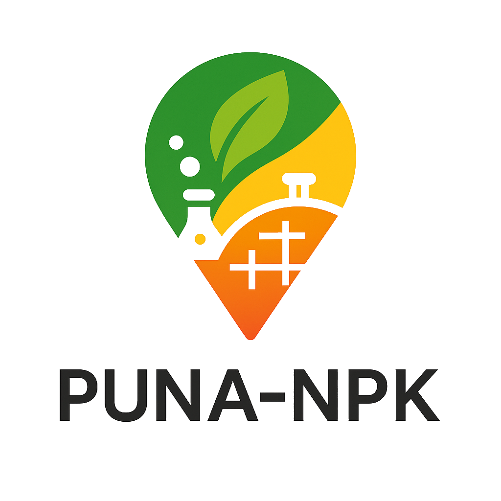
**MANUAL BOOK**

**SINEMA: Landslide Inventory Mapping**



A screenshot of a computer

Description automatically generated

**BADAN RISET DAN INOVASI NASIONAL**

**2025**

**INFORMASI APLIKASI**

Hak cipta yang diusulkan berupa perangkat lunak aplikasi (software application) dengan nama dan judul:

“SINEMA: Landslide Inventory Mapping”

Aplikasi ini digunakan untuk mendokumentasikan dan memetakan kejadian tanah longsor secara terstruktur. Pengguna dapat menggambar area longsor menggunakan polygon di peta, mengisi data detail (lokasi, tanggal, kondisi area, kondisi longsor, dampak, faktor penyebab, mekanisme), serta mengunggah foto pendukung setiap kejadian. Hasilnya tersimpan di basis data PostGIS sehingga dapat dianalisis atau diekspor kapan saja.

Aplikasi ini merupakan pengembangan dari sistem pemetaan web berbasis Leaflet.js dan Flask, dengan tambahan:

* Manajemen pengguna (JWT authentication)
* Penyimpanan atribut detil longsor
* Dukungan unggah foto untuk setiap entri

**A screenshot of a login screen

AI-generated content may be incorrect.**

**A screenshot of a login form

AI-generated content may be incorrect.**

**1. Keunggulan Aplikasi**

**Pemetaan Interaktif Berbasis GIS**

Dengan integrasi Leaflet, pengguna dapat langsung menandai titik lokasi pengukuran di peta dunia nyata, memudahkan visualisasi sebaran data unsur hara tanpa memerlukan perangkat lunak SIG khusus. Dibangun dengan Bootstrap 5, dashboard dan form-form aplikasi otomatis menyesuaikan tampilan pada desktop, tablet, maupun ponsel, sehingga dapat digunakan di lapangan langsung lewat tablet atau smartphone. Dashboard menyediakan filter provinsi dan jenis tanaman, serta chart interaktif (pie/bar) untuk melihat distribusi titik per wilayah atau per jenis tanaman dengan sekali klik—mempercepat analisis awal tanpa keluar dari browser.

**Pengukuran Multi-Parameter Tanah**

Selain koordinat dan metadata lahan, PUNA-NPK mendukung pencatatan mendalam (soil\_depth) serta kadar Nitrogen, Phosporus, dan Kalium dalam satu entri—menggantikan aplikasi satu-parameter yang hanya mencatat satu unsur saja.

**Otentikasi dan Keamanan Data**

Semua endpoint dilindungi JWT (JSON Web Token), sehingga hanya pengguna terdaftar dengan token valid yang dapat mengakses, menambah, atau mengubah data—menjaga kerahasiaan dan integritas hasil pengukuran.

**Dukungan Unggah Foto Lapangan**

Fitur upload foto memungkinkan dokumentasi kondisi lapangan (contoh: tekstur tanah, kondisi tanaman) tersimpan bersama titik data, membantu validasi dan audit kualitas pengambilan sampel.

**Ekstensibilitas dan Open-Source**

* **Backend:** Flask + SQLAlchemy + GeoAlchemy2 — mudah diperluas dan dipelihara.
* **Frontend:** Bootstrap, Leaflet, Chart.js — komponen terbuka yang banyak dokumentasi dan komunitas.
* Strukturnya modular, memudahkan penambahan parameter baru (misalnya pH tanah, kelembaban).

**Pengelolaan Master Data Terpusat**

CRUD untuk Provinsi, Kota, Crop Types, Soil Types, dan Fertilization Types memastikan konsistensi data master di seluruh aplikasi tanpa harus langsung intervensi basis data.

**2. Kebaruan Aplikasi**

PUNA-NPK menghadirkan inovasi signifikan dengan menggabungkan pemetaan spasial interaktif dan pencatatan multi-parameter unsur hara dalam satu platform web. Berbeda dengan banyak aplikasi pengelolaan tanah yang hanya menampilkan data tabel atau grafik terpisah, PUNA-NPK memvisualisasikan titik-titik pengukuran Nitrogen, Phosporus, dan Kalium langsung di peta dunia nyata, sehingga pengguna dapat memahami sebaran kualitas tanah secara instan. Selain itu, integrasi upload foto lapangan ke setiap titik pengukuran memungkinkan dokumentasi visual kondisi tanah dan tanaman yang real-time—fitur yang belum umum ditemukan di aplikasi serupa.

Dibandingkan dengan sistem pemetaan tanah konvensional seperti SoilGrids atau platform pantau pertanian digital yang biasanya berfokus pada satu unsur atau memerlukan lisensi mahal, PUNA-NPK menggunakan komponen open-source (Leaflet, Chart.js, Flask) sehingga dapat dikustomisasi dan dikembangkan oleh lembaga penelitian maupun dinas pertanian daerah tanpa biaya lisensi. Keunggulan lain adalah kemampuan filter dinamis berdasarkan provinsi dan jenis tanaman, yang memungkinkan analisis cepat antar wilayah dan varietas, sesuatu yang jarang ditemui pada aplikasi benchmark. Dengan demikian, PUNA-NPK tidak hanya mempercepat pengambilan keputusan manajemen pemupukan, tetapi juga membuka peluang kolaborasi dan inovasi lebih lanjut dalam riset agro-ekosistem.

**3. Spesifikasi Kebutuhan Aplikasi**

| **Kategori** | **Rincian** |
| --- | --- |

|  |  |
| --- | --- |
| **Bahasa Pemrograman** | - **Backend**: Python 3.8+ dengan Flask 2.x - **Frontend**: HTML5, CSS3 (Bootstrap 5.3), JavaScript (ES6) |

|  |  |
| --- | --- |
| **Syarat Minimum Komputer** | - **OS**: Windows 10 / macOS Catalina / Ubuntu 20.04 - **CPU**: Intel Core i5 atau setara - **RAM**: ≥ 4 GB - **Disk**: ≥ 10 GB ruang kosong - **Web Server**: Nginx atau Apache |

|  |  |
| --- | --- |
| **Ketergantungan terhadap Software Lain** | - **Python packages**: SQLAlchemy, GeoAlchemy2, Shapely, Psycopg2, Flask-JWT-Extended, Flask-CORS - **PostGIS**: ekstensi GIS untuk PostgreSQL - **Node.js / npm** (opsional, untuk build frontend) |

|  |  |
| --- | --- |
| **Jenis Server** | - **Produksi**: Gunicorn 20.x (WSGI) di-backing Nginx 1.18+ - **Pengembangan**: Flask built-in server |

|  |  |
| --- | --- |
| **Jenis Database** | PostgreSQL 13+ dengan ekstensi PostGIS 3.x |

|  |  |
| --- | --- |
| **Framework & Library Lain** | - **CSS/JS**: Bootstrap 5.3, jQuery 3.6 - **Peta & Visualisasi**: Leaflet 1.9, Chart.js 3.x - **Upload File**: Dropzone.js (opsional) |

**4. Algoritma Aplikasi**

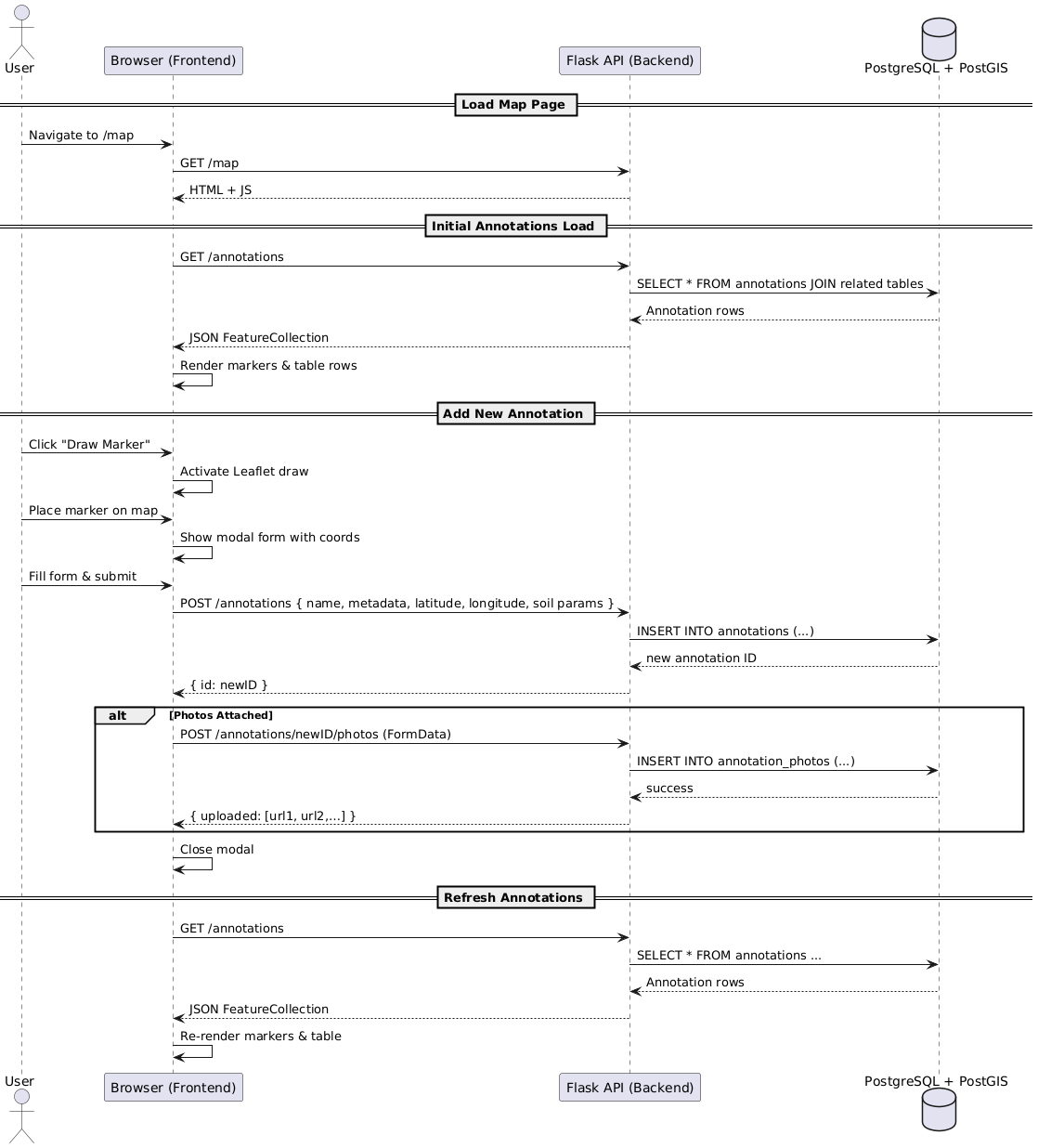
|  |
| --- |
| 1. Mulai Aplikasi  └─ Tampilkan halaman Login/Register  2. Autentikasi  ├─ Jika login berhasil → Simpan JWT di browser → Arahkan ke Dashboard  └─ Jika belum terdaftar → Isi form Register → Simpan akun di DB → Arahkan ke Login  3. Tampilkan Dashboard  ├─ Panggil API `/annotations`, `/crop\_types`, `/soil\_types`, `/fertilization\_types`, `/provinces`, `/cities`  ├─ Hitung & render statistik ringkas  └─ Tampilkan 5 anotasi terbaru  4. Akses Halaman “Manage Map”  ├─ Inisialisasi peta Leaflet (tile layer + kontrol layer)  ├─ Panggil fungsi `loadMasterData()` untuk mengisi dropdown  ├─ Panggil fungsi `loadAnnotations()`:  │ ├─ GET `/annotations` → terima GeoJSON + properti  │ ├─ Untuk tiap feature:  │ │ ├─ Render marker pada peta  │ │ └─ Tambahkan baris pada tabel data  │ └─ Perbarui hitungan anotasi  └─ Pasang event handler untuk pengeditan & penghapusan marker  5. Menambah Anotasi Baru  ├─ User klik ikon “Tambah” → aktifkan kontrol `draw` marker  ├─ Setelah marker dibuat:  │ ├─ Simpan koordinat latitude/longitude di input tersembunyi  │ ├─ Tampilkan modal form isian metadata & parameter tanah  │ └─ Tunggu `submit` form  ├─ Saat `submit`:  │ ├─ Kumpulkan semua field → bangun `payload` JSON  │ ├─ POST ke endpoint `/annotations`  │ ├─ Jika ada file foto → POST ke `/annotations/<id>/photos`  │ └─ Tutup modal & panggil ulang `loadAnnotations()`  └─ Jika request gagal → tampilkan error di console/browser  6. Mengedit Anotasi  ├─ User klik tombol Edit di baris tabel  ├─ Fill form dengan data lama, tampilkan modal  ├─ Saat `submit`: PUT ke `/annotations/<id>` dengan field yang berubah  ├─ Tutup modal & refresh peta + tabel  7. Menghapus Anotasi  ├─ User klik tombol Delete → konfirmasi  └─ Jika “OK”: DELETE ke `/annotations/<id>` → refresh peta + tabel  8. CRUD Master Data (Provinsi, Kota, Crop, Soil, Fert. Types)  ├─ Halaman khusus masing-masing:  │ ├─ GET list → render tabel  │ ├─ Form create → POST, lalu reload tabel  │ └─ Tombol delete → DELETE, lalu reload tabel  └─ (Opsional) Edit via tombol Edit → PUT → reload  9. Selesai |

Gambar 7. Algoritma dari aplikasi PUNA-NPK

Algoritma PUNA-NPK dimulai ketika pengguna membuka aplikasi dan melewati proses autentikasi—login atau pendaftaran—di mana kredensial disimpan dalam token JWT untuk sesi selanjutnya. Setelah berhasil masuk, pengguna diarahkan ke Dashboard yang langsung memanggil berbagai endpoint (seperti anotasi, jenis tanaman, jenis pupuk, provinsi, dan kota) untuk menampilkan statistik ringkas dan daftar lima anotasi terbaru. Ketika pengguna mengakses halaman “Manage Map”, peta Leaflet diinisialisasi dengan lapisan tile dan kontrol layer; kemudian master data (crop, soil, fertilization, provinsi, kota) dimuat ke dropdown, dan fungsi loadAnnotations() mengambil GeoJSON anotasi dari server untuk menampilkan marker di peta dan mengisi tabel data.

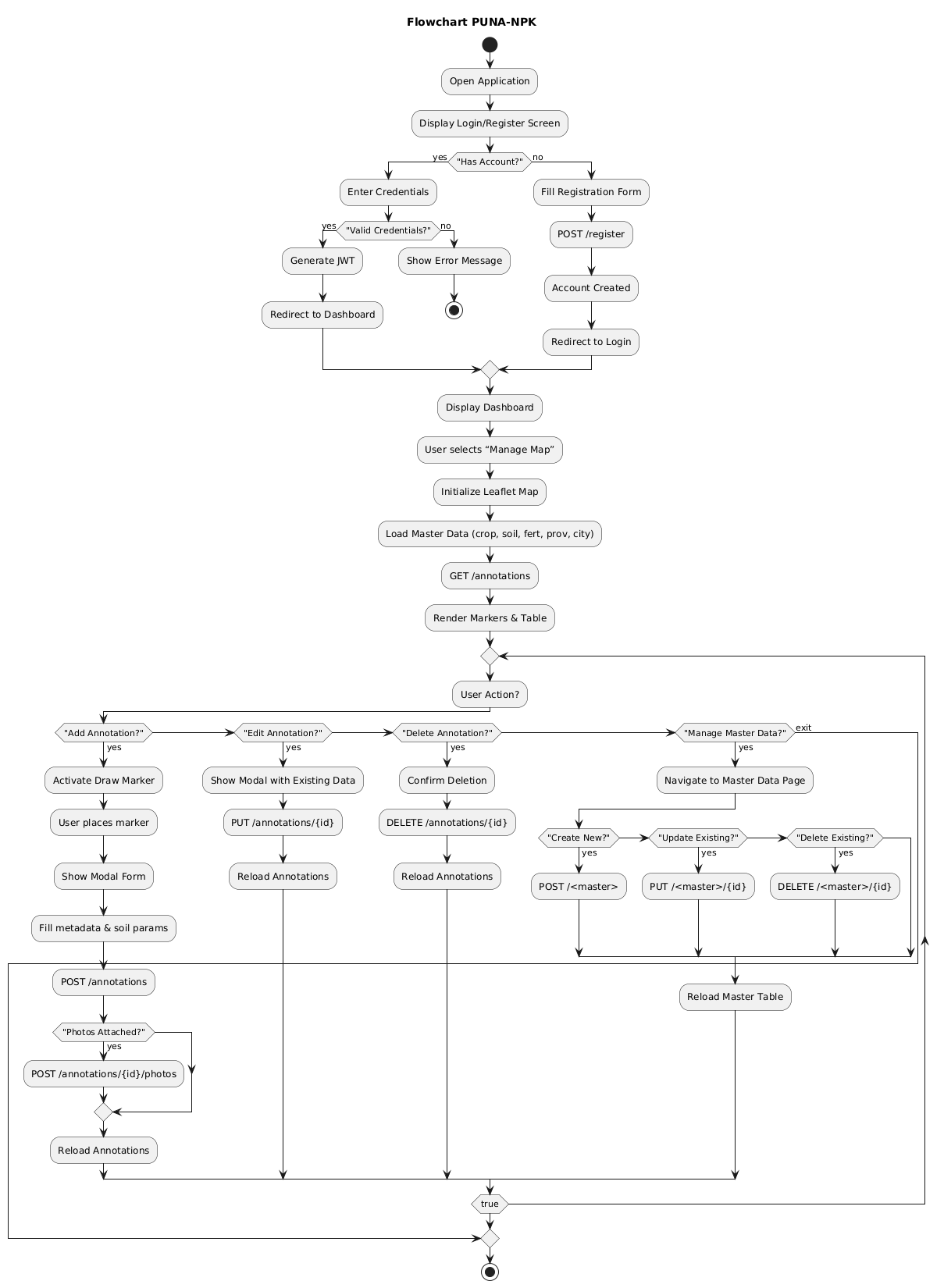
Untuk menambah data, pengguna mengklik kontrol “draw marker” pada peta; setelah marker terpasang, koordinat latitude/longitude disimpan di input tersembunyi dan modal form ditampilkan untuk memasukkan metadata (nama, lokasi, provinsi, dsb.) serta parameter tanah (kedalaman, N, P, K). Saat form disubmit, payload JSON dikirim via POST ke endpoint /annotations, diikuti upload foto (jika ada) ke /annotations/<id>/photos, lalu peta dan tabel direfresh. Proses edit hampir sama: data lama diisi kembali dalam form, kemudian dikirim PUT ke /annotations/<id> dan tampilan diperbarui; sedangkan penghapusan langsung memanggil DELETE dan memuat ulang.

Selain itu, aplikasi menyediakan halaman CRUD terpisah untuk master data (provinsi, kota, jenis tanaman, jenis tanah, jenis pupuk), di mana operasi GET menampilkan daftar, POST menambah item baru, PUT mengubah nama, dan DELETE menghapus entri—semua dengan reload tabel setelah aksi sukses. Pola alur ini memastikan bahwa setiap perubahan data terintegrasi secara real-time ke peta dan tabel, memberikan pengalaman pengguna yang konsisten dan responsif dalam pemetaan unsur hara tanah.



Gambar 8. Sequence diagram aplikasi PUNA-NPK

**5 Cara Penggunaan Aplikasi**



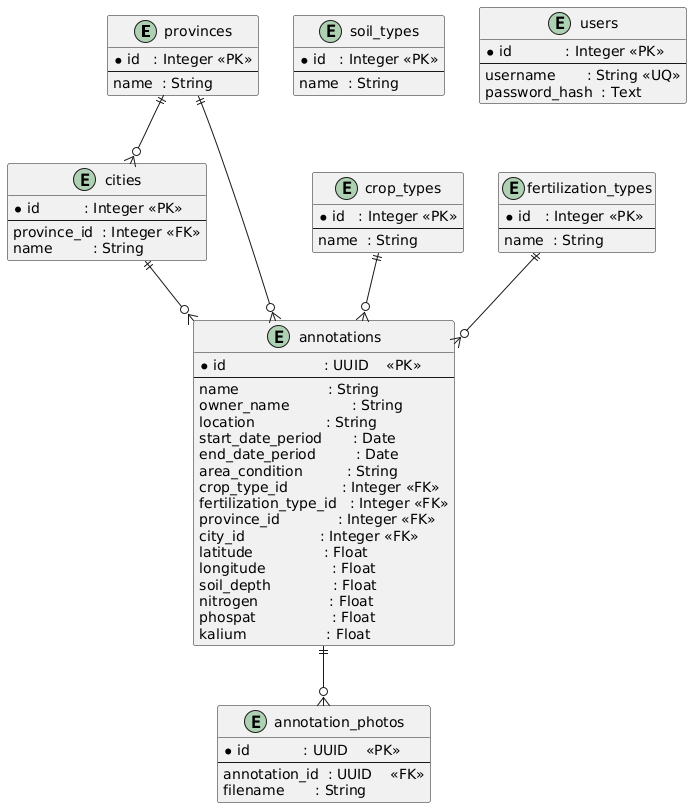
Gambar 9. Flowchart aplikasi PUNA-NPK

A screenshot of a login screen

AI-generated content may be incorrect.

Alur aplikasi PUNA-NPK dimulai ketika pengguna membuka halaman login—jika sudah memiliki akun, mereka memasukkan kredensial, aplikasi memverifikasi dan menghasilkan token JWT, lalu mengarahkan mereka ke Dashboard; jika belum, pengguna dapat mendaftar terlebih dahulu, setelah itu kembali ke halaman login. Di Dashboard, pengguna dapat memilih menu “Manage Map”, yang akan menginisialisasi peta Leaflet, memuat data master (jenis tanaman, jenis tanah, jenis pupuk, provinsi, kota) dan memanggil endpoint /annotations untuk menampilkan marker dan baris tabel. Pada mode peta, pengguna dapat menambahkan anotasi baru dengan menggambar marker, mengisi form metadata serta parameter tanah, dan mengirimkan data lewat POST ke server—jika disertai foto, aplikasi akan mengunggahnya ke endpoint khusus. Untuk mengubah data, pengguna mengklik tombol Edit di tabel, memodifikasi form yang sudah terisi, lalu melakukan PUT; untuk menghapus, pengguna mengonfirmasi dan aplikasi memanggil DELETE. Selain itu, tiap halaman master data (Provinsi, Kota, Crop, Soil, Fert. Types) mendukung operasi CRUD serupa—GET daftar, POST tambah, PUT ubah, DELETE hapus—dan setelah setiap aksi halaman langsung memuat ulang datanya. Dengan demikian, setiap interaksi CRUD secara real-time tercermin di peta dan tabel, hingga pengguna selesai menggunakan aplikasi.

**6. Diagram Relasi Database**



Gambar 10. Diagram relasi database dari aplikasi PUNA-NPK

Pada ERD PUNA-NPK terlihat delapan entitas utama. Provinces menyimpan daftar provinsi (PK id) yang menjadi induk bagi Cities (FK province\_id) dan juga dapat menjadi lokasi anotasi. Cities (PK id) memetakan nama kota dan mengacu ke provinsi asalnya. Crop\_Types, Fertilization\_Types, dan Soil\_Types masing-masing menyimpan jenis tanaman, pupuk, dan kategori jenis tanah (PK id) yang dipakai dalam setiap anotasi. Users berisi data pengguna (PK id) dengan username unik dan hash password untuk autentikasi.

Entitas Annotations (PK id bertipe UUID) adalah pusat, menggabungkan metadata lapangan (nama, lokasi, periode, kondisi, koordinat, parameter tanah seperti N, P, K, dan soil\_depth) serta referensi ke crop\_type\_id, fertilization\_type\_id, province\_id, city\_id—semua sebagai FK ke tabel master terkait. Relasi one-to-many setiap master-data ke Annotations memungkinkan banyak titik pengukuran di berbagai kategori. Terakhir, Annotation\_Photos menyimpan file foto (PK id UUID) yang diunggah untuk setiap anotasi (FK annotation\_id), sehingga satu anotasi dapat memiliki beberapa foto pendukung. Struktur ini memastikan konsistensi data lokasi, jenis tanaman, jenis pupuk, dan parameter tanah di seluruh aplikasi.

**7. Panduan Instalasi Aplikasi**

**1. Prasyarat**

* **Git** (untuk clone repository)
* **Python 3.9+**
* **PostgreSQL 12+** dengan ekstensi **PostGIS**
* Web server (opsional, untuk deploy produksi)

**2. Clone Repository**

bash

CopyEdit

git clone https://git.brin.go.id/tauf022/puna-npk.git

cd puna-npk

**3. Siapkan Virtual Environment**

bash

CopyEdit

python3 -m venv venv

source venv/bin/activate # Linux/macOS

venv\Scripts\activate.bat # Windows

**4. Install Dependencies**

bash

CopyEdit

pip install --upgrade pip

pip install -r requirements.txt

**5. Konfigurasi Database**

1. **Buat database** di PostgreSQL:

sql

CopyEdit

CREATE DATABASE punanpkdb;

\c punanpkdb

CREATE EXTENSION postgis;

1. **Atur koneksi** di app.py (baris konfigurasi SQLALCHEMY\_DATABASE\_URI) jika perlu:

python

CopyEdit

app.config['SQLALCHEMY\_DATABASE\_URI'] =

'postgresql://<user>:<password>@<host>/<dbname>'

Contoh default:

python

CopyEdit

'postgresql://postgres:root@localhost/punanpkdb'

**6. Inisialisasi Skema**

Pada run pertama, aplikasi akan otomatis membuat tabel yang belum ada berkat blok inspeksi di bawah with app.app\_context(): — Anda cukup menjalankan:

bash

CopyEdit

python app.py

Output akan mencetak nama-nama tabel yang dibuat.

**7. Jalankan Aplikasi (Development)**

bash

CopyEdit

export FLASK\_APP=app.py

export FLASK\_ENV=development

flask run

atau langsung:

bash

CopyEdit

python app.py

**Default** server: <http://127.0.0.1:5000>

**8. Akses Aplikasi**

* **Login/Register**: <http://127.0.0.1:5000/>
* **Dashboard**: <http://127.0.0.1:5000/dashboard>
* **Map Management**: <http://127.0.0.1:5000/map>

**8. Source code**

<https://git.brin.go.id/tauf022/puna-npk>