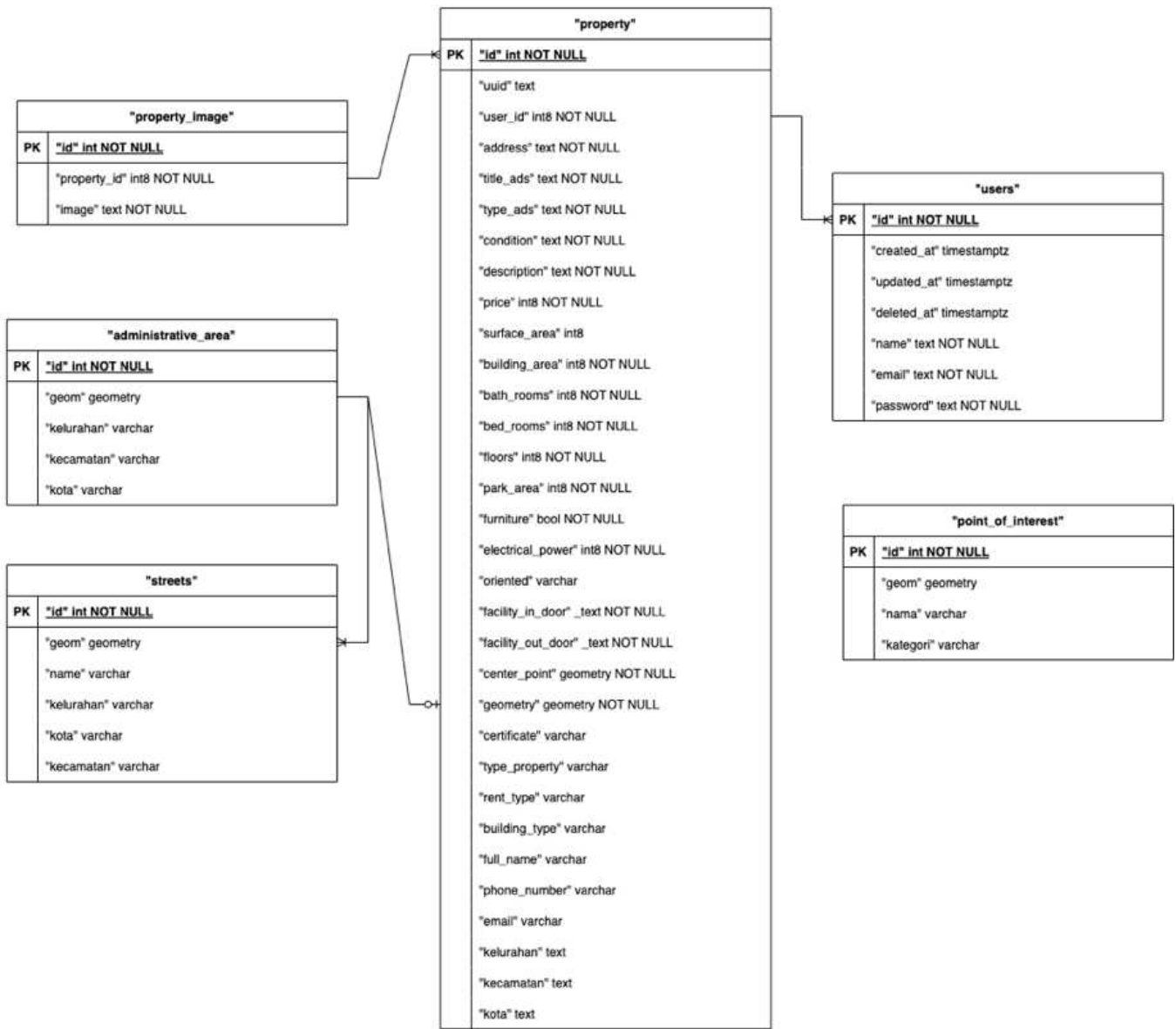
Pada aplikasi *GeoProperty*, *use case diagram* menggambarkan interaksi antara aktor utama seperti agen properti, pengguna individu, dan pengunjung biasa dengan fitur-fitur sistem. Gambar 2 menunjukkan bagaimana setiap aktor menggunakan fitur aplikasi.



Gambar 2. *Use Case* GeoProperty

### 3.1.5. Entity Relationship Diagram (ERD)

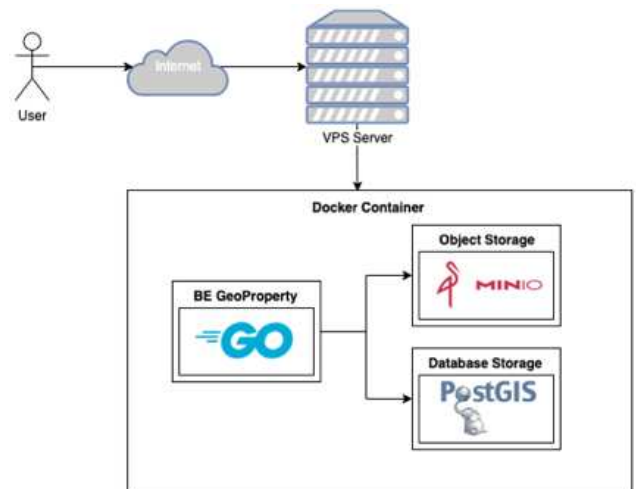
*Entity Relationship Diagram (ERD)* ini dirancang untuk mendukung pengelolaan data dalam aplikasi *GeoProperty*, menggambarkan struktur basis data, relasi antar tabel, dan atribut yang diperlukan. Sistem ini mempermudah pengelolaan properti, wilayah administratif, jalan, dan lokasi menarik dengan struktur data yang efisien. Gambar 3 menunjukkan ERD yang menggambarkan relasi dan struktur data tersebut.



Gambar 3. ERD GeoProperty

### 3.1.6. Rancangan Arsitektur

Arsitektur GeoProperty menggunakan pendekatan *monolith* yang mengintegrasikan *backend* dalam satu aplikasi[13]. *Backend* di *hosting* di VPS dengan Docker, menggunakan *MinIO* sebagai *object storage* untuk menyimpan *file* gambar[14] dan sebagai alternatif ringan dibandingkan layanan *cloud* besar seperti Amazon S3[15], serta PostGIS untuk pengolahan data spasial GIS. Gambar 4 menunjukkan komponen utama dan alur kerja sistem.

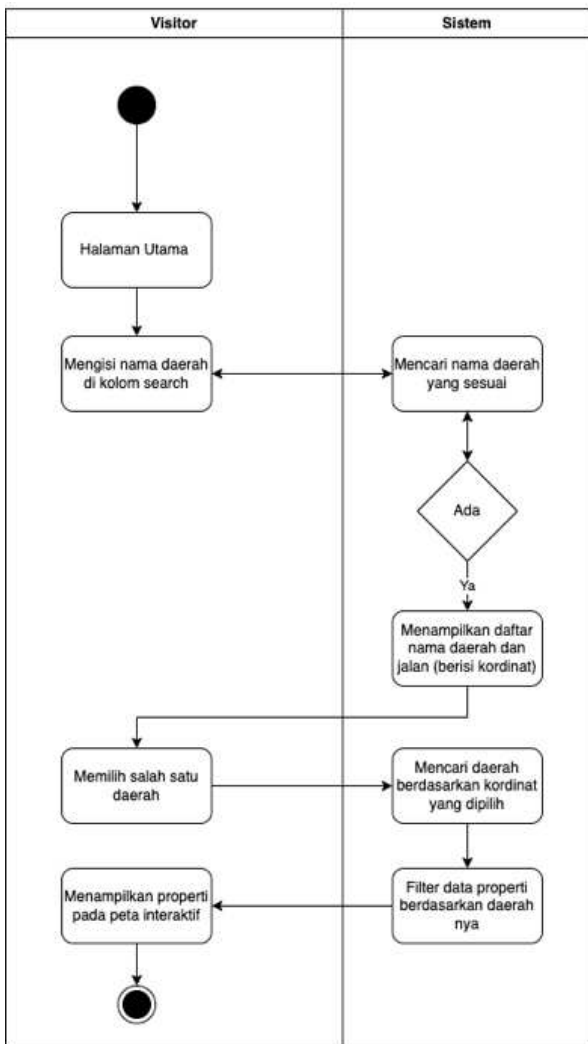


Gambar 4. Arsitektur GeoProperty

3.1.7. Activity Diagram

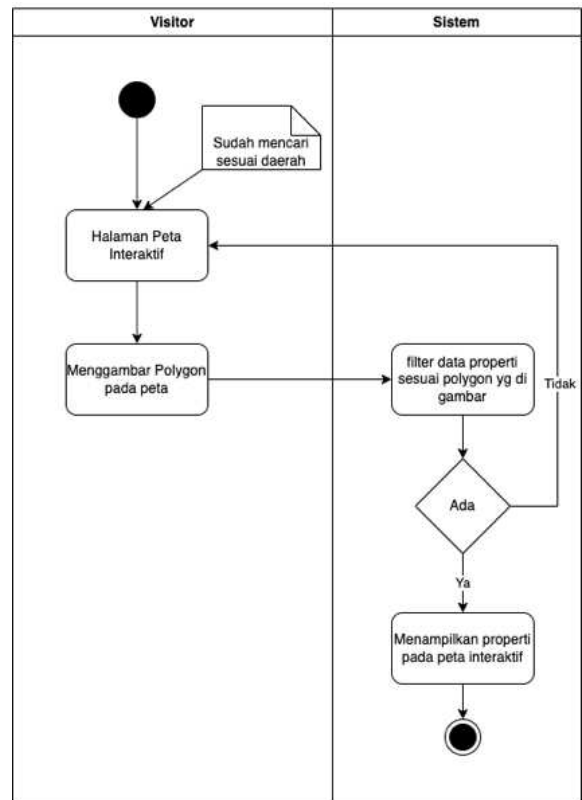
Activity diagram yang disajikan hanya mencakup alur proses dari fitur-fitur utama dalam aplikasi GeoProperty. Diagram ini menggambarkan langkah-langkah penting dalam pengelolaan dan pencarian properti, seperti pencarian berbasis peta interaktif dan penambahan data properti oleh pengguna. Fokus pada fitur utama ini bertujuan untuk memberikan gambaran jelas mengenai proses kerja inti dalam aplikasi.

Gambar 5 menunjukkan alur pencarian properti melalui peta interaktif. Pengguna memulai dengan memasukkan nama daerah pada kolom pencarian. Sistem kemudian menampilkan daftar daerah dan jalan yang sesuai, lengkap dengan koordinatnya. Setelah pengguna memilih daerah yang diinginkan, sistem akan menampilkan properti yang tersedia di area tersebut secara visual pada peta interaktif, memudahkan eksplorasi dan penentuan lokasi.



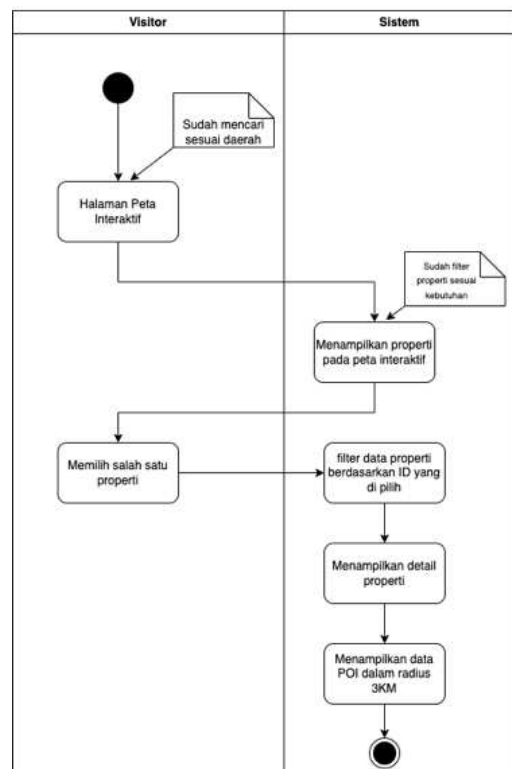
Gambar 5. Peta Interaktif

Gambar 6 menunjukkan proses penyaringan data properti dengan poligon pada peta interaktif. Pengguna menggambar poligon untuk menentukan area pencarian, dan sistem menampilkan properti yang berada di dalam area tersebut.



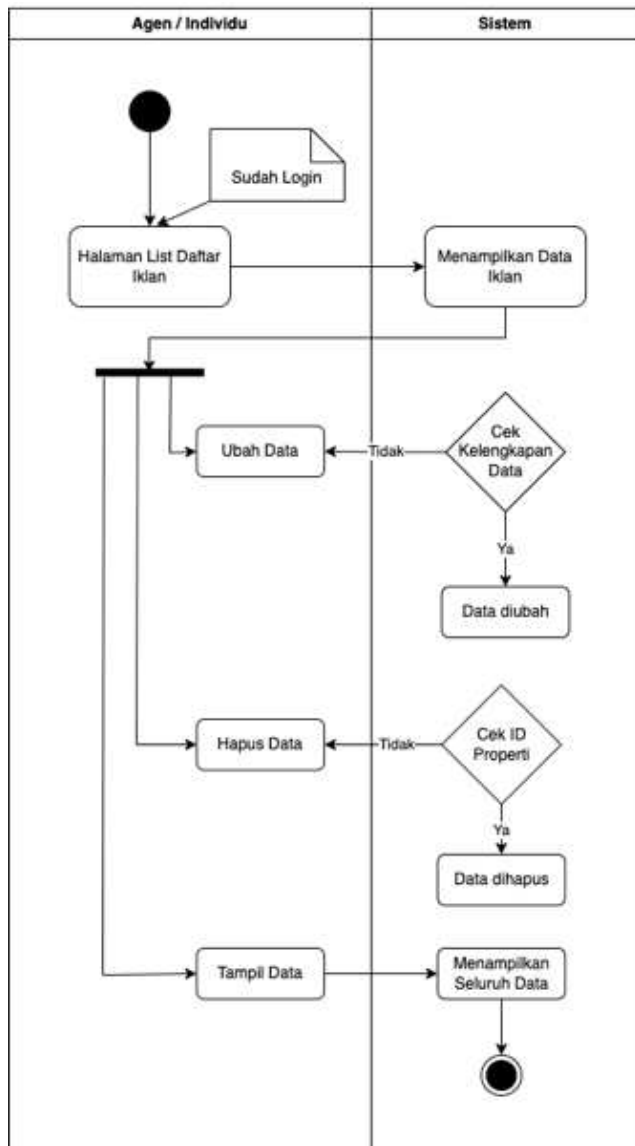
Gambar 6. Filter Data Peta Interaktif dengan Polygon

Gambar 7 menunjukkan proses melihat detail properti dan mencari POI di sekitarnya. Setelah menyaring properti, pengguna dapat memilih properti pada peta interaktif untuk melihat detail lengkapnya, termasuk daftar POI dalam radius 3 kilometer.



Gambar 7. Melihat Data Point of Interest

Gambar 8 menunjukkan alur pengelolaan iklan properti, di mana pengguna dapat melihat, mengedit, atau menghapus iklan yang telah dibuat setelah masuk ke sistem.



Gambar 8. Mengelola Data Property

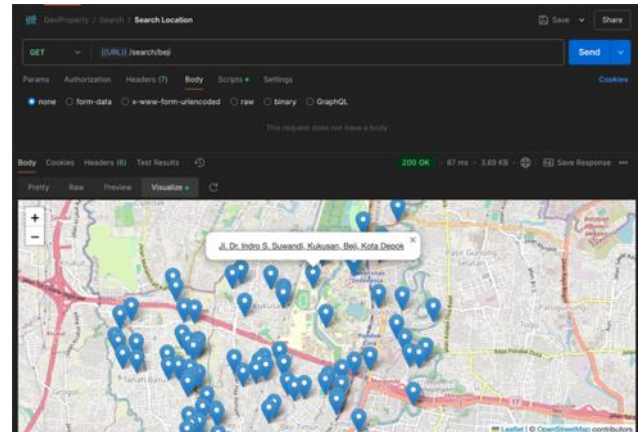
### 3.2. Implementasi Rest-API GeoProperty

Implementasi REST API GeoProperty menggunakan Golang, PostgreSQL, dan PostGIS untuk mendukung fitur penting seperti pencarian berbasis *polygon*, peta interaktif, dan integrasi POI yang berhasil diimplementasikan. Pengembangan dilakukan secara bertahap dengan metode *Agile Scrum* dan pengujian *Black Box* untuk memastikan fungsionalitas yang sesuai.

#### 3.2.1. Search Area

Fitur ini memungkinkan pengguna mencari properti berdasarkan nama wilayah atau jalan tertentu. Seperti yang

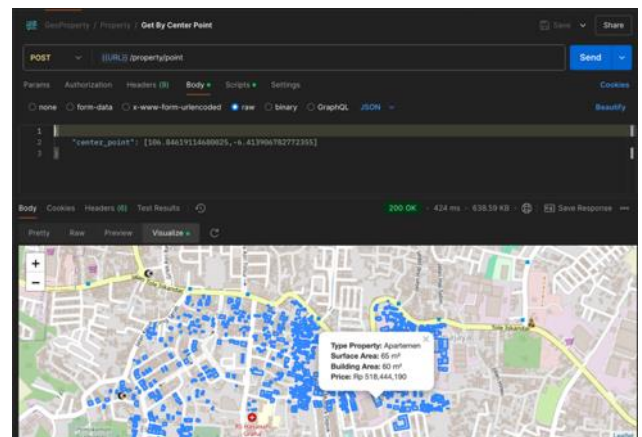
ditunjukkan pada gambar 9, sistem mengembalikan *center point* lokasi yang dicari, yang dapat digunakan untuk pencarian properti lebih lanjut dan dapat divisualisasikan dalam peta menggunakan Postman.



Gambar 9. Search Area

#### 3.2.2. Search by Center Point

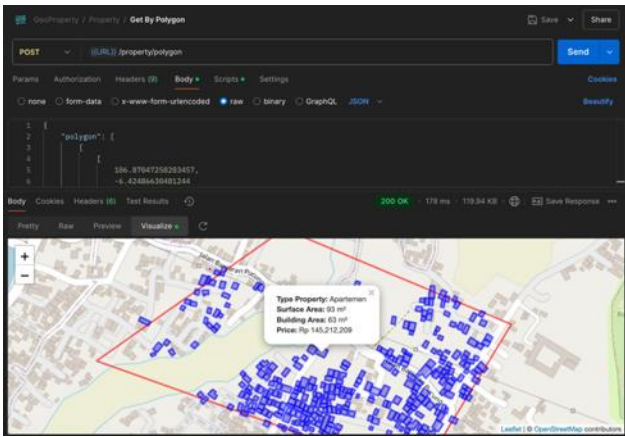
Fitur ini memungkinkan pencarian properti berdasarkan *center point* dari nama daerah atau jalan. Sistem menggunakan koordinat *center point* untuk menampilkan properti dalam radius tertentu. Seperti yang terlihat pada gambar 10, respons dalam format GeoJSON dapat divisualisasikan, memudahkan pengguna melihat properti dan informasi terkait di sekitar lokasi yang dicari.



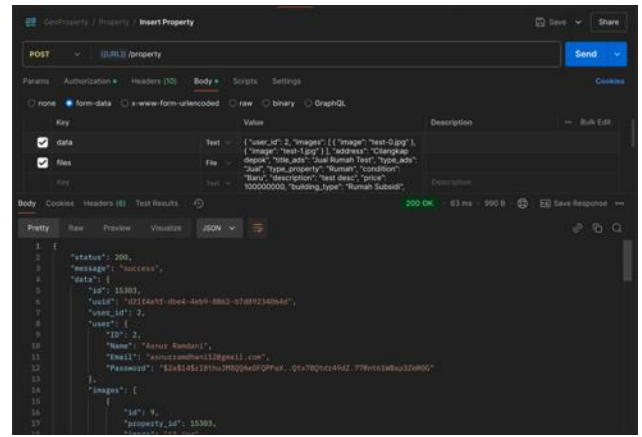
Gambar 10. Search by Center Point

#### 3.2.3. Search by Polygon

Fitur ini memungkinkan pencarian properti dengan menggambar *polygon* pada peta. Seperti pada gambar 11, data spasial properti dalam format GeoJSON divisualisasikan, dan properti dalam area *polygon* ditampilkan, memberikan pengalaman pencarian yang lebih interaktif dan akurat.



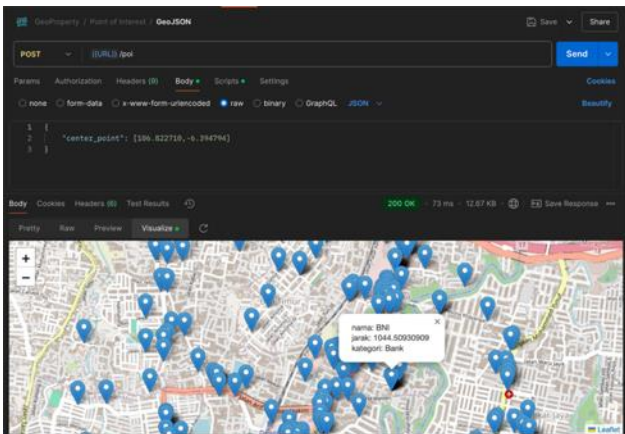
Gambar 11. Search Area by Polygon



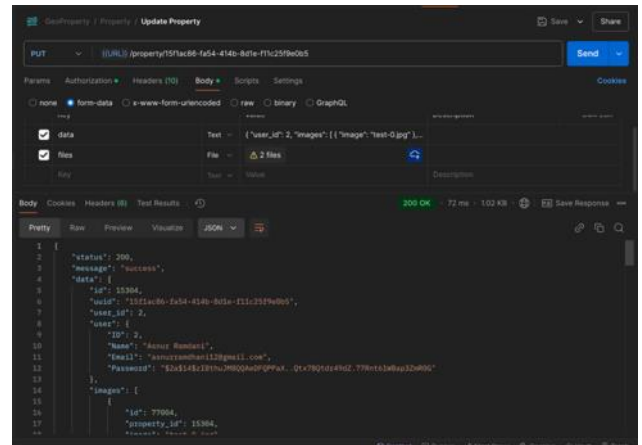
Gambar 13. Insert Data Property

### 3.2.4. Point of Interest

Fitur ini menampilkan informasi tentang fasilitas atau lokasi penting (POI) di sekitar properti, seperti sekolah, rumah sakit, dan transportasi umum, seperti yang terlihat pada gambar 12. Data POI dalam format GeoJSON divisualisasikan, membantu calon pembeli atau penyewa menilai nilai strategis properti berdasarkan lingkungan sekitarnya.



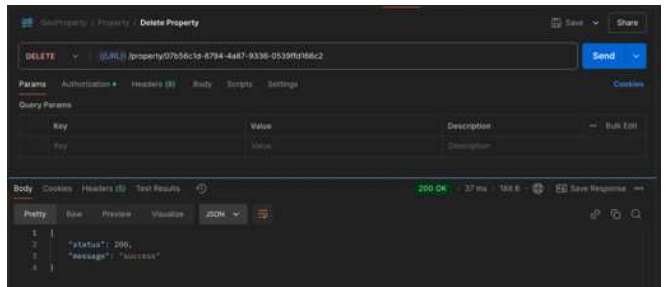
Gambar 12. Point of Interest



Gambar 14. Update Data Property

### 3.2.5. Mengelola Data Properti

Fitur ini memungkinkan pengguna untuk mengelola data properti, termasuk memasukkan, memperbarui, dan menghapus informasi properti. Gambar 13 menunjukkan proses memasukkan data properti baru, gambar 14 menggambarkan cara memperbarui informasi properti yang sudah ada, dan gambar 15 menunjukkan alur untuk menghapus data properti yang tidak diinginkan.



Gambar 15. Delete Data Property

## 3.3. Black Box Testing

Pada bagian ini, dijelaskan gambaran umum mengenai pengujian yang dilakukan terhadap REST API pada aplikasi GeoProperty. Pengujian tersebut meliputi berbagai skenario untuk memastikan fungsionalitas dan kinerja sistem sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Meskipun hanya beberapa contoh yang disajikan di bawah ini, perlu dicatat bahwa pengujian yang dilakukan mencakup total 40 test case, yang terdiri dari berbagai jenis skenario, baik kasus positif maupun negatif. Pengujian ini dilakukan untuk menguji berbagai endpoint API yang ada, mencakup proses pencarian properti, autentikasi pengguna, pengelolaan data properti, serta keamanan sistem. Pengujian terbagi dalam beberapa kategori utama, yaitu:

### 3.3.1. Pengujian Pencarian Lokasi dan Properti

Pengujian ini meliputi pengujian pencarian properti dan lokasi berdasarkan nama daerah, titik koordinat, serta area yang digambar dalam bentuk *polygon*. *Endpoint* yang diuji termasuk */search/:keyword*, */property/point*, dan */property/polygon*. Pengujian juga mencakup validasi hasil pencarian untuk memastikan bahwa sistem dapat memberikan hasil yang sesuai berdasarkan *input* pengguna, serta menangani kondisi *error* jika *input* tidak valid

### 3.3.2. Pengujian Pengelolaan Data Properti

Pengujian ini memastikan bahwa pengguna dapat menambah, memperbarui, dan menghapus data properti melalui API. Pengujian meliputi validasi data *input* properti, termasuk judul, harga, deskripsi, dan *file* gambar. Pengujian juga mencakup penanganan kesalahan, seperti data yang tidak lengkap, format *file* yang salah, dan ukuran *file* yang melebihi batas yang diizinkan. *Endpoint* yang diuji termasuk */property*, */property/:id*, dan */property/own*.

Penulis menggunakan formula untuk menghitung persentase validitas dan non-validitas dari hasil pengujian *black box*. Perhitungan ini memberikan gambaran proporsi skenario uji yang sesuai dengan spesifikasi dibandingkan dengan yang tidak. Hasilnya membantu mengevaluasi kinerja sistem dan mengidentifikasi area yang perlu perbaikan.

$$\text{Presentasi Valid} = \left( \frac{\text{jumlah skenario sesuai}}{\text{total skenario uji}} \right) \times 100$$

$$\text{Presentasi Valid} = \left( \frac{40}{40} \right) \times 100 = 100\%$$

$$\text{Presentasi Tidak Valid} = \left( \frac{\text{jumlah skenario tidak sesuai}}{\text{total skenario uji}} \right) \times 100$$

$$\text{Presentasi Tidak Valid} = \left( \frac{0}{40} \right) \times 100 = 0\%$$

### 3.4. Konfirmasi Pengujian Oleh Pengguna

Berdasarkan wawancara dengan dua informan yang menguji aplikasi GeoProperty langsung melalui *frontend web*, berikut temuan utama:

- Aplikasi mudah digunakan, terutama fitur pencarian dengan filter harga dan lokasi.
- Peta membantu memahami lokasi dan fasilitas sekitar, namun disarankan menambahkan jarak ke tempat penting.
- Aplikasi lancar, meski ada penurunan performa saat memuat data besar dan beberapa *error* teknis pada fitur *polygon*.
- Informan mengusulkan fitur Simpan Favorit, Statistik Iklan, dan perbandingan properti.
- Proses unggah iklan jelas, namun disarankan menyederhanakan unggah foto dan menambahkan validasi otomatis. Fitur Edit iklan sangat dibutuhkan.

### 3.5. Evaluasi

Berdasarkan hasil pengujian *black box* dan konfirmasi pengguna, aplikasi *GeoProperty* menunjukkan kinerja yang baik dengan sebagian besar fitur berfungsi sesuai spesifikasi. Fitur seperti pencarian properti berbasis koordinat berfungsi dengan baik, dan sistem memberikan respons *error* yang konsisten untuk *input* tidak valid. Namun, aplikasi mengalami penurunan performa saat memproses data spasial atau *file* besar, yang juga dilaporkan pengguna terkait pemuatan peta yang melambat. Beberapa fitur penting, seperti Edit iklan properti, belum diimplementasikan, sehingga pengguna harus menghapus dan mengunggah ulang iklan untuk memperbarui informasi. Pengguna mengapresiasi fitur pencarian dan peta interaktif, namun menyarankan peningkatan akurasi pencarian serta penambahan informasi jarak ke tempat penting. Masukan lain mencakup penambahan fitur Simpan Favorit, Statistik Iklan, serta penyederhanaan proses unggah foto dan penambahan validasi otomatis pada deskripsi properti.

## 4. KESIMPULAN

Kesimpulan penelitian ini menunjukkan bahwa *backend* aplikasi GeoProperty berhasil dikembangkan menggunakan Golang, yang efektif dalam mengelola data properti dan mendukung teknologi GIS, khususnya PostGIS, untuk pengolahan data spasial. Pengujian *black box* menunjukkan seluruh fungsi utama *backend* berjalan sesuai spesifikasi. Aplikasi ini juga berhasil menyediakan visualisasi lokasi properti melalui peta interaktif menggunakan format GeoJSON, memungkinkan tampilan informasi spasial yang akurat, termasuk jarak ke fasilitas penting, sehingga mempermudah pengguna memahami posisi properti dan aksesibilitasnya untuk mendukung pengambilan keputusan yang lebih baik.

### Ucapan Terima Kasih

Kami mengucapkan rasa syukur kepada Allah Subhanahu Wa Ta'ala atas rahmat dan hidayah-Nya yang memungkinkan penyelesaian penelitian ini. Ucapan terima kasih kami sampaikan kepada PT. Nerdvana Solusi Teknologi atas fasilitas dan dukungannya selama proses pengembangan, serta kepada dosen pembimbing, Tifanny Nabarian S.Kom., M.T.I., atas bimbingan, arahan, dan kesabarannya selama penelitian ini berlangsung. Kami juga berterima kasih kepada para dosen mata kuliah tugas akhir yang telah memberikan ilmu dan panduan yang menjadi dasar utama penelitian ini. Penghargaan mendalam juga kami tunjukkan kepada keluarga atas doa dan dukungannya, serta kepada rekan-rekan yang telah memberikan motivasi dan bantuan dalam berbagai bentuk. Semoga penelitian ini dapat memberikan kontribusi nyata bagi pengembangan teknologi, khususnya di bidang sistem berbasis GIS, dan menjadi manfaat bagi kemajuan ilmu pengetahuan.

## DAFTAR PUSTAKA

- [1] N. L. Marpaung, S. Hutabarat, and M. H. Saputra, "Perancangan Sistem Promosi Penjualan Property Berbasis Website," *Publikasi Hasil Pengabdian kepada Masyarakat (PADIMAS)*, vol. 2, no. 1, pp. 8–21, 2022, doi: 10.35957/padimas.v2i1.2860.
- [2] National Association of Realtors, "Home Buyers and Sellers Generational Trends Report," 2023.
- [3] Leasey.ai, "Enhancing Website Property Listings with Interactive Property Maps for User Experience." Accessed: Oct. 30, 2024. [Online]. Available: <https://www.leasey.ai/resources/enhancing-website-property-listings-with-interactive-property-maps-for-user-experience/>
- [4] "Unlocking Sales Opportunities with Geospatial and Geofencing in Real Estate." Accessed: Oct. 30, 2024. [Online]. Available: <https://www.gispeople.com.au/unlocking-sales-opportunities-with-geospatial-and-geofencing-in-real-estate/>
- [5] H. Harfizar, M. M. Sari, and R. Marbun, "Rancang Bangun Website Pemasaran dan Penjualan Properti (Study Kasus : PT Indah Cemani Raya)," *Journal Cerita*, vol. 8, no. 2, pp. 171–176, 2022, doi: 10.33050/cerita.v8i2.2454.
- [6] E. H. Budi R, "Analisis Lokasi Penjualan Berbasis Data Geografis Menggunakan Perangkat Lunak Google Map API Studi Kasus PT. Johnson Home Hygiene Product," *Jurnal Ilmiah ILKOMINFO - Ilmu Komputer & Informatika*, vol. 3, no. 1, pp. 21–28, 2020, doi: 10.47324/ilkominfo.v3i1.56.
- [7] P. Fosci and G. Psaila, *Soft Querying Features in GeoJSON Documents: The GeoSoft Proposal*, vol. 16, no. 1. Springer Netherlands, 2023. doi: 10.1007/s44196-023-00325-3.
- [8] J. F. Germinian and S. T. Tricya Esterina Widagdo, "Utilizing PostGIS Extension to Process Spatial Data Stored in Neo4j Database," *Proceedings of 2023 IEEE International Conference on Data and Software Engineering, ICoDSE 2023*, pp. 250–255, 2023, doi: 10.1109/ICoDSE59534.2023.10291400.
- [9] Bhumi Varta Technology, "Point of Interest (POI) : Mengapa Data ini Penting?," <https://bvarta.com/Id/Point-of-Interest-Poi-Mengapa-Data-Ini-Penting/>. Accessed: Sep. 27, 2024. [Online]. Available: [https://bvarta.com/id/point-of-interest-poi-mengapa-data-ini-penting/#Mengenal\\_Point\\_of\\_Interest](https://bvarta.com/id/point-of-interest-poi-mengapa-data-ini-penting/#Mengenal_Point_of_Interest)
- [10] N. A. Putri, "Scrum Prinsip Agile dan Tahapan Dalam Metode Scrum," PuTI. Accessed: Sep. 28, 2024. [Online]. Available: <https://bif.telkomuniversity.ac.id/scrum-prinsip-agile-dan-tahapan-dalam-metode-scrum/>
- [11] "Apa itu Blackbox Testing." Accessed: Sep. 30, 2024. [Online]. Available: <https://smarteksistem.com/blackbox-testing-adalah/?lang=id>
- [12] Okpatrioka Okpatrioka, "Research And Development (R&D) Penelitian Yang Inovatif Dalam Pendidikan," *Dharma Acariya Nusantara: Jurnal Pendidikan, Bahasa dan Budaya*, vol. 1, no. 1, pp. 86–100, 2023, doi: 10.47861/jdan.v1i1.154.
- [13] R. Su and X. Li, "Modular Monolith: Is This the Trend in Software Architecture?," pp. 10–13, 2024, doi: 10.1145/3643657.3643911.
- [14] "Object Storage: Pengertian, fungsi, dan Keunggulannya." Accessed: Sep. 28, 2024. [Online]. Available: <https://cloudraya.com/blog/object-storage-pengertian-fungsi-dan-keunggulannya/>
- [15] "Object Storage MinIO." Accessed: Oct. 02, 2024. [Online]. Available: <https://min.io/product/overview>