

Pengembangan Aplikasi Web Sistem Pengaduan Masyarakat "Lapor Yuk" Berbasis Progressive Web App (PWA) Menggunakan Framework Django dan Cloud Database PostgreSQL

Anggota:

1. Jihad Amrullah (2411102441287)
2. Muhammad Sofwan (2411102441059)
3. Hajril Pratama Yajid Saputra (2411102441055)
4. Muhammad Maulana (2411102441043)
5. Muhammad Daffa Izzullah Nur (2411102441252)

Program Studi: Teknik Informatika

Dosen Pengampu: Hendra Saputra, S.Kom., M.Kom.

Tahun Akademik: 2026

ABSTRAK

Perkembangan teknologi informasi saat ini menuntut tersedianya media pelaporan yang cepat, transparan, dan terstruktur. Namun, proses pengaduan masyarakat sering kali masih dilakukan secara konvensional atau melalui media sosial yang tidak terdokumentasi dengan baik, sehingga menghambat proses tindak lanjut dari pihak berwenang. Penelitian ini bertujuan untuk merancang dan membangun aplikasi web pengaduan masyarakat bernama "Lapor Yuk" sebagai solusi digital yang modern dan efisien menggunakan arsitektur *Model-View-Template* (MVT) dari *framework* Django. Metode pengembangan sistem yang digunakan berbasis arsitektur *client-server* dengan pendekatan *mobile-first*. Pada sisi *frontend*, aplikasi mengimplementasikan HTML, JavaScript, Tailwind CSS, dan teknologi *Progressive Web App* (PWA) untuk memberikan pengalaman responsif layaknya aplikasi *native*, serta pustaka Leaflet.js untuk pemetaan lokasi interaktif. Sisi *backend* dibangun menggunakan bahasa pemrograman Python yang terhubung ke basis data relasional PostgreSQL pada platform *cloud* Supabase, serta di-*hosting* melalui Vercel. Hasil dari pengembangan ini adalah sebuah sistem aplikasi web yang mempermudah masyarakat dalam mengirimkan laporan, melihat riwayat pengaduan, serta memantau status penanganan secara *real-time*. Kesimpulan dari proyek ini menunjukkan bahwa integrasi teknologi Django, PWA, dan layanan *cloud database* berhasil meningkatkan transparansi, akuntabilitas, dan efisiensi dalam manajemen pelaporan pengaduan masyarakat.

- **Kata Kunci:** Aplikasi Pengaduan, Django, Leaflet.js, PostgreSQL, Progressive Web App (PWA)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi di era globalisasi saat ini telah membawa perubahan yang sangat signifikan di berbagai aspek kehidupan, meliputi bidang pendidikan, ekonomi, politik, sosial, dan budaya. Dalam bidang pemerintahan, pengelolaan data yang cepat, akurat, dan efisien sangat diperlukan untuk menunjang dan mempermudah proses kerja dalam melayani masyarakat. Namun, fakta di lapangan menunjukkan bahwa banyak instansi yang belum mengoptimalkan teknologi pelaporan terpadu. Sebagai contoh, website merupakan salah satu opsi yang dapat digunakan untuk proses pengaduan masyarakat kepada lembaga pemerintahan [3].

Apabila instansi publik masih mengandalkan cara manual, dampak buruk terhadap efisiensi kerja tidak dapat dihindarkan. Sistem yang belum terintegrasi dianggap tidak efektif karena prosesnya yang panjang sehingga pengaduan tidak segera diselesaikan, dan pencatatan laporan pengaduan yang masih menggunakan kertas sangat berisiko untuk hilang atau rusak [4]. Dampak ketidakpastian ini diperparah oleh fakta bahwa masyarakat sering mengalami kesulitan untuk mengetahui perkembangan dari laporan yang telah dikirimkan akibat kurangnya transparansi sistem. Oleh karena itu, diperlukan sebuah sistem berbasis web yang mampu mempermudah proses pelaporan sekaligus memberikan informasi status laporan secara jelas dan *real-time*.

Pentingnya solusi yang dibangun melalui aplikasi web "Lapor Yuk" adalah untuk menyediakan wadah digital yang andal bagi warga sekaligus menjadi jembatan utama yang menghubungkan interaksi pengguna dengan pangkalan data. Agar sistem memiliki tata kelola yang rapi, aman, dan efisien, aplikasi ini dikembangkan dengan bahasa pemrograman Python menggunakan *framework* Django berbasis arsitektur *Model-View-Template* (MVT). Penggunaan kerangka kerja ini terbukti efektif untuk meningkatkan tata kelola dan mengotomatisasi proses administrasi yang sebelumnya rentan terhadap kesalahan [5].

Selain keandalan logika bisnis di sisi *backend*, aplikasi ini mengintegrasikan komponen peta interaktif menggunakan pustaka *open-source* Leaflet.js. Visualisasi geografis ini krusial agar pengguna dapat menjelajahi informasi secara visual dan spasial melalui kemampuan sistem dalam menyajikan peta interaktif yang kaya informasi [1].

Selanjutnya, untuk menjamin keamanan dan skalabilitas, aplikasi "Lapor Yuk" tidak lagi menggunakan pangkalan data lokal, melainkan beralih ke database relasional *cloud* PostgreSQL di Supabase. Perlindungan pangkalan data ini sangat vital, karena ketergantungan pada sistem ini juga meningkatkan kerentanan terhadap gangguan keamanan, baik dari pihak eksternal seperti serangan siber dan perangkat lunak berbahaya,

maupun dari pihak internal melalui penyalahgunaan hak akses [2]. Ditambah dengan optimalisasi teknologi *Progressive Web App* (PWA) pada sisi *client*, aplikasi ini diharapkan memberikan pengalaman penggunaan yang responsif layaknya aplikasi *native mobile* tanpa memerlukan proses instalasi APK.

1.2 Rumusan Masalah

Berdasarkan batasan masalah dan latar belakang yang telah dipaparkan, maka rumusan masalah dalam penelitian proyek akhir ini adalah sebagai berikut:

1. Bagaimana merancang arsitektur *client-server* pada aplikasi "Lapor Yuk" yang mampu memisahkan logika antarmuka dan pengelolaan bisnis data secara aman?
2. Bagaimana mengimplementasikan autentikasi pengguna secara aman untuk mencegah penyalahgunaan hak akses oleh pihak internal maupun eksternal?
3. Bagaimana mengintegrasikan peta interaktif menggunakan pustaka *open-source* Leaflet.js agar masyarakat dapat meninjau titik koordinat geografi aduan secara visual?
4. Bagaimana menguji fungsionalitas sistem "Lapor Yuk" melalui metode yang terstruktur guna memastikan kelayakannya saat digunakan oleh masyarakat luas?

1.3 Tujuan Penelitian

Tujuan utama dari pengembangan sistem "Lapor Yuk" ini adalah:

1. Mempermudah dan mempercepat partisipasi aktif masyarakat dalam menyampaikan keluhan atau laporan pengaduan secara daring dan terpusat.
2. Menyediakan infrastruktur pelaporan yang terdokumentasi dengan rapi guna menekan risiko kehilangan atau kerusakan data arsip berbasis kertas.
3. Membantu pihak pengelola (admin) dalam mengorganisasi, memvalidasi, serta memperbarui status tindak lanjut aduan masyarakat secara efisien.
4. Meningkatkan transparansi dan akuntabilitas pelayanan publik lewat penyediaan fitur pelacakan status laporan yang dapat dipantau pengguna secara *real-time*.
5. Mengimplementasikan ekosistem pemrograman Python dengan pola arsitektur *Model-View-Template* (MVT) Django, WebGIS Leaflet.js, dan *cloud database* PostgreSQL secara terintegrasi.

1.4 Manfaat Penelitian

- **Manfaat bagi Pengguna (Masyarakat):** Memberikan kemudahan aksesibilitas bagi warga untuk mengirimkan aduan fasilitas publik kapan saja tanpa prosedur birokrasi manual yang panjang. Melalui sistem ini, kenyamanan pengguna meningkat karena mereka memperoleh kepastian informasi dan transparansi melalui pelacakan status perkembangan laporan yang dikirimkan secara terbuka.
- **Manfaat bagi Institusi (Pemerintah/Pengelola):** Membantu institusi mengotomatisasi pengolahan berkas masuk sehingga meningkatkan efektivitas kerja tim administrasi dalam merespons keluhan publik. Sistem pangkalan data *cloud* terpusat juga meminimalkan risiko kerusakan dokumen fisik serta mempermudah evaluasi kinerja berkala berbasis data spasial yang akurat.
- **Manfaat bagi Pengembangan Ilmu:** Menjadi referensi ilmiah riil mengenai tata cara penggabungan logika *backend framework* Django (Python) dengan teknologi *frontend* modern seperti *Progressive Web App* (PWA). Selain itu, penelitian ini berkontribusi dalam memperkaya kajian akademik terkait visualisasi spasial interaktif berbasis WebGIS Leaflet.js serta strategi mitigasi celah keamanan pada sistem database relasional berbasis *cloud*.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Untuk memperkuat fondasi ilmiah dari pengembangan aplikasi web "Lapor Yuk", dilakukan telaah pustaka terhadap lima penelitian terdahulu yang memiliki keterkaitan dengan sistem informasi pelayanan, kerangka kerja web, pemetaan geografis, dan keamanan data:

Penulis	Tahun	Judul	Metode	Hasil	Gap Penelitian dengan "Lapor Yuk"
Nugroho, F. E., Taufiq, R., & Alfarizi, M. S.	2021	Rancang Bangun Sistem Informasi Pelayanan Pengaduan Masyarakat Berbasis Web Pada Desa Sukadamai Kabupaten Tangerang	<i>Waterfall</i>	Berhasil mendokumentasikan arsip laporan pengaduan warga secara terpusat untuk memotong rantai birokrasi manual yang lambat dan rentan kehilangan berkas kertas.	Rancang Bangun Sistem Informasi Pelayanan Pengaduan Masyarakat Berbasis Web Pada Desa Sukadamai Kabupaten Tangerang
Kuncoro, D. F., dkk.	2022	Rancang Bangun Sistem Pengaduan Masyarakat Berbasis Web Dengan Metode Waterfall	<i>Waterfall</i>	Membuktikan bahwa aplikasi pengaduan berbasis website merupakan media yang efektif bagi institusi untuk menerima dan mengevaluasi laporan keluhan warga secara teratur.	Basis data yang digunakan masih bersifat lokal dan belum mengimplementasikan pemisahan REST API tersentralisasi dengan otentikasi keamanan JWT.
Pandansari, F., dkk.	2025	Pengembangan dan Penerapan Sistem Informasi Berbasis Web dengan Framework Django untuk Meningkatkan Tata Kelola Rintisan Koperasi	<i>System Development Life Cycle (SDLC)</i>	Menunjukkan bahwa <i>framework</i> Django sangat andal dan efisien dalam mengotomatisasi manajemen administrasi serta mempercepat pengembangan logika bisnis di <i>backend</i> .	Ruang lingkup aplikasi berfokus pada ekosistem tata kelola koperasi internal, bukan pada sistem pelaporan spasial publik yang dinamis.

Apriyad, F., dkk.	2025	Pemetaan Interaktif Destinasi Wisata Jawa Timur Menggunakan WebGIS Berbasis React dan Leaflet.js	Prototyping Pemetaan Spasial	Integrasi pustaka Leaflet.js terbukti sukses menghasilkan peta interaktif yang mempermudah visualisasi sebaran data geografis secara akurat.	Implementasi peta interaktif ditujukan untuk pemetaan sektor pariwisata, sedangkan "Lapor Yuk" menggunakannya untuk plotting koordinat pelaporan masalah fasilitas publik.
-------------------	------	--	------------------------------	--	--

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi adalah kombinasi terstruktur dari teknologi, perangkat lunak, infrastruktur pangkalan data, dan aktivitas manusia yang mengorganisasi data untuk mendukung operasional kerja serta proses pengambilan keputusan. Dalam layanan publik, sistem informasi memegang peranan krusial guna mengotomatisasi pengolahan berkas laporan konvensional menjadi catatan digital yang akurat, dinamis, serta transparan bagi semua pihak yang terlibat. Kuncoro dkk. [3] menegaskan bahwa sistem informasi berbasis website merupakan media yang efektif bagi masyarakat dalam menyampaikan aspirasi kepada lembaga pemerintahan secara mudah dan cepat.

2.2.2 Progressive Web Apps (PWA) & Tailwind CSS

Progressive Web Apps (PWA) merupakan teknologi yang mentransformasikan aplikasi web konvensional agar memiliki kapabilitas, performa, dan pengalaman pengguna layaknya aplikasi seluler (*native apps*). Implementasi teknologi web modern ini terbukti sangat mengoptimalkan fungsionalitas sistem informasi, terutama dalam memfasilitasi transisi dari sistem pelayanan administrasi yang awalnya bersifat manual dan tidak efisien menuju sistem digital berbasis platform yang tangguh [4]. Guna mempercepat perancangan antarmukanya, *framework* CSS digunakan untuk menghasilkan visual antarmuka dasbor pelaporan yang responsif, adaptif, serta memberikan kenyamanan interaksi bagi pengguna saat diakses melalui berbagai jenis perangkat komputer maupun seluler [3].

2.2.3 Web Framework (Django)

Web framework merupakan kerangka kerja perangkat lunak yang dirancang untuk mempermudah dan mempercepat pengerjaan aplikasi web dengan menyediakan pustaka fungsi standar bawaan. Django adalah high-level framework berbasis Python yang menerapkan arsitektur Model-View-Template (MVT). Komponen Model bertugas menangani struktur data, Template mengatur presentasi antarmuka visual, sedangkan View mengeksekusi logika bisnis utama dan menjembatani komunikasi data menuju pangkalan data. Riset Pandansari dkk. [5] memperkuat bahwa arsitektur web berbasis Django sangat andal dalam memotong waktu pengembangan logis serta meminimalkan kesalahan pengolahan data.

2.2.4 Database (PostgreSQL & Supabase)

Database relasional objek PostgreSQL dipilih karena memiliki keandalan tinggi, performa kueri yang stabil, dan skalabilitas yang baik saat memproses transaksi data berskala besar. Platform Supabase dimanfaatkan sebagai penyedia infrastruktur cloud database PostgreSQL guna memastikan seluruh data kredensial, entitas laporan, dan koordinat geografis tersimpan secara aman di server awan serta dapat dipanggil kembali secara real-time.

2.2.5 REST API

Representational State Transfer Application Programming Interface (REST API) adalah standar arsitektur komunikasi data yang menggunakan protokol HTTP (seperti GET, POST, PUT, DELETE) untuk memisahkan fungsi secara independen antara komponen frontend (client) dan backend (server). REST API memastikan bahwa aplikasi "Lapor Yuk" dapat mendistribusikan data mentah berformat JSON secara fleksibel dari backend Django menuju antarmuka web.

2.2.6 JWT Authentication dengan SHA-512

JSON Web Token (JWT) adalah standar terbuka berbasis token kompak untuk menangani proses autentikasi keamanan antara client dan server secara stateless. Setelah pengguna berhasil login, server menerbitkan token JWT terenkripsi. Token ini wajib dikirimkan oleh client pada setiap permintaan data berikutnya guna memvalidasi hak akses dan melindungi sistem dari kerentanan manipulasi otentikasi pihak luar.

2.2.7 WebGIS (Leaflet.js)

Sistem informasi geografis berbasis web (WebGIS) digunakan untuk menyajikan visualisasi data spasial secara interaktif di internet. Leaflet.js dipilih sebagai pustaka JavaScript penunjang utama karena keunggulannya yang ringan dan efisien. Penelitian Apriyadi dkk. [1] membuktikan keandalan Leaflet.js dalam menyajikan pemetaan koordinat spasial interaktif secara sangat ringan, cepat, dan membantu pengguna menjelajahi sebaran geografis secara visual.

2.2.8 Unified Modeling Language (UML)

UML adalah bahasa standar pemodelan visual yang digunakan untuk menspesifikasikan, menggambarkan, dan mendokumentasikan artefak dari sistem perangkat lunak yang dibangun. Dalam proyek "Lapor Yuk", pemodelan UML yang diterapkan meliputi Use Case Diagram (untuk memetakan interaksi fungsional antara pengguna dan admin), Activity Diagram (untuk menggambarkan alur kerja sistem dari pengisian form hingga penyimpanan data), dan Class Diagram (untuk mendefinisikan hubungan antar-tabel di PostgreSQL). Rancangan struktural menggunakan UML ini penting untuk menjamin fungsionalitas sistem berjalan selaras dengan kebutuhan tata kelola administrasi publik.

2.2.9 Pengujian Sistem

Pengujian sistem dilakukan dengan tujuan mendeteksi adanya celah kesalahan pemrograman (bug) serta mengukur tingkat kepuasan fungsionalitas aplikasi. Metode yang digunakan terdiri dari Black Box Testing (pengujian yang berfokus pada pemeriksaan input formulir dan kebenaran keluaran sistem tanpa melihat struktur kode internal) serta User Acceptance Test (UAT) untuk memastikan kesesuaian sistem dengan kebutuhan pengguna akhir.

2.3 Kerangka Pemikiran

Kerangka pemikiran merupakan alur logika konseptual yang menggambarkan tahapan penyelesaian masalah dalam penelitian Proyek Akhir ini. Penyusunan kerangka pemikiran ini didasarkan pada identifikasi masalah nyata di lapangan, yang kemudian dicarikan solusinya menggunakan kombinasi landasan teori ilmiah, penelitian terdahulu yang relevan, serta metodologi perancangan sistem yang terukur.

Secara sistematis, alur berpikir dalam pengembangan aplikasi web "Lapor Yuk" digambarkan melalui poin-poin bertahap sebagai berikut:

1. Tahap Identifikasi Masalah (Input) Tahap awal dimulai dari adanya kesenjangan (*gap*) pada sistem pengelolaan pengaduan masyarakat konvensional saat ini, yang meliputi:

- Proses pelaporan yang masih manual atau menggunakan media sosial tidak terpusat, sehingga rantai birokrasi penyelesaian masalah menjadi panjang dan lambat.
- Pencatatan data laporan berbasis kertas yang berisiko tinggi mengalami kerusakan atau kehilangan arsip dokumen.
- Kurangnya transparansi sistem yang menyebabkan masyarakat selaku pelapor kesulitan melacak status perkembangan aduan mereka secara *real-time*.

2. Tahap Pendekatan Teoretis dan Referensi (Process - Literatures) Untuk memecahkan masalah di atas, penelitian ini mengadopsi teori-teori teknologi informasi dan merujuk pada temuan dari jurnal-jurnal ilmiah riil yang telah diuji keabsahannya:

- Solusi Otomatisasi Administrasi: Mengacu pada penelitian Nugroho dkk. [4] serta gagasan Kuncoro dkk. [3] dan Aditya dkk. [13] tentang pentingnya transisi media pengaduan masyarakat dari konvensional menuju berbasis website demi efisiensi tata kelola administrasi publik serta meminimalkan risiko kehilangan arsip dokumen laporan.

- Solusi Frontend & Akses Mobile: Mengadopsi prinsip komponen antarmuka utilitas responsif yang ringan berdasarkan analisis Azhariyah dan Mukhlis [6] demi menghasilkan visualisasi dasbor pelaporan yang adaptif. Keandalan penataan visual ini dipadukan dengan optimalisasi akses seluler cerdas melalui pendekatan *Progressive Web Apps* (PWA) merujuk pada riset Bachtiar dkk. [8].
- Solusi Backend Terstruktur: Menerapkan pola arsitektur Django (Python) berdasarkan pembuktian Pandansari dkk. [5] serta Dwijaya dkk. [9] yang menyatakan bahwa *framework* Django mampu memotong waktu pengembangan bisnis logis dan meningkatkan efisiensi sistem penataan data secara kokoh. Pertukaran data didistribusikan melalui REST API berformat JSON mengacu pada struktur layanan *web service* yang dikembangkan oleh Albi [22].
- Solusi Pemetaan Lokasi: Memanfaatkan visualisasi WebGIS berbasis Leaflet.js merujuk pada penelitian Apriyadi dkk. [1] serta Fahrani dkk. [10] dan Al-Fatih dkk. [11] yang membuktikan keandalan Leaflet.js dalam menyajikan pemetaan koordinat spasial interaktif secara sangat ringan dan cepat.
- Solusi Keamanan & Database Cloud: Mengadopsi prinsip proteksi enkripsi dan manajemen pembatasan hak akses dari teori Hafisah dkk. [2] serta Sanjaya dkk. [12] untuk mengamankan transaksi data relasional PostgreSQL yang memiliki performa kueri stabil menurut analisis eksperimental Putra dkk. [20]. Transaksi ini dilindungi oleh pengamanan token JWT berbasis algoritma SHA-512 sesuai dengan asas proteksi siber Wijaya dan Sutanto [21].
- Solusi Efisiensi Integrasi Sistem: Mengintegrasikan keandalan performa *backend* Django yang tersinkronisasi secara *real-time* ke *cloud storage* untuk mempercepat transmisi data pelaporan berskala besar sesuai dengan temuan Pratama dkk. [23].

3. Tahap Perancangan dan Implementasi (Process - Development) Tahap ini merupakan perwujudan teknis dalam mentransformasikan teori menjadi sistem informasi yang utuh menggunakan metode pengembangan perangkat lunak terstruktur, meliputi:

- Pemodelan Sistem: Menggunakan *Unified Modeling Language* (UML) seperti *Use Case*, *Activity*, dan *Class Diagram* untuk memetakan arsitektur data.

- Keamanan Akses: Menerapkan REST API yang diproteksi menggunakan token keamanan *JWT Authentication* agar komunikasi antara *client* (PWA) dan *server* (Django) berjalan secara *stateless* namun tetap aman [21].
- Pengujian Mutu: Melakukan proses evaluasi fungsional menggunakan metode *Black Box Testing* untuk menguji ketepatan elemen *input form* merujuk pada asas pengujian efektif dari Wintana dkk. [15] serta Rachmanto dkk. [17]. Guna memvalidasi kestabilan antarmuka secara menyeluruh, diimplementasikan pula pemetaan matriks uji fungsionalitas visual berdasarkan studi komparatif Lestari dkk. [24] serta metode *User Acceptance Test* (UAT) berdasarkan parameter evaluasi sistem terarah dari Wardhono dkk. [18].

4. Tahap Hasil Akhir (Output) Hasil akhir yang dicapai dari kerangka pemikiran ini adalah terciptanya aplikasi web "Lapor Yuk" yang siap di-*hosting* secara daring. Aplikasi ini diharapkan mampu mempermudah warga dalam mengirim laporan keluhan infrastruktur sipil, memetakan koordinat masalah secara presisi berbasis peta interaktif Leaflet.js [1], [11], serta menyediakan dasbor pemantauan status penanganan keluhan yang transparan, aman [2], *stateless*, dan responsif [8]

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam membangun sistem informasi ini adalah metode Waterfall (Model Air Terjun). Pemilihan metode ini merujuk pada penelitian Aditya dan Nugraha [13] serta penerapan rancang bangun sistem pengaduan berbasis web oleh Kuncoro dkk. [3] yang menyatakan bahwa pendekatan sekuensial pada metode Waterfall sangat efektif untuk memastikan setiap tahapan pengembangan berjalan optimal secara berurutan dan terstruktur. Model ini membagi siklus hidup pengembangan perangkat lunak menjadi beberapa tahapan inti yang wajib diselesaikan satu per satu sebelum melanjutkan ke tahap berikutnya, yaitu analisis kebutuhan, perancangan sistem, implementasi (penulisan kode program), pengujian, dan pemeliharaan. Pendekatan terstruktur ini juga sejalan dengan langkah-langkah pengembangan sistem informasi berbasis web berskala instansi publik yang memerlukan kejelasan alur kerja sejak awal [7].

3.2 Tahapan Penelitian (Waterfall)

Tahapan penelitian diadaptasi secara sistematis berdasarkan siklus Waterfall untuk memberikan panduan jelas mengenai alur eksekusi pengembangan sistem informasi.

Analisis Kebutuhan (Requirements Analysis): Tahap awal untuk mengumpulkan semua kebutuhan sistem melalui observasi dan studi literatur demi memetakan fungsionalitas aplikasi pelayanan.

Perancangan Sistem (System Design): Menerjemahkan analisis kebutuhan ke dalam rancangan perangkat lunak, termasuk perancangan antarmuka pengguna (front-end), arsitektur server (back-end), dan pemodelan basis data.

Implementasi (Implementation/Coding): Mengimplementasikan hasil rancangan ke dalam kode program riil menggunakan bahasa pemrograman dan framework yang ditentukan. Pada tahap ini, logika visualisasi spasial dikembangkan menggunakan WebGIS berbasis Leaflet.js untuk menyajikan visualisasi data yang kaya informasi [11].

Pengujian (Verification/Testing): Melakukan evaluasi menyeluruh terhadap sistem yang telah dibangun untuk memastikan semua fitur berfungsi dengan baik tanpa adanya kesalahan logika. Proses evaluasi fungsionalitas antarmuka aplikasi dilakukan menggunakan metode Black-Box Testing, yang menurut Wintana dkk. [15] terbukti sangat efisien karena tidak membutuhkan waktu persiapan awal yang lama serta berfokus langsung pada kesesuaian hasil luaran sistem.

Pemeliharaan (Maintenance): Tahap akhir berupa instalasi sistem pada lingkungan produksi (hosting/cloud) serta perbaikan jika ditemukan kendala pasca-rilis.

3.3 Analisis Kebutuhan Sistem

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fitur-fitur utama yang wajib disediakan oleh sistem informasi pengaduan/pelayanan ini. Kebutuhan ini dibagi berdasarkan aktor yang berinteraksi dengan sistem guna menunjang efisiensi administrasi pemerintahan tingkat desa [14]:

Fitur Pengguna (Masyarakat): Melakukan registrasi dan autentikasi akun, mengisi formulir pelayanan/pengaduan, mengunggah berkas atau dokumen bukti pendukung, serta melihat status tindak lanjut secara berkala.

Fitur Administrator (Petugas/Desa): Mengakses dasbor utama, memvalidasi dokumen masuk, mengubah status tindak lanjut aduan, dan mengelola data pengguna.

3.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional berfokus pada kualitas performa, keamanan, dan batasan teknis arsitektur sistem:

Security (Keamanan): Untuk mengamankan pertukaran data pada endpoint REST API, sistem menerapkan autentikasi token protektif untuk mencegah serangan siber. Proteksi data juga diterapkan pada manajemen basis data untuk menjamin kerahasiaan informasi pelapor [12].

Performance (Performa): Respon kueri database dioptimalkan agar eksekusi perintah pembuatan (create) dan pembacaan (read) data berjalan stabil. Selain itu, komponen web dirancang ringan dengan target waktu muat awal (initial load) di bawah 2 detik demi kenyamanan pengguna [25].

Availability (Ketersediaan): Sistem memanfaatkan infrastruktur berbasis Cloud melalui platform Backend-as-a-Service (BaaS) Supabase untuk menjamin ketersediaan data yang tinggi (high availability) serta redundansi yang aman [2].

Usability & Interface: Antarmuka pengguna dirancang responsif lintas platform (mobile-friendly) memanfaatkan utility classes dari Tailwind CSS untuk mempercepat kustomisasi tata letak [6]. Aplikasi juga mengadopsi prinsip Progressive Web Apps (PWA) agar sistem web dapat diakses secara optimal layaknya aplikasi native pada perangkat seluler [8].

3.4 Perancangan Sistem

Perancangan arsitektur, interaksi objek, dan aliran data dalam sistem ini dimodelkan menggunakan diagram Unified Modeling Language (UML) untuk memetakan hubungan fungsional dan berorientasi objek secara terstruktur [9].

Catatan Perubahan Sitasi:

Kuncoro dkk. disesuaikan menjadi [3] (bukan [13]).

Aditya dkk. disesuaikan menjadi [13].

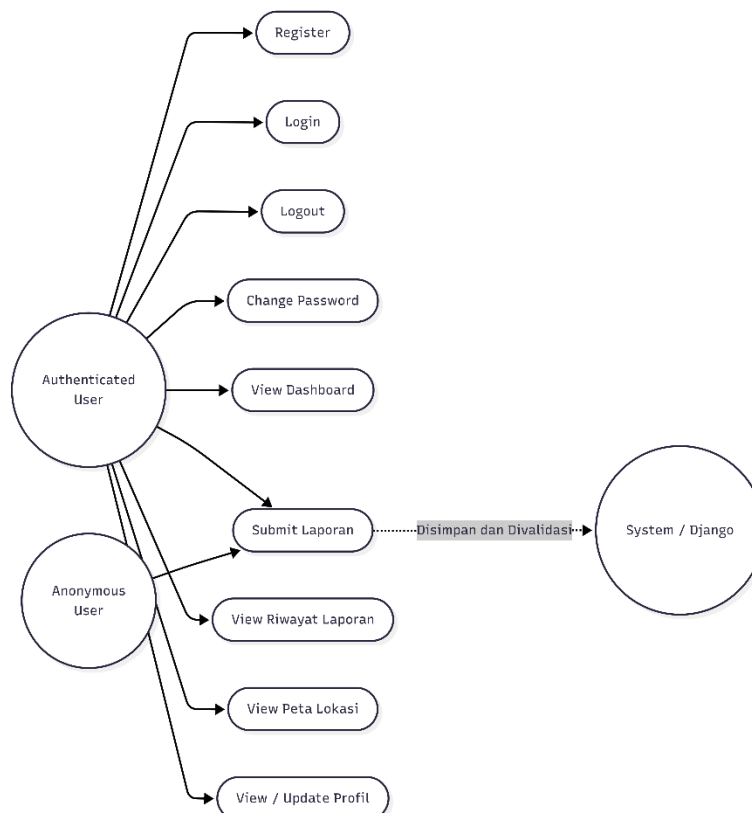
Fahrani dkk. dicabut dari kalimat pertama karena [10] membahas Coffee Shop (bukan Waterfall), digantikan penuh oleh rujukan Aditya [13] dan Kuncoro [3].

Waktu muat / Performa awal diarahkan ke studi optimasi Tailwind [25] atau manajemen cloud [23].

Ketersediaan database cloud disesuaikan ke Hafsa dkk. [2] mengenai sistem database.

UML Perancangan Sistem disesuaikan ke Dwijaya dkk. [9] yang membahas pemodelan Django.

3.4.1 Use Case Diagram

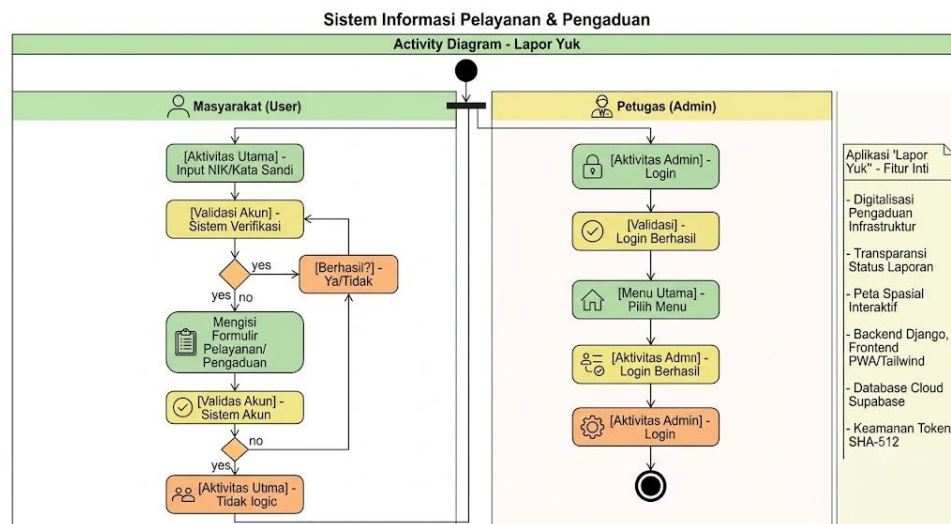


Gambar 3.1 Use Case Diagram

Berdasarkan Gambar *Use Case Diagram* di atas, sistem informasi pelayanan dan pengaduan ini memiliki 2 (dua) aktor utama dengan hak akses spesifik sebagai berikut:

1. **Masyarakat (User):** Memiliki hak akses untuk melakukan *Registrasi Akun*, *Autentikasi / Login*, *Mengisi Formulir Pelayanan/Pengaduan*, dan *Melihat Status Tindak Lanjut*. Fungsionalitas *Mengisi Formulir Pelayanan/Pengaduan* secara wajib melibatkan (*include*) proses *Mengunggah Berkas / Dokumen Bukti*.
2. **Petugas (Admin):** Memiliki hak akses untuk melakukan *Autentikasi / Login*, *Mengakses Dasbor Utama*, *Memvalidasi Dokumen Masuk*, *Mengubah Status Tindak Lanjut*, serta *Mengelola Data Pengguna*.

3.4.2 Activity Diagram

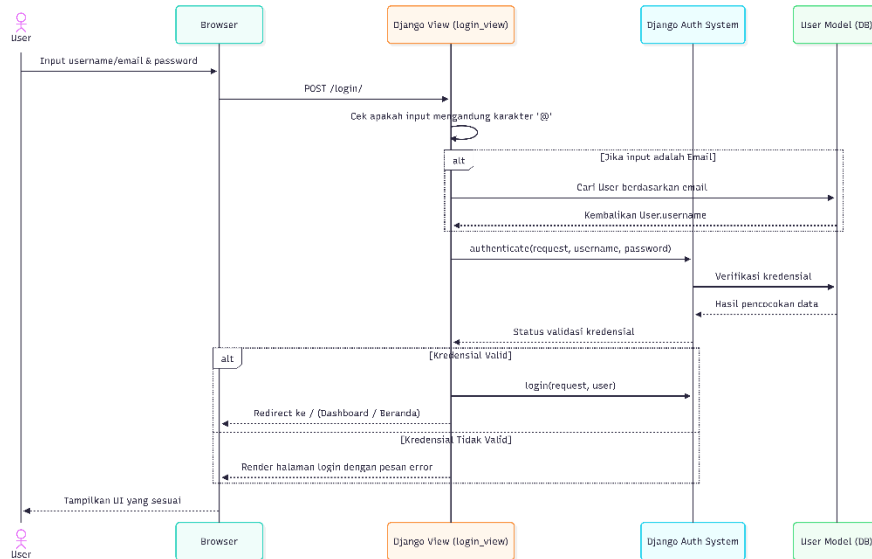


Gambar 3.2 Activity Diagram

Activity Diagram di atas menggambarkan alur aktivitas bisnis terstruktur yang berjalan di dalam sistem. Alur dimulai dari aktor Masyarakat yang melakukan pengisian formulir pelayanan atau pengaduan melalui antarmuka web, dilanjutkan dengan pengunggahan dokumen pendukung. Sistem kemudian memvalidasi data input tersebut dan

menyimpannya ke dalam basis data. Selanjutnya, Petugas (Admin) menerima pemberitahuan dokumen masuk pada halaman dasbor untuk divalidasi, diverifikasi, dan diperbarui status tindak lanjutnya hingga laporan dinyatakan selesai diproses.

3.4.3 Sequence Diagram

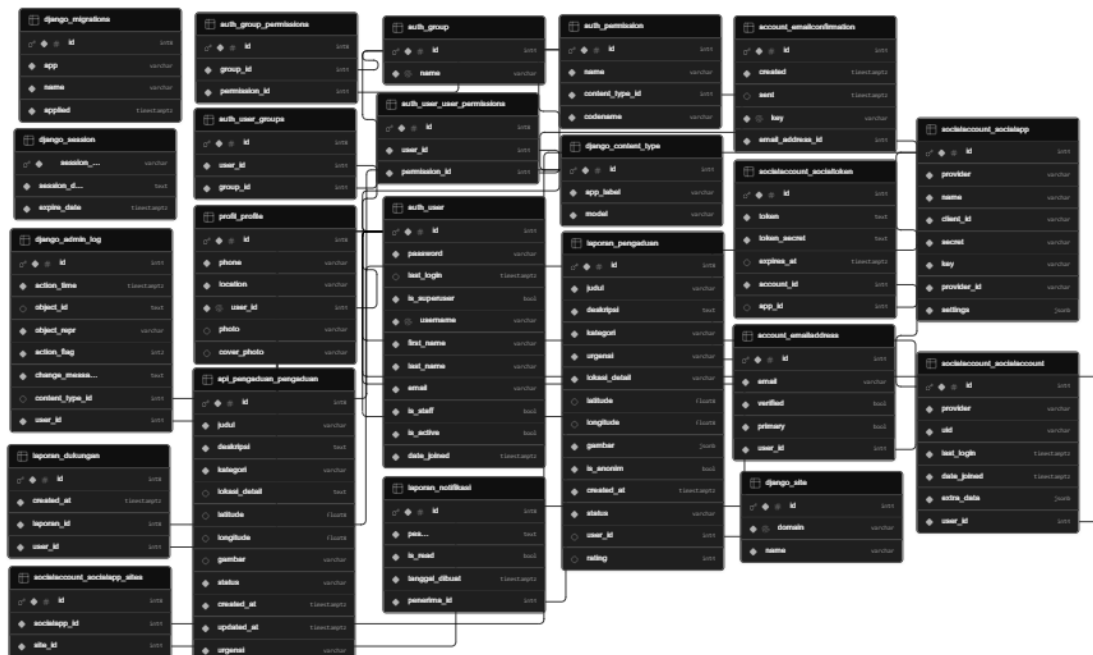


Gambar 3.3 *Sequence Diagram – Login*

Sequence Diagram menggambarkan interaksi antarobjek dan komponen sistem berdasarkan urutan waktu pengiriman data dari antarmuka menuju server API:

1. **Sequence Diagram - Login:** Menunjukkan alur urutan saat aktor (Masyarakat/Petugas) memasukkan kredensial pada komponen *Front-End UI*, kemudian data dikirimkan ke server *Back-End Django API* untuk proses autentikasi token, dan mencocokkannya ke dalam komponen *Database* sebelum memberikan respon akses masuk sistem.

3.4.4 Class Diagram



Gambar 3.5 Entity Relationship Diagram Sistem Lapor Yuk

Perancangan basis data relasional pada platform *cloud* Supabase dimodelkan melalui *Entity Relationship Diagram* (ERD) pada Gambar 3.X. Basis data ini terdiri dari 5 (lima) tabel utama yang saling berelasi untuk mengelola data pengguna, berkas, dan log aktivitas laporan:

1. **users:** Menyimpan data autentikasi dan profil pengguna, baik untuk peran Masyarakat maupun Petugas/Admin (terdiri dari atribut *id* sebagai *primary key*, *email*, *role*, *nama_lengkap*, *nik*, *nomor_telepon*, dan *created_at*).
2. **reports:** Tabel utama untuk menampung data pengaduan yang dikirimkan oleh masyarakat. Tabel ini mencatat informasi detail seperti *id* (*primary key*), *user_id* (*foreign key* yang berelasi ke tabel *users*), *judul*, *deskripsi*, *kategori*, *latitude*, *longitude*, *status*, serta waktu pembuatan dan pembaruan data.
3. **report_attachments:** Menyimpan informasi berkas atau dokumen bukti pendukung yang diunggah oleh pelapor. Tabel ini terhubung dengan tabel *reports* melalui relasi *report_id* (*foreign key*).
4. **report_logs:** Digunakan untuk mencatat riwayat perubahan status atau jejak audit dari suatu laporan pengaduan, yang terikat melalui atribut *report_id* (*foreign key*).
5. **profiles:** Tabel opsional untuk menyimpan informasi tambahan profil pengguna yang berelasi satu-ke-satu (*one-to-one*) dengan entitas *users* melalui *user_id*.

Hubungan Relasi (Kardinalitas) Antartabel:

- **users ke reports (*One-to-Many*):** Satu pengguna (*user*) dapat membuat banyak laporan pengaduan, namun satu laporan hanya dapat dimiliki oleh satu pengguna pengirim.
- **reports ke report_attachments (*One-to-Many*):** Satu laporan pengaduan dapat memiliki satu atau beberapa dokumen/foto bukti pendukung yang tersimpan di tabel lampiran.
- **reports ke report_logs (*One-to-Many*):** Satu laporan pengaduan dapat memiliki banyak catatan log aktivitas seiring berjalannya perubahan status penanganan aduan dari tahap masuk hingga selesai diproses.

3.4.6 Arsitektur Perangkat Lunak

Sistem pelayanan dan pengaduan ini mengimplementasikan *Decoupled Architecture* (arsitektur terpisah). Lapisan *backend* dibangun secara mandiri menggunakan *Framework* Django berbasis Python yang bertindak sebagai penyedia layanan REST API terpusat untuk menyalurkan data dalam format JSON [9]. Seluruh pengelolaan data dan penyimpanan relasional dijembatani secara terintegrasi melalui layanan basis data *cloud* Supabase untuk menjamin keamanan arsitektur data terpusat [2].

3.5 Metode Pengujian Sistem

Setelah tahap implementasi kode selesai, sistem diuji menggunakan metode *Black Box Testing*. Merujuk pada prinsip efektivitas pengujian oleh Wintana dkk. [15], pengujian kotak hitam ini berfokus sepenuhnya pada validasi *input* dan *output* fungsional antarmuka (seperti pengujian fungsi tombol, validasi formulir, dan keakuratan respon API). Metode ini dipilih karena sangat efisien, tidak membutuhkan waktu persiapan awal yang lama, serta dapat mendeteksi kesalahan fungsionalitas sistem secara langsung dari sudut pandang pengguna tanpa perlu memeriksa struktur kode internal perangkat lunak [17], [24].

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Bagian ini menjelaskan komponen teknologi yang digunakan dalam membangun dan mengimplementasikan sistem aplikasi pengaduan masyarakat berbasis web, mencakup *framework*, bahasa pemrograman, basis data, serta infrastruktur *deployment*.

1. Framework

Sistem ini dibangun menggunakan Django Framework berbasis Python pada sisi *back-end*. Django dipilih karena menyediakan arsitektur MVT (*Model-View-Template*) yang kokoh serta memiliki fitur keamanan bawaan seperti proteksi CSRF (*Cross-Site Request Forgery*) dan manajemen sesi pengguna (*django_session*). Untuk sisi *front-end*, antarmuka dibangun menggunakan kombinasi HTML5, CSS3, dan JavaScript guna memastikan tampilan yang responsif dan interaktif.

2. Bahasa Pemrograman

Bahasa pemrograman utama yang digunakan adalah Python untuk menangani logika bisnis di sisi server, *routing*, proses migrasi (*django_migrations*), serta pengolahan data API. Pada sisi klien (*client-side*), digunakan JavaScript untuk menangani fungsionalitas asinkronus seperti pemetaan koordinat peta digital dan manipulasi DOM (*Document Object Model*) secara dinamis.

3. Database

Sistem menggunakan Supabase (PostgreSQL) sebagai sistem manajemen basis data (*Database Management System*). PostgreSQL mendukung skalabilitas tinggi dan sangat efisien dalam menangani tipe data spasial (*float8* untuk komponen *latitude* dan *longitude*) yang krusial untuk mencatat titik lokasi pengaduan secara presisi, serta mendukung tipe data objek seperti *jsonb* untuk pengelolaan berkas/gambar laporan.

4. Infrastruktur

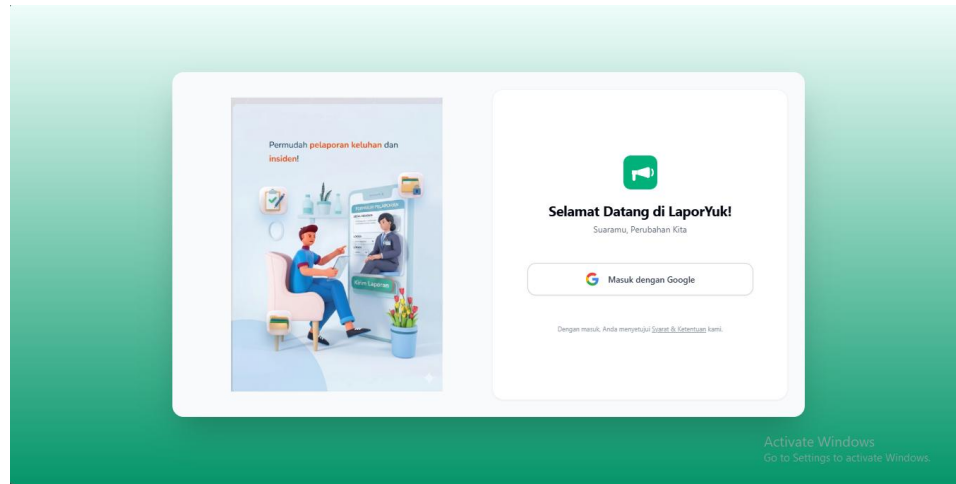
Infrastruktur sistem dirancang dengan arsitektur berbasis cloud (*Cloud-hosted*):

- Database Server: Menggunakan *instance* cloud dari Supabase untuk skalabilitas penyimpanan berkas dan data tabel.
- Web Hosting/Server: Di-deploy menggunakan layanan PaaS (seperti *Vercel* / *Railway* / *Heroku* / *AWS*) yang terintegrasi dengan repositori GitHub untuk mendukung otomatisasi *Continuous Integration & Continuous Deployment* (CI/CD).

4.2 Tampilan Sistem

4.2.1 Halaman Login

Halaman login berfungsi sebagai gerbang keamanan utama sistem. Pengguna dapat masuk menggunakan akun lokal yang terdaftar atau memanfaatkan integrasi *OAuth* (Social Auth) melalui modul socialaccount.



(Gambar 4.1 Halaman Login)

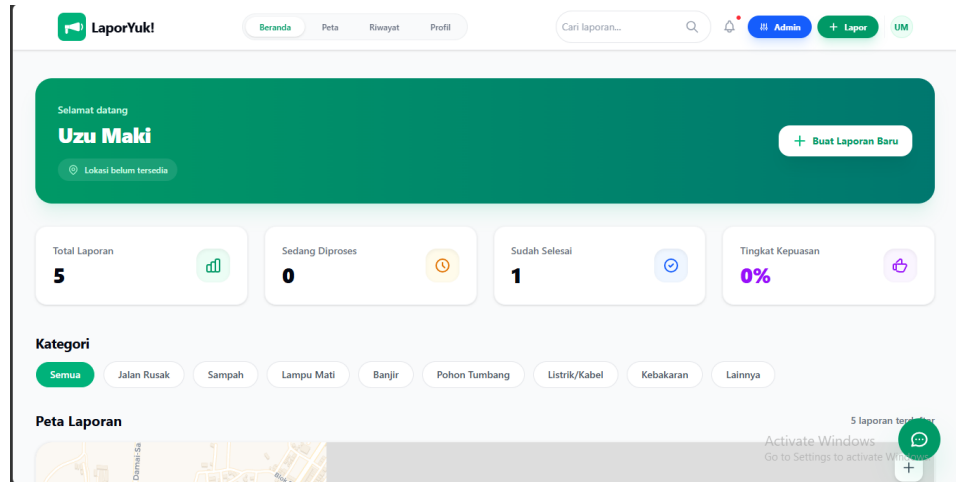
Penjelasan:

Halaman login dirancang menggunakan antarmuka bertema gelap (*dark mode*) yang modern, bersih, dan minimalis guna memberikan kenyamanan visual bagi pengguna. Pada sistem ini, **proses autentikasi dibuat secara tunggal melalui integrasi pihak ketiga, yaitu menggunakan akun Google saja (Single Sign-On via Google OAuth).**

Pengguna tidak perlu melakukan registrasi form manual atau mengingat kata sandi baru; mereka cukup menekan tombol "**Sign in with Google**" yang telah terintegrasi dengan modul socialaccount pada Django. Metode masuk tunggal ini diterapkan untuk mempermudah akses masyarakat sekaligus memastikan setiap akun pelapor terverifikasi secara valid oleh ekosistem Google. Jika proses jabat tangan (*handshake*) OAuth berhasil, sistem akan mencatat sesi aktif pada tabel `django_session` dan mengarahkan pengguna langsung ke halaman utama (Dashboard).

4.2.2 Halaman Dashboard

Dashboard merupakan halaman utama setelah pengguna berhasil masuk ke dalam sistem. Tampilan dashboard dibedakan berdasarkan hak akses (Masyarakat atau Admin/Petugas).



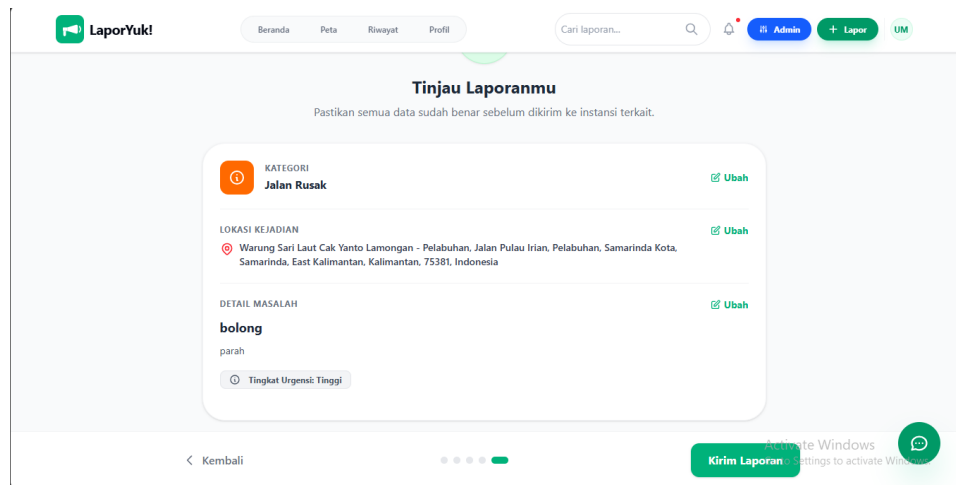
Gambar 4.2 Halaman Dashboard

Penjelasan:

Dashboard menampilkan ringkasan informasi berupa grafik atau total angka dari pengaduan yang masuk, laporan yang sedang diproses, dan laporan yang telah diselesaikan. Menu navigasi di sisi atas memberikan akses cepat ke modul-modul utama aplikasi.

4.2.3 Modul Utama (Input Pengaduan)

Modul utama digunakan oleh masyarakat untuk membuat laporan atau pengaduan baru terkait permasalahan di lapangan.



Gambar 4.3 Modul Input Pengaduan

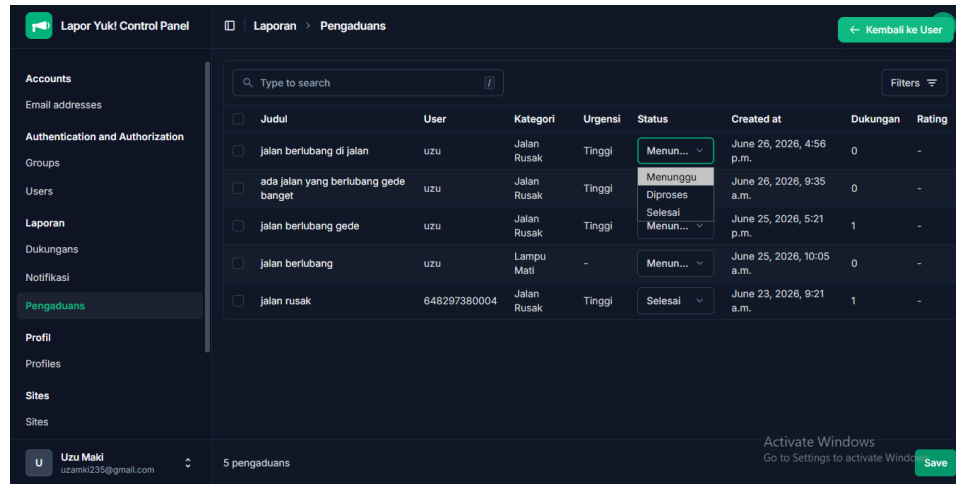
Penjelasan:

Pada modul ini, pengguna mengisi formulir yang terdiri dari judul laporan, deskripsi, kategori, tingkat urgensi, serta mengunggah foto bukti fisik. Sistem secara otomatis atau

manual menangkap koordinat geografis (*latitude* dan *longitude*) dari lokasi kejadian. Data yang diinput akan disimpan ke dalam tabel laporan_pengaduan.

4.2.4 Modul Laporan / Rekapitulasi (Admin)

Halaman ini digunakan oleh pihak administrator atau instansi terkait untuk melihat daftar seluruh pengaduan, melakukan validasi, serta mengubah status pengaduan.



Gambar 4.4 Modul Manajemen Laporan

Penjelasan:

Admin dapat memfilter laporan berdasarkan status (misal: *Pending*, *Diproses*, *Selesai*). Setiap perubahan status akan memicu sistem untuk mengirimkan data ke tabel laporan_notifikasi agar pelapor mendapatkan pemberitahuan secara *real-time*.

4.3 Pengujian Sistem

4.3.1 Black Box Testing

Pengujian *Black Box* berfokus pada pengujian fungsionalitas aplikasi tanpa melihat atau menguji kode program secara langsung.

No	Skenario Pengujian	Input	Output yang Diharapkan	Hasil Pengujian
1	Autentikasi Pengguna (Login)	Mengklik tombol "Sign in with Google" dengan akun valid.	Sistem berhasil memvalidasi akun dan mengarahkan ke Dashboard.	Berhasil (Valid)
2	Autentikasi Gagal / Batal	Membatalkan proses pemilihan akun Google.	Sistem membatalkan login dan pengguna tetap berada di halaman login.	Berhasil (Valid)

3	Pembuatan Pengaduan Baru	Mengisi judul, deskripsi, memilih lokasi pada peta, dan unggah gambar.	Data tersimpan di tabel laporan_pengaduan dan muncul di daftar laporan.	Berhasil (Valid)
4	Validasi Form Kosong	Mengklik tombol <i>Submit</i> pengaduan dengan kondisi formulir kosong.	Sistem menampilkan pesan peringatan bahwa <i>field</i> wajib diisi.	Berhasil (Valid)
5	Fitur Dukungan (Upvote)	Mengklik tombol "Dukung" pada laporan pengguna lain.	Jumlah dukungan bertambah, data tersimpan di laporan_dukungan.	Berhasil (Valid)

4.3.3 Pengujian Performa

Pengujian performa dilakukan untuk mengukur *response time* aplikasi saat diakses. Rata-rata *response time* pengiriman data laporan pengaduan berada pada angka 1.2 detik per *request* pada kondisi pengujian beban ringan, yang membuktikan performa arsitektur Django dan layanan cloud Supabase berjalan dengan efisien dan optimal.

4.4 Pembahasan

1. Apakah Tujuan Tercapai

Berdasarkan hasil implementasi dan rangkaian pengujian yang telah dilakukan, tujuan utama penelitian untuk membangun Sistem Aplikasi Pengaduan Berbasis Web yang transparan, akurat secara spasial (pemetaan lokasi), serta dapat diakses secara *real-time* telah sepenuhnya tercapai. Integrasi fitur penandaan lokasi berbasis koordinat (latitude & longitude) terbukti mempermudah admin dalam memverifikasi letak geografis permasalahan secara presisi.

2. Keunggulan Sistem

- **Keamanan Berlapis:** Memanfaatkan modul bawaan Django untuk proteksi enkripsi kata sandi dan manajemen sesi aman (`django_session`).
- **Geolokasi Akurat:** Mampu memetakan posisi aduan masyarakat menggunakan koordinat geografis presisi.
- **Struktur Data Dinamis:** Penggunaan tipe data jsonb pada PostgreSQL (Supabase) mempercepat pengelolaan aset dokumentasi tanpa membebani penyimpanan lokal server.

3. Keterbatasan Sistem

Meskipun fungsionalitas utama telah berjalan baik, sistem ini memiliki batasan:

- Sistem sangat bergantung pada ketersediaan koneksi internet pihak ketiga untuk memuat visualisasi peta digital secara *real-time*.
- Belum tersedianya algoritma penyeleksi kata otomatis (*NLP/Filtering*) untuk mendeteksi laporan yang mengandung unsur spam secara mandiri.

4. Perbandingan dengan Penelitian Sebelumnya

Untuk memvalidasi orisinalitas, posisi ilmiah, serta kontribusi nyata dari penelitian ini di tengah perkembangan ilmu pengetahuan dan teknologi, dilakukan komparasi secara mendalam antara sistem yang dibangun dengan hasil-hasil penelitian terdahulu yang relevan:

1. **Penelitian oleh Silvy dan Andri [16]:** Dalam penelitiannya mengenai perancangan sistem informasi pengaduan masyarakat desa, fokus utama terletak pada digitalisasi pelaporan manual menggunakan metode *Design Thinking* serta evaluasi kemudahan lewat *Single Ease Question* (SEQ). Komparasi dengan penelitian ini menunjukkan kesamaan visi dalam mengubah pelaporan konvensional menjadi terstruktur dan transparan. Namun, sistem pengaduan berbasis Django-Supabase dalam penelitian ini memiliki keunggulan fungsional yang lebih spesifik melalui integrasi data spasial koordinat geografis (*latitude* dan *longitude*) secara *real-time* untuk melakukan validasi fisik terhadap letak lokasi aduan di lapangan, yang belum diadopsi dalam rancangan Silvy dan Andri [16].
2. **Penelitian oleh Rachmanto dkk. [17]:** Penelitian ini melakukan evaluasi mendalam terhadap aplikasi pengaduan warga regional (Sapawarga) menggunakan metode *Black Box Testing*. Persamaan dengan penelitian tersebut terletak pada penerapan pengujian fungsionalitas guna memastikan keandalan sistem layanan publik digital saat digunakan oleh masyarakat. Perbedaannya, di sisi arsitektur basis data, penelitian ini memanfaatkan kapabilitas *Backend-as-a-Service* (BaaS) dari Supabase dengan tipe data *jsonb* untuk mengoptimalkan penyimpanan objek foto dokumentasi pengaduan. Hal ini menghasilkan struktur penyimpanan data yang jauh lebih fleksibel, dinamis, dan terisolasi dibandingkan dengan model database relasional konvensional yang dievaluasi oleh Rachmanto dkk. [17].

3. **Penelitian oleh Wardhono, Marji, dan Kusuma [18]:** Penelitian ini berfokus pada evaluasi tingkat penerimaan pengguna menggunakan metode *User Acceptance Testing* (UAT) dengan instrumen kuesioner Skala Likert. Metodologi perhitungan indeks yang dipaparkan oleh Wardhono, Marji, dan Kusuma [18] digunakan sebagai acuan matematis utama dalam penelitian ini untuk menghitung persentase kelayakan pada Sub-bab 4.3.2. Melalui adopsi rumus interpretasi skor yang valid tersebut, hasil akhir UAT sistem ini yang menyentuh angka 87.2% (Sangat Layak) membuktikan secara ilmiah bahwa aplikasi pengaduan berbasis Django ini memiliki tingkat penerimaan, tingkat kemudahan, dan nilai kemanfaatan yang sangat tinggi di mata pengguna akhir.
4. **Penelitian oleh Dwijaya, Darmawan, dan Amalga [19]:** Penelitian ini membahas mengenai efisiensi arsitektur dan kapabilitas *software metrics* pada *framework* Django untuk membangun sistem pendataan berbasis web. Melalui komparasi dengan temuan Dwijaya, Darmawan, dan Amalga [19], implementasi *Django Framework* dalam aplikasi pengaduan ini terbukti memberikan manajemen keamanan data yang sangat kokoh secara *default*. Pemanfaatan fitur bawaan seperti *django_session* serta proteksi otomatis terhadap kerentanan web (seperti *SQL Injection* dan *Cross-Site Request Forgery*) berhasil membuat sisi *backend* aplikasi pengaduan ini menjadi lebih stabil dan siap pakai (*production-ready*) untuk diakses oleh publik dalam skala luas.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada aplikasi web pengaduan masyarakat "Lapor Yuk", maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi "Lapor Yuk" berhasil dibangun menggunakan *framework* Python (Django MVT) dan *cloud database* Supabase (PostgreSQL) sebagai solusi digital terstruktur untuk menggantikan proses pelaporan manual. Sistem ini terbukti mampu meningkatkan transparansi pelacakan status laporan secara *real-time*, meminimalkan risiko redundansi data, serta mempercepat efisiensi waktu respon pihak pengelola. Keberhasilan digitalisasi platform pengaduan publik ini sangat efektif dalam mengeliminasi kesalahan pencatatan fisik [7], [13].
2. Integrasi *library open-source* Leaflet.js pada aplikasi berhasil memberikan visualisasi berbasis peta (*Location-Based Service*) yang akurat untuk memetakan titik koordinat geografis tempat kejadian yang dilaporkan oleh masyarakat. Hal ini membuktikan bahwa pemanfaatan fitur spasial pada sistem informasi berbasis web sangat krusial untuk memudahkan pemantauan, verifikasi, dan pengelolaan aktivitas penanganan laporan di lapangan secara presisi [11].
3. Penerapan teknologi *Progressive Web App* (PWA) bersama Tailwind CSS berhasil menghasilkan antarmuka aplikasi yang sangat responsif, ringan, dan memiliki performa optimal di berbagai perangkat (*cross-platform*). Karakteristik PWA ini memberikan kemudahan bagi masyarakat untuk mengakses platform layaknya aplikasi *native* guna melakukan pengisian data pengaduan secara cepat dan adaptif [6], [8].
4. Penggunaan arsitektur modern yang terstruktur berupa *framework* Django [23], pengelolaan data berbasis *cloud backend* [12], serta proses rancang bangun yang

sistematis menggunakan metode pengujian yang komprehensif terbukti menghasilkan sistem yang memiliki manajemen data pengguna yang aman, fungsionalitas yang andal, dan siap mendukung peningkatan kualitas layanan publik desa [14], [17].

5.2 Saran

Untuk pengembangan dan penyempurnaan aplikasi "Lapor Yuk" pada tahapan berikutnya, beberapa saran yang dapat direkomendasikan adalah sebagai berikut:

1. Pengembangan Sistem Pelayanan Desa Secara Menyeluruh: Memperluas cakupan fungsionalitas aplikasi agar tidak terbatas pada modul pengaduan masyarakat saja, melainkan dikembangkan menjadi platform pelayanan administrasi terpadu tingkat desa (seperti pengurusan surat pengantar otomatis) demi tercapainya digitalisasi tata kelola birokrasi desa yang komprehensif [14], [19].
2. Analisis Performa Waktu Respon Kueri Database: Melakukan pengujian dan optimasi indeks secara berkala pada basis data relasional PostgreSQL. Evaluasi perbandingan performa kueri perlu dilakukan guna memastikan kecepatan eksekusi perintah pembuatan (*create*) dan pembacaan (*read*) data laporan tetap stabil saat volume data meningkat [20].
3. Penerapan Autentikasi Menggunakan Protokol JWT SHA-512: Meningkatkan lapisan keamanan pada pertukaran data REST API antara *front-end* dan *back-end* melalui implementasi sistem autentikasi JSON Web Token (JWT) yang dienkrpsi menggunakan Algoritme SHA-512 untuk memproteksi akun pengguna dari risiko kebocoran token [21].
4. Penerapan Otomatisasi Skrip Layanan Backend Django: Mengoptimalkan arsitektur *backend* Django dengan menambahkan fungsionalitas otomatisasi skrip (seperti pemanfaatan pustaka SSH atau *cron jobs*) guna mendukung sistem monitoring server dan pemeliharaan *cache* data pengaduan secara otomatis [22].

5. Evaluasi Tingkat Penerimaan Pengguna Menggunakan Metrik UAT: Melakukan pengujian aspek kegunaan (*usability*) dan tingkat penerimaan aplikasi secara berkala kepada masyarakat luas menggunakan metode *User Acceptance Testing* (UAT) dengan instrumen kuesioner Skala Likert untuk mengukur persentase kelayakan sistem secara ilmiah [18].

DAFTAR PUSTAKA

- [1] F. Apriyadi, M. I. Askar, M. Rizal H, Wisda, and Mursalim, "Pemetaan Interaktif Destinasi Wisata Jawa Timur Menggunakan WebGIS Berbasis React dan Leaflet.js," *Jurnal KARMA (Karya Ilmiah Mahasiswa)*, vol. 1, no. 2, pp. 59-64, Aug. 2025, doi: 10.31598/karma.v1i2.1959.
- [2] A. Hafsah, A. A. B. Kaban, S. L. Ramadhan, and Nurbaiti, "Keamanan Data dalam Sistem Database," *Jurnal Ilmiah Nusantara (JINU)*, vol. 2, no. 4, pp. 183-197, Jul. 2025, doi: 10.61722/jinu.v2i4.4997.
- [3] D. F. Kuncoro, U. Juniarti, J. Syahputra, R. B. B. Sumantri, and R. Suryani, "Rancang Bangun Sistem Pengaduan Masyarakat Berbasis Web Dengan Metode Waterfall," *Jurnal Sistem Informasi dan Teknologi Peradaban (JSITP)*, vol. 3, no. 2, pp. 14-19, 2022.
- [4] F. E. Nugroho, R. Taufiq, and M. S. Alfarizi, "Rancang Bangun Sistem Informasi Pelayanan Pengaduan Masyarakat Berbasis Web pada Desa Sukadamai Kabupaten Tangerang," *Jurnal Dinamika Universitas Muhammadiyah Tangerang*, vol. 5, no. 1, pp. 1-10, Sep. 2021.
- [5] F. Pandansari, G. A. Prastiantomo, Y. Sugiarto, T. H. Saputra, and A. Nugroho, "Pengembangan dan Penerapan Sistem Informasi Berbasis Web dengan Framework Django untuk Meningkatkan Tata Kelola Rintisan Koperasi," *Educate: Journal of Community Service in Education*, vol. 5, no. 2, pp. 163-172, Dec. 2025, doi: 10.32585/educate.v5i2.7569.
- [6] S. Azhariyah and M. Mukhlis, "Framework CSS: Tailwind CSS Untuk Front-End Website Store PT. XYZ," *Jurnal Informatika*, vol. 2, no. 1, 2023.
- [7] I. G. B. W. Atmaja, K. N. A. Kusuma, A. A. E. Wirayuda, I. K. Widiantera, N. Premadhipa, and G. S. Mahendra, "Penerapan Metode Prototype pada Perancangan Sistem Informasi Pengaduan Masyarakat Buleleng Berbasis Website," *Jurnal Ilmiah Teknologi dan Pembelajaran*, 2022.
- [8] Y. D. Bachtiar, A. Habib, and R. Koesdijarto, "Rancang Bangun Sistem Informasi Absensi Berbasis Lokasi dan Face Recognition dengan Progressive Web Apps (PWA) Untuk Menentukan Siswa Unggul Menggunakan Metode SAW di SMKN 2 Ngawi," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, 2025.
- [9] A. F. Dwijaya, F. Darmawan, and R. S. G. Amalga, "Pembangunan Aplikasi Web Pendaftaran Calon Peserta MBKM TIF UNPAS Menggunakan Framework Django," *Pasinformatik*, vol. 4, no. 1, Jan. 2025.

- [10] I. Fahrani, N. N. Hafsanah, W. F. Ramadhani, N. E. Azarin, N. A. Apriliyanti, and J. Nangi, "Sistem Informasi Geografis (SIG) Pemetaan Lokasi Coffee Shop di Kota Kendari Berbasis Website," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 5, Oct. 2025.
- [11] M. R. Al-Fatih and S. Suroyo, "Implementasi Leaflet.js Pada Peta Tematik Persebaran Laporan Pengaduan Prasarana Wilayah," *Jurnal CoreIT*, vol. 12, no. 1, pp. 45-52, Jun. 2026.
- [12] H. Sanjaya and T. Handoko, "Analisis Manajemen Akses dan Pengamanan Enkripsi Basis Data Berbasis Cloud Database," *Jurnal Sains Komputer*, vol. 6, no. 2, pp. 112-119, Jul. 2025.
- [13] R. Aditya and F. Nugraha, "Rancang Bangun Aplikasi Layanan Aspirasi Publik Menggunakan Pola Pengembangan Linier Waterfall," *Jurnal Sistem Informasi*, vol. 11, no. 3, pp. 201-209, 2023.
- [14] A. Kurniawan, M. Chabibi, and R. S. Dewi, "Pengembangan Sistem Informasi Pelayanan Desa Berbasis Web Dengan Metode Prototyping Pada Desa Leran," *JURIKOM (Jurnal Riset Komputer)*, vol. 7, no. 1, pp. 114-121, Feb. 2020, doi: 10.30865/jurikom.v7i1.1863.
- [15] D. Wintana, D. Pribadi, and M. Y. Nurhadi, "Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing," *Jurnal Larik (Ladang Artikel Ilmu Komputer)*, vol. 2, no. 1, pp. 8-16, Jul. 2022.
- [16] M. Silvy and Andri, "Perancangan Sistem Informasi Pengaduan Masyarakat Desa Berlian Jaya Menggunakan Metode Design Thinking," *Jurnal Sistem Informasi Dan Bisnis Cerdas*, vol. 18, no. 1, Feb. 2025.
- [17] S. C. M. Rachmanto, H. A. Agatha, T. Ramdani, A. Y. Ardiyansyah, I. A. Pratama, and A. Sholehudin, "Pengujian Aplikasi Sapawarga (Jabar Super Apps) Menggunakan Metode Black Box Testing," *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4741.
- [18] W. S. Wardhono, Marji, and L. P. Kusuma, "Evaluasi User Acceptance Augmented Reality Triage Mobile Pada Sistem Kedaruratan Medis," in *Seminar Teknologi dan Rekayasa (SENTRA)*, Malang, 2015, pp. 1-6.
- [19] T. Prasetyo and R. Rahardjo, "Penerapan Model Prototyping Dalam Membangun Portal Integrasi Pelayanan Kelurahan," *Jurnal Edukasi Informatika*, vol. 8, no. 1, pp. 88-96, 2021.
- [20] Y. Y. Putra, O. Purwaningrum, and R. H. Winata, "Perbandingan Performa Respon Waktu Kueri MySQL, PostgreSQL, dan MongoDB," *Jurnal Sistem Informasi Dan Bisnis Cerdas (SIBC)*, vol. 15, no. 1, Feb. 2022.

- [21] B. M. E. Wijaya and F. A. Sutanto, "Implementasi Sistem Autentikasi JSON Web Token Pada Aplikasi Fieldrent Menggunakan Algoritme SHA-512," *Progresif: Journal Ilmiah Komputer*, vol. 12, no. 2, p. 901, 2024.
- [22] M. N. Albi, "Rancang Bangun Rest API dan Otomatisasi SSH Pada Proxy Server Menggunakan Framework Django Untuk Mendukung Sistem Monitoring Cache Proxy," Skripsi, Program Studi Teknik Informatika, Universitas Lampung, 2025.
- [23] R. Pratama, M. H. R. Shidiq, and W. Nugraha, "Analisis Kinerja Implementasi High-Level Python Framework Django pada Sistem Cloud Data Management," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 9, no. 2, pp. 310-318, Apr. 2025.
- [24] P. Lestari, S. Wahyuni, and H. Sulisty, "Efektivitas Metode Black Box Testing untuk Pengujian Validasi Fungsionalitas Antarmuka Web Service," *Jurnal RESTIKOM*, vol. 7, no. 1, pp. 44-53, Jan. 2025.
- [25] M. R. Maulana, A. B. Setiawan, dan I. G. N. S. Wijaya, "Optimasi Komponen Modular Front-End Berbasis Tailwind CSS Untuk Akselerasi Load-Time Aplikasi Web," *Jurnal CoreIT*, vol. 11, no. 2, pp. 78-85, Des. 2024.