

LAPORAN KERJA PRAKTIK

SISTEM INFORMASI PELAPORAN KEGIATAN PETUGAS

LAPANGAN DI DINAS LINGKUNGAN HIDUP

KOTA MEDAN BERBASIS WEB



Disusun oleh:
Dionikxon Manurung (228160012)

TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2025

LEMBAR PENGESAHAN

LAPORAN KERJA PRAKTEK SISTEM INFORMASI PELAPORAN KEGIATAN PETUGAS LAPANGAN DI DINAS LINGKUNGAN HIDUP KOTA MEDAN BERBASIS WEB

Diajukan Untuk Memenuhi Salah Satu Syarat Mata Kuliah Kerja Praktek

Jenjang Studi

S-1 Program Studi Teknik Informatika

Oleh :

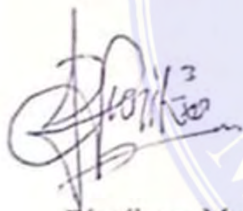
Dionikxon Manurung (228160012)

Medan, 14 Oktober 2025

Menyetujui

Dosen Pembimbing

Mahasiswa



Dionikxon Manurung
NPM:228160012



Susilawati S.Kom, M.Kom
NIDN 0126068702

Mengetahui,
Ketua Prodi Teknik Informatika



Rizki Mulliono S.Kom, M.Kom
NIDN 0109038902



UNIVERSITAS MEDAN AREA

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INFORMATIKA

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan, 20223
 Kampus II : Jalan Seliabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402564, Medan, 20122
 Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

BERITA ACARA DAN NILAI SEMINAR KERJA PRAKTEK

Pada hari ini 14 Oktober 2025 telah diselenggarakan Seminar Kerja Praktek Program Studi Teknik Informatika untuk Tahun Akademik 2025/2026 atas :

Nama : Dionikxon Manurung
 NIM : 228160012
 Program Studi : Teknik Informatika
 Jenjang Pendidikan : S1 (Sarjana)
 Judul Kerja Praktek : Sistem Informasi Pelaporan Kegiatan Petugas Lapangan di Dinas Lingkungan Hidup Kota Medan Berbasis Web
 Tempat Seminar : Ruang Seminar Fakultas Teknik
 Tanda Tangan Pembawa Seminar :
 Nilai Pembawa Seminar : 91,2 (A)

Seminar Kerja Praktek bersangkutan disetujui/tidak disetujui dengan catatan perubahan seperti yang tercantum pada tabel berikut :

Saran:	Susilawati, S.Kom, M.Kom Pembimbing Kerja Praktek
Tambahkan Status verifikasi laporan petugas	
Persetujuan Seminar:	
Saran:	Rizki Muliono S.Kom, M.Kom Ka. Prodi
Persetujuan Seminar:	

PANITIA SEMINAR KERJA PRAKTEK:

No.	Jabatan	Nama Dosen	Tanda Tangan
1	Pembimbing Kerja Praktek	Susilawati, S.Kom, M.Kom	1
2	Ka. Prodi	Rizki Muliono S.Kom, M.Kom	2

Medan, 14 Oktober 2025
 Ketua Prodi.



Rizki Muliono S.Kom, M.Kom



ABSTRAK

Perkembangan teknologi informasi menuntut Dinas Lingkungan Hidup Kota Medan mengoptimalkan proses kerja melalui sistem yang efisien dan transparan. Salah satu aspek penting dalam mendukung kinerja tersebut adalah melalui penggunaan sistem informasi pelaporan kegiatan petugas lapangan yang berfungsi sebagai dasar evaluasi, monitoring, serta pengambilan keputusan. Sebelumnya, proses pelaporan masih menggunakan dokumen cetak sehingga sering menimbulkan kendala, yaitu pengarsipan yang tidak terstruktur, keterlambatan verifikasi, hingga kurangnya transparansi dan akuntabilitas petugas dalam melaporkan hasil temuan. Untuk mengatasi permasalahan tersebut, kerja praktik ini merancang dan membangun sistem informasi pelaporan berbasis web dengan menerapkan metode pengembangan sistem, yang mencakup tahapan analisis kebutuhan, perancangan alur kerja dan basis data, implementasi kode menggunakan *framework* Django dan basis data *PostgreSQL* melalui layanan Supabase. Sistem ini dirancang dengan fitur utama berupa pembuatan laporan digital oleh petugas lapangan yang dihubungkan dengan GPS (*Global Positioning System*), verifikasi laporan oleh kepala bidang selaku atasan, manajemen tugas, pencatatan kehadiran, serta notifikasi secara real-time. Dengan demikian, sistem informasi pelaporan ini dapat menjadi solusi alternatif yang lebih efektif dibandingkan metode manual dalam mendukung tata kelola lingkungan hidup yang baik.

Kata Kunci: Sistem Pelaporan Kegiatan, Django, Supabase, Sistem Informasi, Dinas Lingkungan Hidup

ABSTRACT

The development of information technology demands the Environmental Agency of Medan City to optimize work processes through efficient and transparent systems. One important aspect in supporting this performance is the use of an information system for field officers' activity reporting, which serves as the basis for evaluation, monitoring, and decision-making. Previously, the reporting process relied on printed documents, which often led to several issues, such as unstructured archiving, delays in verification, and a lack of transparency and accountability in officers' reporting of findings. To address these problems, this internship project designs and develops a web-based reporting information system by applying a system development method, which includes the stages of requirements analysis, workflow and database design, and code implementation using the Django framework and PostgreSQL database by the Supabase service. The system is designed with core features such as digital report submission by field officers integrated with GPS (Global Positioning System), report verification by supervisors, task management, attendance tracking, and real-time notifications. Thus, this reporting information system can serve as a more effective alternative solution compared to manual methods in supporting good environmental governance.

Keyword: Reporting System, Django, Supabase, Information System, Department of Environment

KATA PENGANTAR

Segala puji dan rasa syukur dipanjatkan kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga saya selaku penulis dapat menyelesaikan laporan ini sebagai salah satu persyaratan dalam menyelesaikan Laporan Kerja Praktik.

Penyusunan laporan ini tidak lepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, saya ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc selaku Rektor Universitas Medan Area
2. Bapak Dr. Eng. Supriatno, ST., MT, selaku Dekan Fakultas Teknik Universitas Medan Area
3. Bapak Rizki Muliono, S.kom., M.kom., selaku Kepala Program Studi Teknik Informatika
4. Ibu Susilawati S.Kom., M.Kom., selaku dosen pembimbing kerja praktik.
5. Pegawai dan *Staff* Dinas Lingkungan Hidup Kota Medan yang sudah memberikan kesempatan untuk melakukan kerja praktik.

Saya menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa mendatang. Akhir kata, saya berharap laporan ini dapat memberikan manfaat, tidak hanya bagi penulis tetapi juga bagi pembaca dan pihak-pihak yang berkepentingan.

Medan, 14 Oktober 2025

Dionikxon Manurung

DAFTAR ISI

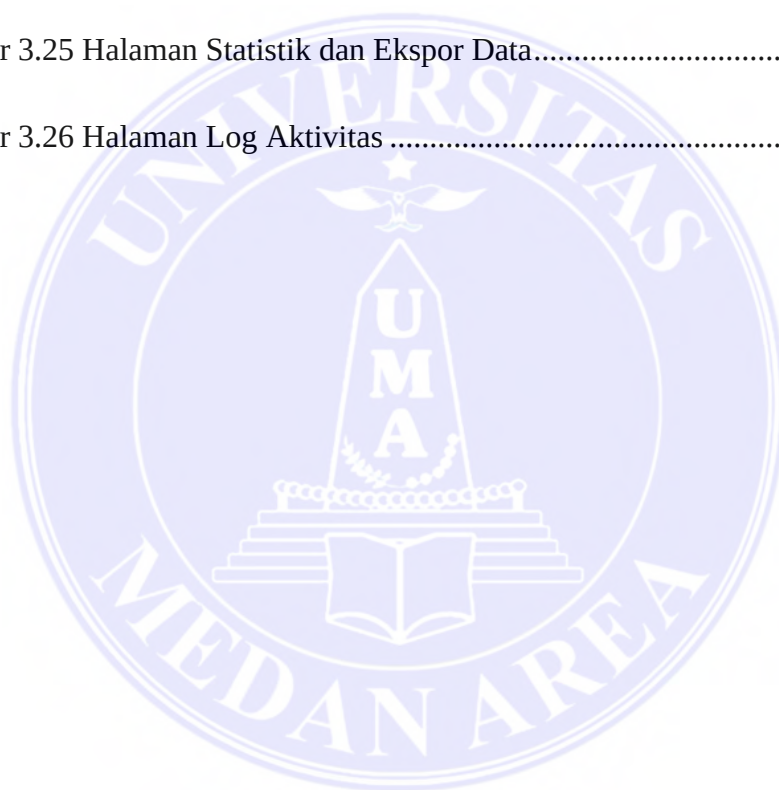
LEMBAR PENGESAHAN	i
BERITA ACARA	ii
ABSTRAK	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vii
DAFTAR TABEL	ix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan Kerja Praktik	2
1.4 Manfaat Kerja Praktik	2
1.5 Waktu dan Tempat Pelaksanaan Kerja Praktik	3
BAB II TINJAUAN TEORI.....	4
2.1 Dinas Lingkungan Hidup	4
2.2 Aplikasi dan Web	5
2.3 <i>Framework</i> Django.....	5
2.4 Supabase	6
2.5 <i>PostgreSQL</i>	6
2.6 <i>Flowchart</i>	7
2.7 UML	8

2.7.1 Use Case Diagram.....	9
2.7.2 Activity Diagram.....	10
2.7.3 Class Diagram.....	10
2.7.4 Sequence Diagram	12
BAB III PEMBAHASAN	14
3.1 Bentuk Kegiatan	14
3.2 Analisis Sistem yang Berjalan.....	14
3.3 Analisis Sistem yang Diusulkan.....	16
3.3.1 Flowchart Pelaporan Petugas	17
3.3.2 Use Case Diagram Pelaporan Petugas	19
3.3.3 Activity Diagram Pelaporan Petugas	20
3.3.4 Sequence Diagram Pelaporan Petugas	21
3.3.5 Entity Relationship Diagram (ERD) Pelaporan Petugas	22
3.4 Hasil Pengembangan Sistem	24
3.4.1 Fitur Umum Sistem Pelaporan Petugas	24
3.4.2 Hak Akses Petugas Lapangan.....	25
3.4.3 Hak Akses Atasan.....	30
3.4.4 Hak Akses Admin.....	33
BAB IV PENUTUP	38
4.1 Kesimpulan.....	38
4.2 Saran	38
DAFTAR PUSTAKA	39
LAMPIRAN.....	41

DAFTAR GAMBAR

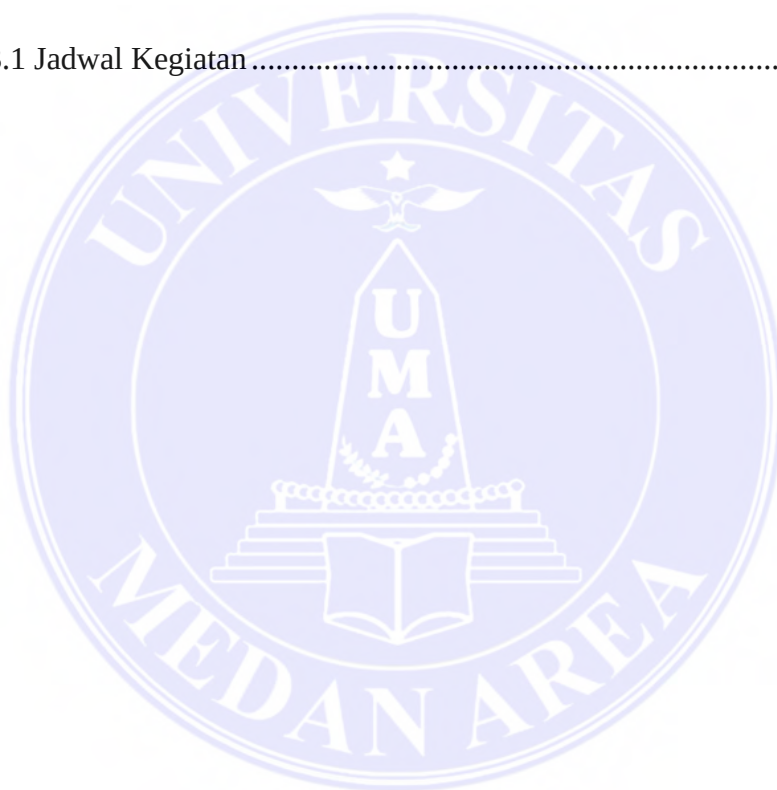
Gambar 1.1 Dinas Lingkungan Hidup Kota Medan	3
Gambar 2.1 Struktur Dinas Lingkungan Hidup Kota	4
Gambar 3.1 Diagram Analisis Sistem yang Sedang Berjalan.....	15
Gambar 3.2 <i>Flowchart</i> Sistem Pelaporan	18
Gambar 3.3 <i>Use Case Diagram</i> Sistem Pelaporan	20
Gambar 3.4 <i>Activity Diagram</i> Sistem Pelaporan	21
Gambar 3.5 Sequence Diagram Sistem Pelaporan.....	22
Gambar 3.6 ERD Sistem Pelaporan	23
Gambar 3.7 Halaman Login.....	24
Gambar 3.8 Halaman Profil	25
Gambar 3.9 Halaman Dashboard Petugas Lapangan.....	25
Gambar 3.10 Halaman Dashboard Petugas Lapangan.....	26
Gambar 3.11 Halaman Tugas Petugas Lapangan	27
Gambar 3.12 Halaman Detail Tugas Petugas Lapangan.....	28
Gambar 3.13 Halaman Buat Laporan Petugas Lapangan	28
Gambar 3.14 Halaman Riwayat Kehadiran Petugas Lapangan	29
Gambar 3.15 Halaman Dashboard Atasan.....	30
Gambar 3.16 Halaman Manajemen Tugas.....	31
Gambar 3.17 Halaman Detail dan Edit Tugas	31
Gambar 3.18 Halaman Monitoring Tim.....	32

Gambar 3.19 Halaman Laporan yang Dikirim Petugas	32
Gambar 3.20 Halaman Detail Laporan	33
Gambar 3.21 Halaman Dashboard Admin	34
Gambar 3.22 Halaman Manajemen Pengguna	34
Gambar 3.23 Halaman Panel Admin	35
Gambar 3.24 Halaman Manajemen Data	36
Gambar 3.25 Halaman Statistik dan Ekspor Data	37
Gambar 3.26 Halaman Log Aktivitas	37



DAFTAR TABEL

Tabel 2.1 Simbol dan Fungsi dalam <i>Flowchart</i>	7
Tabel 2.2 Simbol dan Keterangan <i>Use Case Diagram</i>	9
Tabel 2.3 Simbol dan Keterangan <i>Activity Diagram</i>	10
Tabel 2.4 Simbol dan Keterangan <i>Class Diagram</i>	11
Tabel 2.5 Simbol dan Keterangan <i>Sequence Diagram</i>	12
Tabel 3.1 Jadwal Kegiatan	14



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dinas Lingkungan Hidup (DLH) merupakan instansi pemerintah daerah yang memiliki tugas pokok dan fungsi dalam menjaga, mengelola, serta mengawasi kualitas lingkungan hidup. Dalam melaksanakan tugasnya, DLH tidak hanya menyusun kebijakan, tetapi juga melakukan kegiatan lapangan, seperti pengawasan pencemaran, pemantauan kualitas udara dan air, pengelolaan sampah, serta penegakan hukum lingkungan. Aktivitas lapangan ini memerlukan dokumentasi yang akurat sebagai bahan evaluasi dan dasar dalam pengambilan keputusan.

Pelaporan kegiatan lapangan Dinas Lingkungan Hidup masih dilakukan secara manual yaitu menggunakan dokumen cetak yang melalui rangkaian tahapan untuk sampai kepada atasan (*supervisor*), yakni Kepala Bidang Penataan sehingga menimbulkan keterlambatan pengumpulan laporan dan proses verifikasi, pengarsipan dokumen yang tidak terstruktur dan sulit untuk dikelola secara efisien, serta kurangnya transparansi laporan dan akuntabilitas petugas. Kondisi tersebut dapat menghambat efektivitas kinerja DLH Kota Medan dalam menjalankan fungsi pengawasan dan pengendalian lingkungan.

Seiring dengan perkembangan teknologi informasi, sistem berbasis *website* menjadi salah satu solusi yang efektif untuk meningkatkan efisiensi pelaporan (Arujisaputra, 2025). Sistem informasi pelaporan berbasis *website* memungkinkan petugas lapangan untuk langsung memasukkan data kegiatan ke dalam aplikasi, sehingga laporan dapat segera diverifikasi oleh kepala bidang dan diolah lebih lanjut menjadi rekapitulasi data yang terstruktur. Selain itu, penggunaan basis data terintegrasi seperti *PostgreSQL* juga mendukung manajemen data dalam jumlah besar dan memungkinkan integrasi dengan data spasial (*geo-tagging*) guna memvisualisasikan lokasi kegiatan (Surachkaryadi et al., 2025).

Pemanfaatan *framework* modern seperti *Django*, serta layanan *backend* as a *service* seperti Supabase, memberikan keunggulan dalam pengembangan sistem yang cepat, terukur, dan aman. Dengan demikian, penerapan sistem informasi pelaporan kegiatan lapangan di DLH diharapkan mampu meningkatkan kualitas pengelolaan lingkungan, mendukung prinsip tata kelola pemerintahan yang baik

(*good governance*), serta mendorong transparansi dan akuntabilitas kerja petugas lapangan (Yulianto, 2021).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam kerja praktik ini dapat dirumuskan sebagai berikut:

1. Bagaimana membangun suatu sistem informasi berbasis web yang dapat memudahkan pelaporan kegiatan petugas lapangan di Dinas Lingkungan Hidup secara efektif dan efisien?
2. Bagaimana sistem yang dikembangkan dapat mempermudah atasan dalam memverifikasi laporan, melakukan *monitoring*, serta mengevaluasi kinerja petugas lapangan?
3. Bagaimana penerapan basis data terintegrasi dengan memanfaatkan PostgreSQL dan layanan *backend as a service Supabase* dalam mendukung pengelolaan data laporan yang aman, terstruktur, dan mudah diakses?

1.3 Tujuan Kerja Praktik

Tujuan dari pelaksanaan kerja praktik ini adalah sebagai berikut:

1. Mengembangkan sistem informasi berbasis *website* yang dapat digunakan oleh petugas lapangan Dinas Lingkungan Hidup untuk melakukan pelaporan kegiatan secara cepat, terstruktur, dan terdokumentasi dengan baik.
2. Menyediakan fitur verifikasi laporan, *monitoring*, serta evaluasi kinerja petugas lapangan sehingga dapat membantu kepala bidang dalam pengambilan keputusan dan pengendalian kegiatan lapangan.
3. Menerapkan teknologi basis data *PostgreSQL* serta layanan *backend as a service Supabase* untuk mendukung pengelolaan data laporan yang aman, konsisten, dan mudah diakses kapanpun diperlukan.

1.4 Manfaat Kerja Praktik

Manfaat yang diharapkan dari pelaksanaan kerja praktik ini adalah sebagai berikut:

1. Bagi Instansi (Dinas Lingkungan Hidup):

Sistem informasi pelaporan dapat meningkatkan efektivitas pengawasan dan

kinerja lapangan, mempercepat proses verifikasi, serta mendukung transparansi dan akuntabilitas dalam pengelolaan lingkungan.

2. Bagi Petugas Lapangan:

Mempermudah proses pelaporan kegiatan dengan antarmuka yang sederhana, aksesibilitas tinggi, serta dukungan dokumentasi digital seperti foto, lokasi, dan catatan lapangan sehingga pekerjaan menjadi lebih efisien.

3. Bagi Atasan/Pengawas:

Memfasilitasi *monitoring real-time* terhadap kegiatan petugas, membantu dalam evaluasi kinerja, serta menyediakan dasar data yang kuat untuk pengambilan keputusan strategis.

4. Bagi Pengembang/Penulis:

Memberikan pengalaman praktis dalam menerapkan ilmu yang diperoleh selama perkuliahan, khususnya dalam bidang pengembangan sistem berbasis web, manajemen basis data, serta implementasi *framework* Django.

1.5 Waktu dan Tempat Pelaksanaan Kerja Praktik

Kerja praktik ini dilaksanakan dalam kurun waktu tertentu sesuai dengan jadwal yang telah ditetapkan oleh program studi. Kerja praktik dilaksanakan selama satu bulan, dimulai pada bulan 1 Agustus s/d 31 Agustus 2025. Selama periode tersebut, kegiatan berfokus pada analisis sistem yang berjalan, perancangan sistem yang diusulkan, serta pengembangan sistem informasi pelaporan kegiatan petugas lapangan. Kegiatan kerja praktik dilaksanakan di Dinas Lingkungan Hidup Kota Medan, yang berlokasi di Jl. Pinang Baris No.114, Lalang, Kec. Medan Sunggal, Kota Medan, Sumatera Utara.



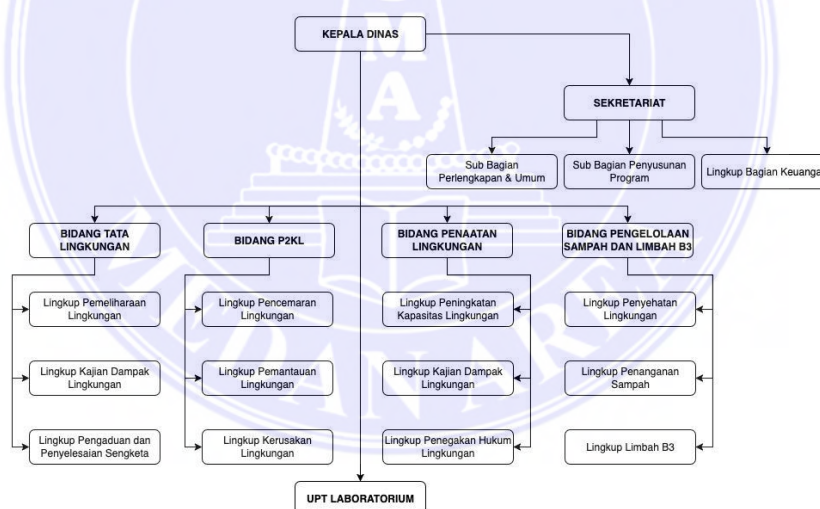
Gambar 1.1 Dinas Lingkungan Hidup Kota Medan

BAB II

TINJAUAN TEORI

2.1 Dinas Lingkungan Hidup

Dinas Lingkungan Hidup (DLH) Kota Medan merupakan instansi pemerintah daerah yang memiliki peran utama dalam mengelola urusan pemerintahan di sektor lingkungan hidup. DLH berkontribusi secara signifikan dalam menjaga kelestarian lingkungan, memastikan pengelolaan sumber daya alam yang berkelanjutan, serta memantau pelaksanaan kebijakan yang mendukung upaya pelestarian lingkungan di wilayah Kota Medan. Peran ini mencakup aspek perencanaan, pelaksanaan, hingga evaluasi program-program lingkungan hidup, baik yang bersifat preventif maupun represif terhadap potensi kerusakan lingkungan (Novia Gardini & Zulkarnaini Zulkarnaini, 2025).



Gambar 2.1 Struktur Dinas Lingkungan Hidup Kota

Struktur organisasi DLH Kota Medan terdiri dari kepala dinas, sekretariat, sejumlah bidang teknis, serta unit pelaksana teknis yang bekerja secara terkoordinasi dalam mewujudkan visi dan misi lembaga. Setiap bidang memiliki fokus tanggung jawab yang berbeda, seperti bidang penataan lingkungan, bidang pengendalian dan penegakan hukum lingkungan, bidang pengelolaan sampah dan limbah, serta bidang konservasi dan pengelolaan sumber daya alam.

2.2 Aplikasi dan Web

Aplikasi merupakan perangkat lunak yang dirancang untuk membantu pengguna dalam menyelesaikan suatu tugas atau kebutuhan tertentu melalui perangkat komputer maupun perangkat *mobile* (Daryono et al., 2024). Aplikasi dapat berbasis desktop, *mobile*, maupun web, tergantung pada kebutuhan dan lingkungan pengembangan yang digunakan.

Web adalah suatu sistem informasi berbasis internet yang memungkinkan pertukaran data dan interaksi antar pengguna melalui peramban (*browser*) (Febriyani & Martanto, 2023). Web bekerja dengan menggunakan protokol HTTP/HTTPS dan ditunjang oleh bahasa pemrograman seperti HTML, CSS, dan Python. Dalam konteks pengembangan sistem informasi modern, web menjadi media utama karena sifatnya yang *platform-independent* dan dapat diakses dari berbagai perangkat.

2.3 Framework Django

Django merupakan salah satu *framework* pengembangan aplikasi berbasis web yang menggunakan bahasa pemrograman Python. *Framework* ini dirancang untuk mempermudah pengembang dalam membangun aplikasi dengan menyediakan struktur yang rapi, fitur keamanan, serta dukungan integrasi dengan berbagai basis data. Django menggunakan pola arsitektur *Model-View-Template* (MVT) yang mirip dengan *Model-View-Controller* (MVC), sehingga memisahkan logika aplikasi, data, dan tampilan antarmuka pengguna secara lebih terstruktur (Anggarah et al., 2025)

Keunggulan utama Django terletak pada prinsip “*batteries included*” yakni sudah menyertakan banyak komponen bawaan yang mempermudah proses pengembangan, seperti sistem autentikasi, panel administrasi otomatis, serta manajemen basis data (Muhamad Givari Ramadan & Mardi Hardjianto, 2025). Selain itu, Django juga dikenal memiliki tingkat keamanan yang tinggi karena dilengkapi dengan perlindungan terhadap ancaman umum pada aplikasi web, seperti *SQL Injection*, *Cross Site Scripting* (XSS), dan *Cross Site Request Forgery*

(CSRF) (Muhamad Givari Ramadan & Mardi Hardjianto, 2025)

Dalam konteks pengembangan sistem informasi pelaporan kegiatan petugas lapangan di Dinas Lingkungan Hidup, Django sangat relevan digunakan karena dapat menghasilkan aplikasi yang skalabel, cepat dikembangkan, dan mudah dipelihara. Selain itu, Django mendukung integrasi dengan berbagai basis data seperti *PostgreSQL* yang sangat andal dalam pengelolaan data berskala besar.

2.4 Supabase

Supabase adalah sebuah platform *backend as a service* (BaaS) yang dirancang untuk mempermudah pengembangan aplikasi dengan menyediakan layanan manajemen basis data, autentikasi, penyimpanan *file*, serta *real-time* API (Rasyaad & Wibowo, 2025). Supabase dibangun di atas *PostgreSQL* sebagai inti pengelolaan basis datanya. Keunggulan Supabase adalah kemampuannya dalam menyediakan *auto-generated* API sehingga pengembang tidak perlu membuat API dasar secara manual, serta mendukung integrasi *real-time* yang memungkinkan data dapat diperbarui dan diakses secara langsung (Hadianto & Zubaidi, 2024).

2.5 PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional (RDBMS) *open-source* yang telah diakui secara global karena reputasinya dalam hal kestabilan, skalabilitas, dan kepatuhan terhadap standar SQL yang ketat. Keunggulan utama *PostgreSQL* terletak pada kemampuannya untuk menangani berbagai beban kerja, mulai dari aplikasi kecil hingga sistem *enterprise* yang kompleks dan membutuhkan kinerja tinggi. Selain fitur dasar pengelolaan data relasional, *PostgreSQL* juga menyediakan berbagai kemampuan tingkat lanjut, seperti *stored procedures* dan *triggers* yang memungkinkan otomatisasi proses bisnis secara efisien. Sistem ini mendukung indeksasi yang kompleks, termasuk indeks berbasis B-tree, hash, dan GiST, yang sangat berguna untuk meningkatkan performa pencarian dan *query*. *PostgreSQL* mendukung berbagai fitur tingkat lanjut, seperti *stored procedures*, *triggers*, indeks yang kompleks, serta dukungan untuk data

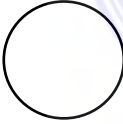
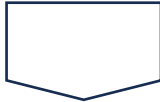
spasial dengan ekstensi PostGIS (Hartono, 2021)

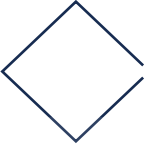
2.6 Flowchart

Flowchart adalah representasi grafis dari suatu algoritma, proses, atau alur kerja yang menggunakan simbol-simbol standar untuk menggambarkan langkah-langkah penyelesaian suatu masalah (Kus Indrani Listyoningrum et al., 2023). Dalam konteks pengembangan perangkat lunak, *flowchart* berfungsi untuk memberikan gambaran menyeluruh mengenai logika sistem, urutan kegiatan, serta hubungan antar proses sebelum diimplementasikan ke dalam kode program.

Flowchart bermanfaat untuk memudahkan komunikasi antara pengembang dan pengguna karena menyajikan alur kerja dalam bentuk visual yang sederhana dan mudah dipahami. Dengan demikian, proses analisis dan pengembangan sistem dapat dilakukan secara lebih sistematis dan terarah. Simbol yang digunakan untuk menggambarkan *flowchart* adalah sebagai berikut:

Tabel 2.1 Simbol dan Fungsi dalam Flowchart

Simbol	Fungsi	Simbol	Fungsi
	Ikon koneksi digunakan untuk masuk dan keluar atau menghubungkan prosedur ke halaman yang sama.		Simbol input-output: Simbol dapat menunjukkan proses input dan output perangkat apa pun.
	Simbol pemrosesan: Simbol ini menunjukkan bahwa komputer melakukan pemrosesan.		Seperti yang telah ditetapkan sebelumnya, simbol penilaian menunjukkan pelaksanaan suatu bagian atau prosedur.

	Simbol operasi manual: Ini menunjukkan bahwa komputer tidak melakukan pemrosesan.		Simbol punch card ditulis untuk menunjukkan bahwa input atau output berasal dari kartu.
	Simbol keputusan: Keadaan menentukan proses pengambilan keputusan.		Simbol Dokumen: Simbol yang menunjukkan apakah entri tersebut berasal dari buku yang dicetak atau dari buku yang ditulis.

2.7 UML

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan dalam rekayasa perangkat lunak untuk memvisualisasikan, merancang, serta mendokumentasikan sistem secara terstruktur (Perwitasari et al., 2024). UML menyediakan seperangkat notasi grafis yang dapat menggambarkan berbagai aspek dari sistem, mulai dari alur kerja, interaksi antaraktor, hingga struktur data dan hubungan antarobjek.

Secara umum, UML terbagi menjadi dua kelompok diagram utama, yaitu:

1. Diagram struktural, yang berfokus pada representasi statis dari sistem, misalnya *object diagram*, dan *component diagram*.
2. Diagram perilaku, yang menekankan pada interaksi dinamis serta alur aktivitas sistem, seperti *use case diagram*, *activity diagram*, dan *sequence diagram*.

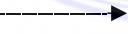
Dalam laporan ini, beberapa jenis diagram UML yang digunakan antara lain *use case diagram*, *activity diagram* dan *sequence diagram*. Diagram-diagram tersebut dipilih karena mampu memberikan gambaran yang menyeluruh, baik dari segi kebutuhan fungsional pengguna, alur proses bisnis, struktur sistem, hingga interaksi antar objek dalam sistem.

2.7.1 Use Case Diagram

Use Case Diagram merupakan salah satu jenis diagram perilaku dalam UML yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem (Kurniawan et al., 2020). Diagram ini berfungsi untuk menunjukkan kebutuhan fungsional sistem dari sudut pandang pengguna.

Aktor dalam *use case diagram* dapat berupa pengguna manusia, perangkat eksternal, maupun sistem lain yang berinteraksi dengan sistem yang sedang dikembangkan. Sementara itu, *use case* merepresentasikan fungsi-fungsi atau layanan yang diberikan sistem kepada aktor. Hubungan antara aktor dan *use case* digambarkan dengan garis asosiasi, yang menandakan adanya interaksi langsung. Berikut ini merupakan simbol *use case diagram*.

Tabel 2.2 Simbol dan Keterangan *Use Case Diagram*

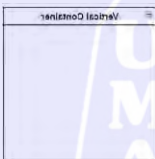
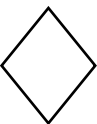
No	Nama	Gambar	Keterangan
1.	<i>Actor</i>		Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer.
2.	<i>Ekstends</i>		Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
3.	<i>Include</i>		Relasi <i>use case</i> tambahan sebuah <i>use case</i> yang ditambahkan melalui <i>use case</i> ini untuk menjalankan fungsinya untuk sebagai syarat dijalankan <i>use case</i> ini
4.	<i>Use Case</i>		Menjelaskan yang dilakukan aktor dari sistem untuk mencapai tujuan tertentu
5.	<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya.

6.	System		Menspesifikasikan data paket yang menampilkan system secara terbatas
----	--------	---	--

2.7.2 Activity Diagram

Activity Diagram adalah salah satu jenis diagram perilaku dalam UML yang digunakan untuk menggambarkan alur aktivitas atau proses bisnis dari suatu sistem secara terperinci (Sopriani & Purwanto, 2023). Diagram ini menekankan pada urutan aktivitas yang dilakukan oleh aktor maupun sistem, termasuk percabangan alur, kondisi keputusan, serta paralelisme dalam suatu proses (Rosa & Shalahuddin, 2021). *Activity diagram* terdiri dari beberapa elemen penting, antara lain:

Tabel 2.3 Simbol dan Keterangan *Activity Diagram*

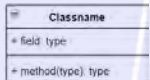
No	Nama	Simbol	Keterangan
1	Swimlane		mewakili pihak yang melakukan tindakan. Karena alur kerja biasanya memiliki agen yang berbeda yang melakukan langkah-langkah yang berbeda
2	InitialState		awal dari proses.
3	ActionState		melambangkan aktivitas alur kerja yang unik.
4	Transition		melambangkan urutan tindakan di antaranya.
5	FinalState		akhir proses.
6	Decision		titik pengambilan keputusan di mana alur kerja akan pergi ke salah satu arah.

2.7.3 Class Diagram

Diagram kelas menunjukkan hubungan antara kelas dalam model desain sistem dan penjelasan detail kelas masing-masing. Dia juga menunjukkan aturan dan

tanggung jawab entitas yang menentukan perilaku sistem (Saifulloh et al., 2021). Menurut (Sumirat et al., 2023), ada beberapa notasi dalam *Class Diagram*, yaitu sebagai berikut.

Tabel 2.4 Simbol dan Keterangan *Class Diagram*

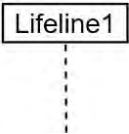
No	Nama	Simbol	Keterangan
1	Kelas		Kelas terdiri dari tiga komponen. Komponen-komponennya adalah sebagai berikut: • Komponen atas berisikan nama <i>class</i>. Setiap kelas memiliki nama yang unik; nama sederhana adalah sebutan lain untuk nama ini. • Komponen Tengah: komponen ini mengandung atribut kelas dan digunakan untuk menjelaskan kualitas kelas. Dengan memasukkan tipe nilai, atribut ini dapat dijelaskan lebih lanjut. • Komponen Bawah: Komponen ini mencakup operasi yang ditampilkan dalam bentuk daftar. Bagaimana suatu <i>class</i> berinteraksi dengan data dapat dijelaskan dengan operasi ini.
2	Asosiasi		Hubungan antara dua <i>class</i> statis dapat digambarkan sebagai asosiasi. Asosiasi biasanya menjelaskan <i>class</i> yang memiliki atribut tambahan seperti <i>class</i> lain.

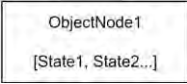
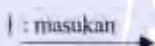
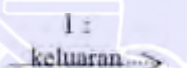
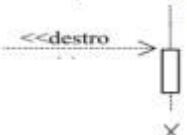
3	Agresi		Agregasi adalah hubungan antara dua <i>class</i> yang dapat berdiri sendiri, meskipun salah satunya merupakan bagian dari <i>class</i> lain.
4	Generalisasi		Kemampuan untuk memindahkan semua atribut dan metode dari <i>class</i> aslinya (<i>superclass</i>) ke <i>class</i> lain (<i>subclass</i>) dikenal sebagai generalisasi dalam diagram <i>class</i> .

2.7.4 Sequence Diagram

Sequence Diagram merupakan salah satu jenis diagram interaksi dalam UML yang digunakan untuk menggambarkan alur komunikasi antar objek dalam sistem berdasarkan urutan waktu. Diagram ini menekankan pada kronologi pesan yang dikirimkan dan diterima oleh objek, sehingga memberikan pemahaman yang lebih jelas mengenai proses bisnis atau skenario tertentu dalam aplikasi (Sopriani & Purwanto, 2023). Hal ini membantu dalam memastikan bahwa kebutuhan fungsional yang diharapkan oleh pengguna telah terakomodasi dengan baik dan meminimalisir terjadinya kesalahpahaman dalam implementasi sistem.

Tabel 2.5 Simbol dan Keterangan *Sequence Diagram*

No	Nama	Simbol	Keterangan
1	<i>Actor</i>		Aktor tidak selalu orang, meskipun representasi mereka adalah gambar seseorang; namun, kata "benda" biasanya digunakan untuk menggambarkan nama aktor. Selain itu, proses, orang, atau sistem yang berhubungan dengan sistem data yang dibuat berada di luar sistem informasi yang dibuat itu sendiri.
2	<i>Lifeline</i>		Ini menunjukkan siklus hidup suatu benda.

3	<i>Object</i>		menentukan item apa yang berinteraksi dengan pesan.
4	Waktu Aktif		Menegaskan bahwa benda itu aktif dan interaktif, dan bahwa segala sesuatu yang berkaitan dengan waktu aktif ini dibuat di dalamnya
5	Pesaan tipe create		Mendeklarasikan bahwa sebuah objek membuat objek lain, dan panah menunjuk ke objek yang dibuat. Arah panah menunjuk ke objek yang memiliki operasi atau metode karena itu memanggilnya. Karena itu, operasi atau metode yang dipanggil harus ada dalam diagram kelas sesuai dengan kelas objek yang berinteraksi.
6	Pesan tipe Send		merupakan item yang mengirimkan input, data, atau informasi ke item lain dan memiliki panah yang menunjuk ke item yang dikirim.
7	Pesan tipe return		Menunjukkan bahwa suatu objek kembali ke objek yang ditentukan setelah melakukan operasi atau metode tertentu, dan panah menunjuk ke objek yang menerima pengembalian.
8	Pesan tipe Destroy		Deklarasikan objek untuk mengakhiri hidup objek lain, dengan arah panah menunjuk ke objek yang diakhiri. Untuk hasil terbaik, jika ada pembuatan maka ada penghancuran.

BAB III

PEMBAHASAN

3.1 Bentuk Kegiatan

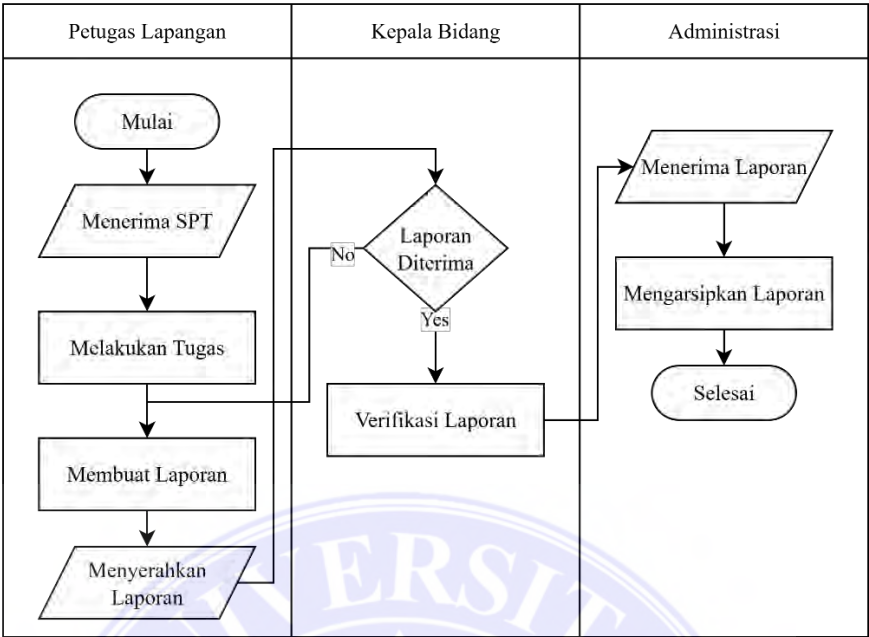
Kerja praktik yang dilaksanakan di Dinas Lingkungan Hidup bertujuan untuk mengembangkan sebuah sistem informasi pelaporan kegiatan petugas lapangan. Bentuk kegiatan utama yang dilakukan meliputi proses analisis kebutuhan melalui wawancara dan *individual search*, pengamatan terhadap sistem berjalan, Pembuatan sistem baru, serta pembuatan laporan.

Tabel 3.1 Jadwal Kegiatan

No	Kegiatan	Minggu Ke-1	Minggu Ke-2	Minggu Ke-3	Minggu Ke-4
1	Melakukan wawancara untuk sistem				
2	Melakukan pengamatan sistem yang berjalan dan Observasi				
3	Membuat sistem pelaporan baru				
4	Melakukan pembuatan laporan				

3.2 Analisis Sistem yang Berjalan

Sistem pelaporan kegiatan yang saat ini berjalan di Dinas Lingkungan Hidup masih bersifat konvensional dan belum sepenuhnya terintegrasi dengan teknologi informasi. Petugas lapangan umumnya melakukan pencatatan kegiatan secara manual, baik dalam bentuk catatan tertulis maupun dokumen sederhana yang kemudian diserahkan kepada kepala bidang untuk diverifikasi. Proses ini sering kali menimbulkan beberapa kendala, di antaranya keterlambatan dalam penyampaian laporan, kesulitan dalam mengarsipkan data, serta kurangnya transparansi dalam proses verifikasi.



Gambar 3.1 Diagram Analisis Sistem yang Sedang Berjalan

Selain itu, mekanisme komunikasi antara petugas lapangan, kepala bidang, dan administrasi dalam proses pelaporan kegiatan masih memiliki keterbatasan signifikan. Berdasarkan alur kerja yang ada, informasi mengenai kegiatan lapangan hanya disampaikan melalui laporan fisik yang dibuat dan diserahkan secara manual oleh petugas lapangan. Kepala bidang kemudian harus melakukan verifikasi secara manual sebelum meneruskan laporan tersebut ke administrasi. Proses ini menyebabkan keterlambatan dalam penerimaan dan pengarsipan laporan. Demikian juga, proses pengarsipan laporan di bagian administrasi juga masih bersifat manual, rentan terhadap kerusakan atau kehilangan dokumen fisik, serta kesulitan dalam pencarian dan analisis data jangka panjang. Kurangnya jejak digital dalam proses pelaporan juga menimbulkan kesulitan verifikasi keakuratan dan keaslian laporan secara tepat waktu. Oleh karena itu, diperlukan sistem informasi pelaporan berbasis web yang mampu mengintegrasikan pembuatan laporan digital dengan fitur GPS untuk memastikan lokasi dan waktu pelaksanaan kegiatan, serta verifikasi oleh kepala bidang secara cepat untuk memastikan validitas data.

3.3 Analisis Sistem yang Diusulkan

Sistem Informasi Pelaporan Kegiatan Petugas Lapangan berbasis web ini dirancang untuk mengatasi berbagai permasalahan pada sistem pelaporan manual yang selama ini digunakan di Dinas Lingkungan Hidup. Sistem ini memungkinkan petugas lapangan untuk membuat laporan digital secara langsung di lokasi kegiatan dengan dukungan fitur GPS, yang memastikan keakuratan data lokasi dan waktu pelaksanaan. Laporan yang telah dibuat kemudian dapat diverifikasi secara cepat oleh kepala bidang melalui platform yang sama, sehingga mengurangi keterlambatan dan meningkatkan transparansi dalam proses verifikasi. Seluruh data dan laporan tersimpan secara terpusat dalam basis data *PostgreSQL* melalui layanan Supabase, mempermudah pengarsipan, pencarian, serta analisis data dalam jangka panjang tanpa risiko kehilangan atau kerusakan dokumen. Dengan demikian, sistem ini tidak hanya mempercepat proses pelaporan dan verifikasi, tetapi juga meningkatkan akuntabilitas serta mendukung pengambilan keputusan berbasis data yang lebih akurat dan terkini untuk Dinas Lingkungan Hidup Kota Medan.

Melalui sistem ini, petugas lapangan dapat langsung memasukkan laporan kegiatan secara digital melalui antarmuka web. Laporan tersebut dapat mencakup deskripsi kegiatan, lokasi, waktu pelaksanaan, serta dokumentasi berupa foto sebagai bukti kegiatan. Data yang telah dimasukkan akan secara otomatis tersimpan dalam basis data sehingga memudahkan proses pencarian dan pengarsipan. Selain itu, petugas dapat mengisi kehadiran pada sistem, sehingga meningkatkan transparansi pelaksanaan pengawasan.

Kepala bidang memiliki akses untuk melakukan verifikasi secara langsung terhadap laporan yang masuk. Proses verifikasi dilakukan melalui akun yang terintegrasi, di mana kepala bidang dapat memberikan catatan, menyetujui, atau menolak laporan. Hal ini memberikan transparansi yang lebih tinggi serta mempercepat alur pengawasan.

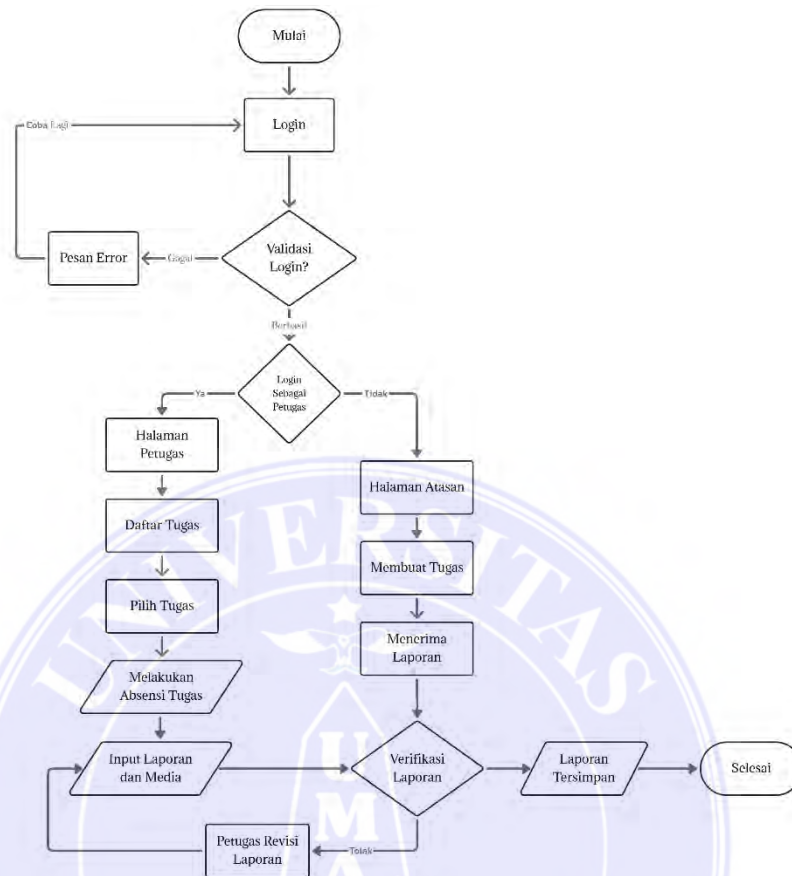
Admin, sebagai pihak pengelola sistem, memegang peran sentral dalam menjaga integritas dan operasional aplikasi. Tanggung jawab utamanya mencakup manajemen data pengguna, di mana Admin memiliki wewenang penuh untuk

menambah, mengubah, serta menetapkan peran (*role*) untuk setiap akun, baik itu Petugas Lapangan, Kepala bidang, maupun Admin lainnya. Selain itu, Admin bertugas melakukan pemeliharaan data sistem, seperti mengelola daftar jenis kegiatan dan memastikan konsistensi data di seluruh aplikasi. Admin juga berfungsi sebagai auditor utama melalui fitur log aktivitas yang mencatat semua perubahan penting di dalam sistem. Untuk kebutuhan analisis dan pelaporan yang lebih mendalam, Admin dapat mengakses dan mengekspor data rekapitulasi seluruh laporan ke dalam format CSV. Dengan demikian, pelaku dalam ruang lingkup Dinas Lingkungan Hidup Kota Medan yang berperan sebagai admin adalah ahli staf IT.

Selain menyederhanakan proses pelaporan, sistem ini juga memungkinkan pengolahan dan distribusi data laporan secara real-time kepada pihak yang berwenang. Setiap laporan yang dikirimkan oleh petugas lapangan akan langsung tersimpan dalam sistem dan dapat diakses oleh kepala bidang maupun admin sesuai hak aksesnya masing-masing. Kepala bidang dapat segera menindaklanjuti laporan tanpa menunggu dokumen fisik, sementara admin dapat memantau aktivitas sistem dan memastikan integritas data. Mekanisme ini secara langsung mempercepat proses monitoring terhadap kegiatan petugas di lapangan dan mengurangi potensi keterlambatan dalam penanganan temuan di lapangan. Dengan tersedianya data yang akurat dan dapat diakses kapan saja, proses pengawasan dan evaluasi menjadi lebih terstruktur dan responsif terhadap dinamika operasional di lapangan, sekaligus memperkuat transparansi dan akuntabilitas dalam pelaksanaan tugas di lingkungan Dinas Lingkungan Hidup Kota Medan.

3.3.1 *Flowchart* Pelaporan Petugas

Flowchart digunakan untuk menggambarkan alur kerja sistem pelaporan yang diusulkan, mulai dari proses login pengguna, input laporan kegiatan oleh petugas lapangan, verifikasi oleh kepala bidang, hingga pengelolaan data oleh admin.



Gambar 3.2 Flowchart Sistem Pelaporan

Flowchart yang disajikan menggambarkan alur kerja sistem pelaporan kegiatan petugas lapangan dan verifikasi laporan oleh kepala bidang dalam Dinas Lingkungan Hidup. Proses dimulai dari tahap autentikasi pengguna melalui fitur login pada sistem. Pengguna memasukkan kredensial yang kemudian dilakukan validasi oleh sistem. Jika validasi *login* gagal, sistem menampilkan pesan *error* dan memberikan opsi kepada pengguna untuk mencoba kembali proses *login*. Sebaliknya, apabila validasi berhasil, sistem akan menentukan peran pengguna berdasarkan kredensial yang dimasukkan, apakah sebagai petugas lapangan atau kepala bidang.

Apabila pengguna berhasil *login* sebagai petugas lapangan, sistem mengarahkan ke halaman utama petugas yang menampilkan daftar tugas yang harus diselesaikan. Petugas kemudian memilih salah satu tugas dari daftar tersebut dan melakukan absensi sebagai tanda kehadiran pada tugas yang dilakukan. Selanjutnya, petugas melanjutkan proses dengan menginput laporan kegiatan beserta media

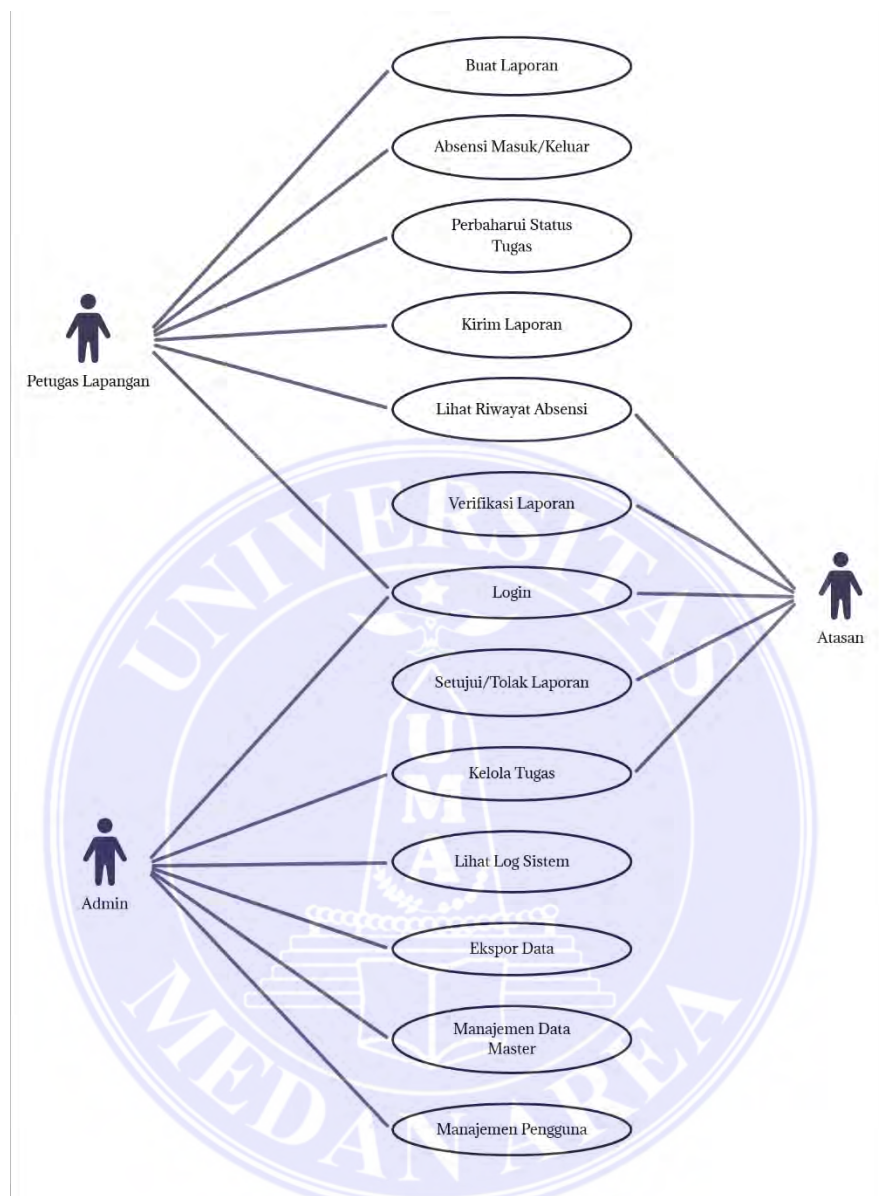
pendukungnya, seperti foto atau dokumen digital terkait aktivitas di lapangan.

Setelah laporan selesai di-*input*, laporan tersebut dikirimkan ke kepala bidang untuk proses verifikasi. Jika laporan diterima dan valid, maka laporan tersebut akan disimpan dalam sistem secara permanen, menandai akhir dari proses pelaporan untuk tugas tersebut. Namun, jika kepala bidang menolak laporan berdasarkan hasil verifikasi, laporan akan dikembalikan kepada petugas untuk dilakukan revisi. Petugas kemudian harus memperbaiki dan melengkapi laporan sesuai dengan catatan atau koreksi yang diberikan sebelum mengajukan ulang laporan tersebut untuk diverifikasi kembali oleh kepala bidang.

Di sisi lain, apabila pengguna *login* sebagai atasan/kepala bidang, sistem mengarahkan pengguna ke halaman atasan yang berfungsi untuk membuat dan mengelola tugas bagi petugas lapangan. Setelah membuat tugas, atasan juga memiliki akses untuk menerima laporan yang telah disampaikan oleh petugas, kemudian melakukan proses verifikasi terhadap laporan tersebut. Verifikasi ini merupakan langkah krusial untuk memastikan validitas dan kelengkapan laporan yang diterima.

3.3.2 Use Case Diagram Pelaporan Petugas

Use case diagram menggambarkan interaksi antara aktor (petugas lapangan, atasan, dan admin) dengan sistem. Diagram ini menunjukkan fitur utama yang dapat diakses masing-masing pengguna sesuai dengan hak aksesnya, sehingga peran tiap aktor dalam sistem menjadi jelas. Petugas lapangan bertugas membuat dan mengirim laporan kegiatan digital, melakukan absensi masuk dan keluar, serta memperbarui status tugas dan melihat riwayat absensi. Sedangkan atasan bertanggung jawab untuk memverifikasi laporan, memberikan persetujuan atau penolakan, serta memantau riwayat absensi petugas guna memastikan akurasi dan kelengkapan data yang digunakan untuk evaluasi dan pengambilan keputusan. Admin memiliki peran sentral dalam pengelolaan sistem dengan tanggung jawab melakukan manajemen tugas, pengguna, dan data master, serta memantau aktivitas sistem melalui log. Admin juga dapat mengeksport data untuk keperluan analisis lebih lanjut dan pelaporan resmi.

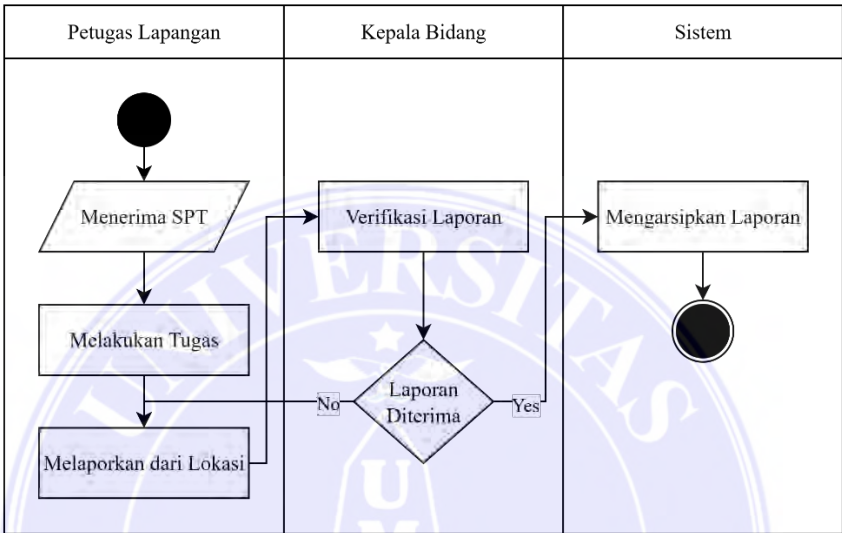


Gambar 3.3 Use Case Diagram Sistem Pelaporan

3.3.3 Activity Diagram Pelaporan Petugas

Activity diagram pada sistem pelaporan petugas lapangan menggambarkan alur utama yang melibatkan tiga aktor, yaitu petugas lapangan, kepala bidang, dan sistem. Proses dimulai ketika petugas menerima Surat Perintah Tugas (SPT), kemudian melaksanakan tugas yang diberikan serta menyusun laporan dari lokasi. Laporan tersebut selanjutnya diverifikasi oleh kepala bidang melalui proses pengawasan. Apabila laporan ditolak, maka dikembalikan kepada petugas untuk

diperbaiki, sedangkan apabila laporan diterima, maka sistem secara otomatis mengarsipkannya. Dengan demikian, diagram ini memperlihatkan hubungan yang terstruktur antara pelaksanaan tugas di lapangan, proses pengawasan oleh kepala bidang, serta pengelolaan arsip laporan oleh sistem sehingga alur pelaporan berjalan lebih tertib dan efisien.

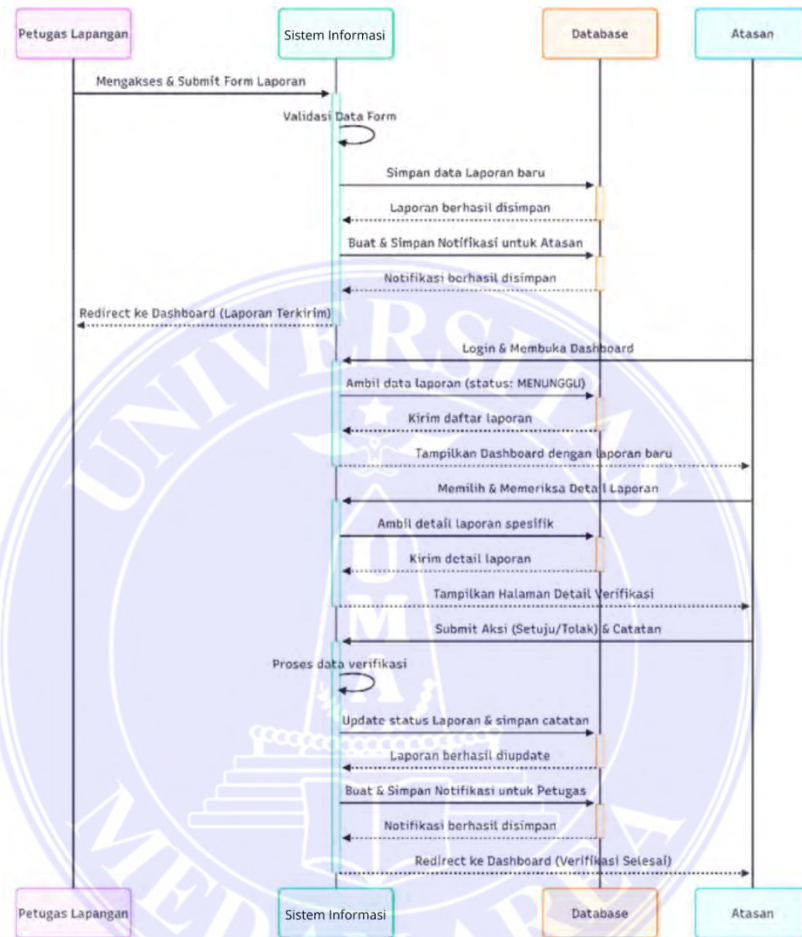


Gambar 3.4 Activity Diagram Sistem Pelaporan

3.3.4 Sequence Diagram Pelaporan Petugas

Sequence diagram merupakan representasi visual yang digunakan untuk menggambarkan urutan interaksi antar objek dalam suatu sistem berdasarkan alur waktu terjadinya proses. Diagram ini berfungsi untuk menjelaskan mekanisme komunikasi dan pertukaran data antara aktor serta komponen sistem dalam rangka mencapai suatu tujuan operasional. Berdasarkan gambar yang ditampilkan, sequence diagram ini menunjukkan proses pelaporan dan verifikasi data lapangan yang melibatkan tiga komponen utama, yaitu petugas lapangan, sistem informasi, dan kepala bidang, dengan dukungan basis data sebagai media penyimpanan. Proses yang digambarkan mencakup tahapan mulai dari pengiriman laporan oleh petugas, penyimpanan data dan pembuatan notifikasi oleh sistem, hingga proses verifikasi oleh kepala bidang serta pemberitahuan hasil verifikasi kepada petugas lapangan. Diagram ini memberikan gambaran menyeluruh mengenai aliran proses dan interaksi yang terjadi dalam sistem pelaporan berbasis digital secara

terstruktur dan kronologis.menjelaskan urutan interaksi antar objek dalam sistem, misalnya interaksi antara petugas lapangan dengan sistem saat mengirim laporan, atau interaksi antara atasan dengan sistem saat melakukan verifikasi. Diagram ini menekankan pada urutan waktu terjadinya proses.

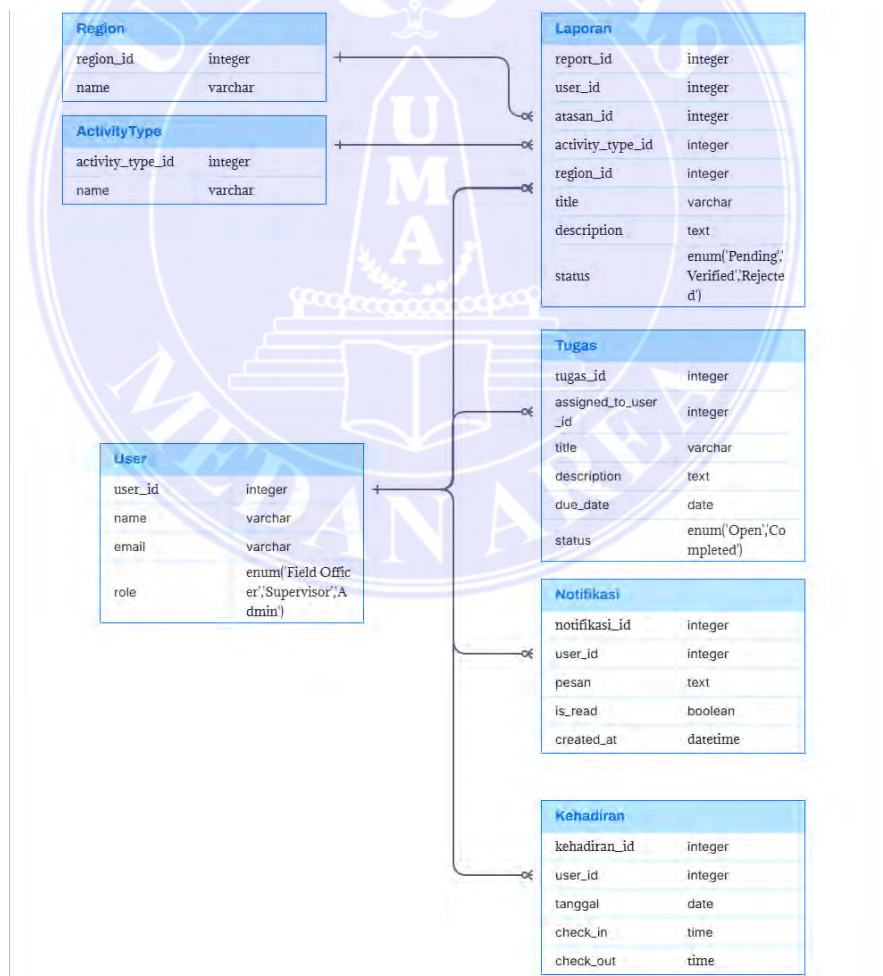


Gambar 3.5 Sequence Diagram Sistem Pelaporan

3.3.5 Entity Relationship Diagram (ERD) Pelaporan Petugas

Entity Relationship Diagram (ERD) untuk sistem informasi pelaporan yang melibatkan pengguna, kepala bidang, dan admin. ERD ini menjelaskan struktur basis data beserta hubungan antar entitas. Terdapat entitas *User* yang menyimpan informasi pengguna dengan atribut *role* yang membedakan peran sebagai *Field Officer* (petugas lapangan), *Supervisor* (Kepala Bidang Penataan), atau *Admin*. Entitas Laporan berelasi dengan *User*, *ActivityType*, dan *Region* untuk mencatat laporan yang dibuat petugas berdasarkan jenis kegiatan dan wilayah kerja, dengan

status laporan dapat berupa *Pending*, *Verified*, atau *Rejected*. Entitas Tugas digunakan untuk mengelola penugasan dari kepala bidang kepada petugas dengan atribut seperti *due_date* dan status (*Open* atau *Completed*). Entitas Notifikasi berfungsi mengirimkan pesan otomatis ke pengguna terkait status laporan atau tugas, dengan atribut penanda apakah notifikasi sudah dibaca. Selain itu, entitas Kehadiran mencatat aktivitas presensi pengguna berupa *check_in* dan *check_out* setiap tanggal tertentu. Sementara itu, entitas Region dan *ActivityType* berfungsi sebagai data referensi untuk mendefinisikan wilayah kerja serta jenis kegiatan. Dengan struktur ini, sistem mampu mendukung alur utama yang telah dijelaskan sebelumnya, yaitu pelaporan oleh petugas, verifikasi oleh kepala bidang, hingga pengarsipan laporan oleh sistem, serta dilengkapi fitur pendukung seperti penugasan, notifikasi, dan presensi pengguna.



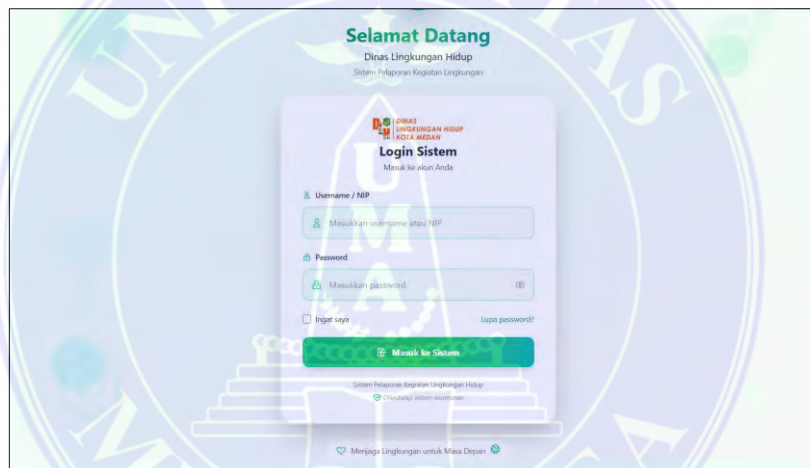
Gambar 3.6 ERD Sistem Pelaporan

3.4 Hasil Pengembangan Sistem

3.4.1 Fitur Umum Sistem Pelaporan Petugas

1. Antarmuka Login

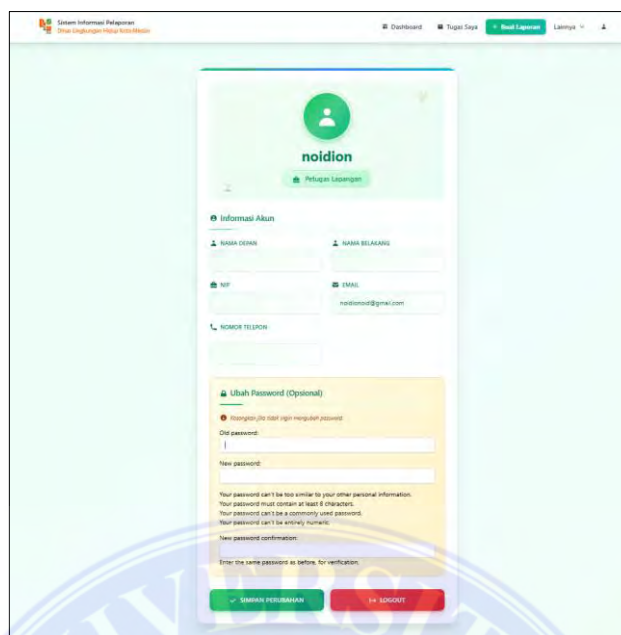
Fitur login merupakan gerbang akses utama bagi semua pengguna untuk masuk ke dalam sistem. Pengguna diwajibkan memasukkan *username* dan *password* mereka yang telah terdaftar. Sistem kemudian akan memvalidasi kredensial tersebut untuk memastikan keabsahan akun. Jika berhasil, pengguna akan langsung diarahkan ke halaman *dashboard* yang telah disesuaikan dengan perannya, baik sebagai Petugas, Kepala bidang, maupun Admin. Hal ini memastikan bahwa setiap pengguna hanya dapat mengakses fitur dan data yang relevan dengan tugasnya. Apabila proses login gagal, pengguna akan tetap berada di halaman login untuk mencoba kembali.



Gambar 3.7 Halaman Login

2. Antarmuka Profil

Halaman profil memungkinkan setiap pengguna untuk mengelola informasi akun pribadinya. Pengguna dapat melihat data diri mereka serta mengubah informasi yang diizinkan seperti email, nomor telepon, dan *password*. Fitur ini bertujuan untuk menjaga akurasi data pengguna dan keamanan akses akun.

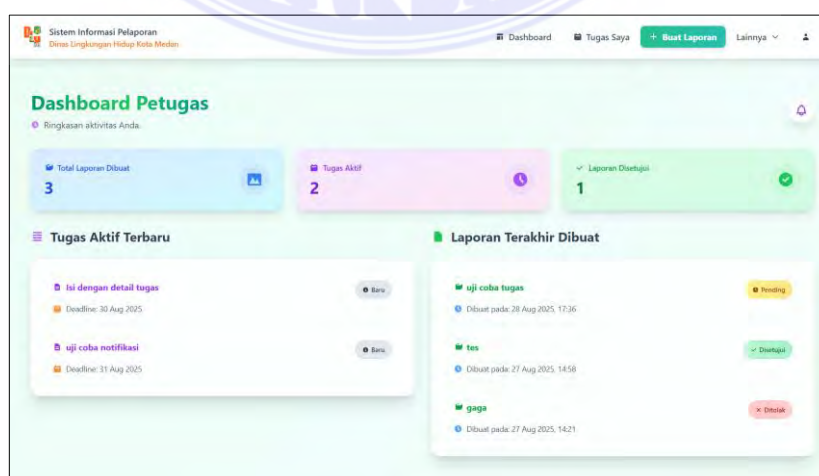


Gambar 3.8 Halaman Profil

3.4.2 Hak Akses Petugas Lapangan

1. Dashboard Petugas Lapangan

Dashboard Petugas Lapangan adalah halaman utama yang menyambut pengguna setelah berhasil login, berfungsi sebagai pusat informasi personal. Halaman ini dirancang untuk menyajikan ringkasan visual dari semua aktivitas yang relevan dengan pekerjaan petugas, sehingga memungkinkan pengguna memperoleh gambaran cepat mengenai beban kerja dan status terkini. Secara terintegrasi, *dashboard* menyajikan beberapa komponen informasi kunci, antara lain kartu statistik, daftar tugas aktif, dan daftar laporan terbaru.

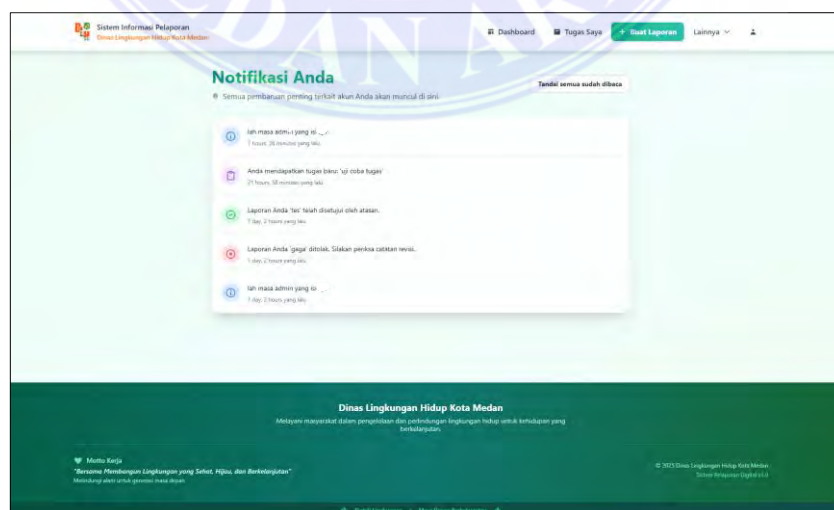


Gambar 3.9 Halaman Dashboard Petugas Lapangan

Komponen pertama adalah tiga kartu statistik yang menampilkan data agregat, yaitu jumlah total laporan yang pernah dibuat, jumlah tugas yang masih aktif, dan total laporan yang telah disetujui oleh atasan. Di bawah kartu statistik, terdapat dua daftar dinamis. Bagian pertama adalah “Tugas Aktif Terbaru” yang menampilkan tugas-tugas yang belum selesai dan diurutkan berdasarkan urgensi batas waktu. Bagian kedua adalah "Laporan Terakhir Dibuat" yang menampilkan riwayat laporan terkini beserta status verifikasi, yang ditandai dengan label berwarna untuk memudahkan identifikasi seperti menunggu verifikasi, disetujui, atau ditolak. Terakhir, di sudut kanan atas halaman terdapat ikon notifikasi yang akan menampilkan indikator visual jika terdapat pemberitahuan baru yang belum dibaca.

2. Notifikasi

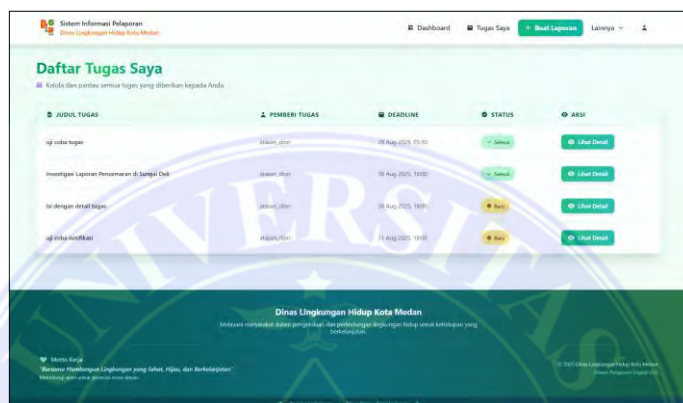
Fitur notifikasi berfungsi sebagai sistem pemberitahuan otomatis untuk menginformasikan pengguna tentang aktivitas penting yang relevan. Sistem akan secara otomatis mengirimkan pemberitahuan kepada Atasan setiap kali ada laporan baru yang perlu diverifikasi. Sebaliknya, petugas akan menerima notifikasi ketika laporannya telah disetujui atau ditolak oleh Atasan. Pengguna dapat mengakses daftar semua notifikasi melalui ikon lonceng yang terletak di sudut kanan atas halaman *dashboard* mereka. Fitur ini bertujuan untuk memastikan alur komunikasi berjalan efisien dan setiap pihak mendapatkan informasi status pekerjaan secara *real-time*.



Gambar 3.10 Halaman Dashboard Petugas Lapangan

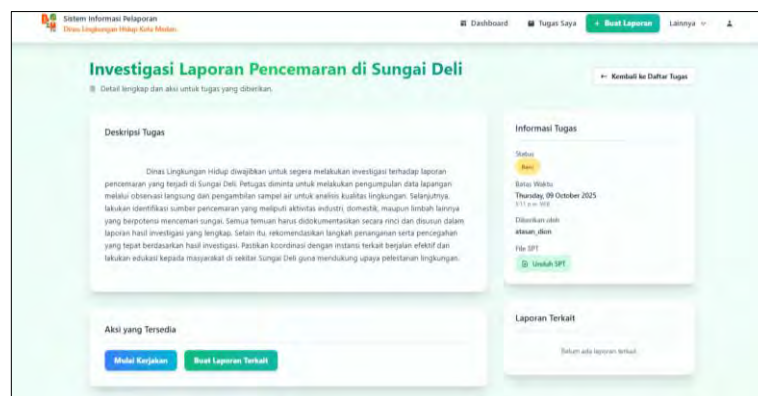
3. Halaman Tugas Petugas Lapangan

Fitur ini berfungsi sebagai agenda kerja digital bagi petugas. Halaman ini menampilkan daftar semua tugas yang diberikan oleh atasan, lengkap dengan judul, batas waktu, dan status pengerjaan. Petugas dapat melihat detail setiap tugas dan mengubah statusnya, memungkinkan pengelolaan beban kerja yang lebih terstruktur dan efisien.



Gambar 3.11 Halaman Tugas Petugas Lapangan

Antarmuka fitur ini menyajikan data dalam format tabel yang informatif, di mana setiap baris merepresentasikan satu tugas. Kolom-kolom pada tabel ini mencakup informasi esensial seperti judul tugas, pemberi tugas, batas waktu (*Deadline*), serta status pengerjaan tugas saat ini. Status setiap tugas divisualisasikan menggunakan label berwarna untuk kemudahan identifikasi, seperti baru, dikerjakan, atau selesai, yang memberikan gambaran cepat mengenai progres kerja. Selain itu, pada kolom aksi, terdapat tombol interaktif lihat detail untuk setiap tugas. Ketika tombol ini ditekan, sistem akan mengarahkan pengguna ke halaman Detail Tugas yang lebih komprehensif.



Gambar 3.12 Halaman Detail Tugas Petugas Lapangan

Di halaman tersebut, petugas dapat membaca deskripsi lengkap, mengunduh Surat Perintah Tugas (SPT) jika ada, mengubah status pengerjaan tugas, dan membuat laporan yang secara otomatis terhubung dengan tugas tersebut. Dengan demikian, fitur ini tidak hanya berfungsi sebagai pengingat, tetapi juga sebagai titik awal dari alur kerja pelaporan berbasis tugas yang akuntabel dan efisien.

4. Halaman Buat Laporan Petugas Lapangan

Fitur "Buat Laporan" adalah fungsi inti untuk input data oleh petugas. Melalui formulir digital ini, petugas dapat melaporkan hasil kegiatan lapangan dengan mengisi detail seperti jenis kegiatan, lokasi, temuan, serta mengunggah bukti foto dan dokumen. Fitur ini menggantikan proses manual, sehingga pelaporan menjadi lebih cepat, terstandar, dan terdokumentasi dengan baik.

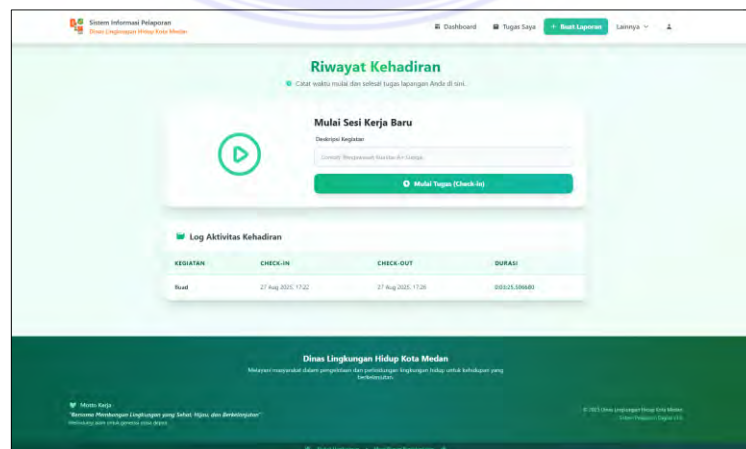
Gambar 3.13 Halaman Buat Laporan Petugas Lapangan

Halaman ini dapat diakses melalui tombol utama di menu navigasi dan menyajikan sebuah formulir komprehensif yang dirancang untuk menggantikan pencatatan manual. Formulir ini terbagi menjadi beberapa input field terstruktur yang mencakup semua informasi esensial, antara lain judul kegiatan, jenis kegiatan (dalam bentuk pilihan dropdown), waktu kegiatan, dan lokasi sesuai pengamatan petugas.

Bagian paling krusial dari formulir ini adalah kolom-kolom untuk dokumentasi dan validasi. Petugas diwajibkan untuk mengisi hasil temuan dalam format teks dan dapat melampirkan bukti pendukung melalui *field* dokumentasi foto dan dokumen pendukung. Sebagai inovasi utama, sistem secara otomatis mengaktifkan fitur geolokasi untuk menangkap koordinat GPS dari lokasi pelaporan. Informasi lokasi otomatis ini akan ditampilkan kepada petugas untuk konfirmasi dan disimpan bersama laporan sebagai data pembanding, yang secara signifikan meningkatkan akurasi dan akuntabilitas data. Setelah semua field terisi, petugas menekan tombol kirim laporan untuk menyimpan data ke dalam sistem dan secara otomatis mengirimkan notifikasi kepada Atasan.

5. Halaman Riwayat Kehadiran Petugas Lapangan

Fitur Riwayat Kehadiran berfungsi sebagai sistem presensi digital untuk mencatat jam kerja petugas di lapangan. Sebelum memulai kegiatan, petugas dapat melakukan *Check-in* dengan memasukkan deskripsi singkat tugasnya, dan sistem akan secara otomatis mencatat waktu mulainya. Setelah kegiatan selesai, petugas melakukan *Check-out* untuk mencatat waktu berakhirnya.

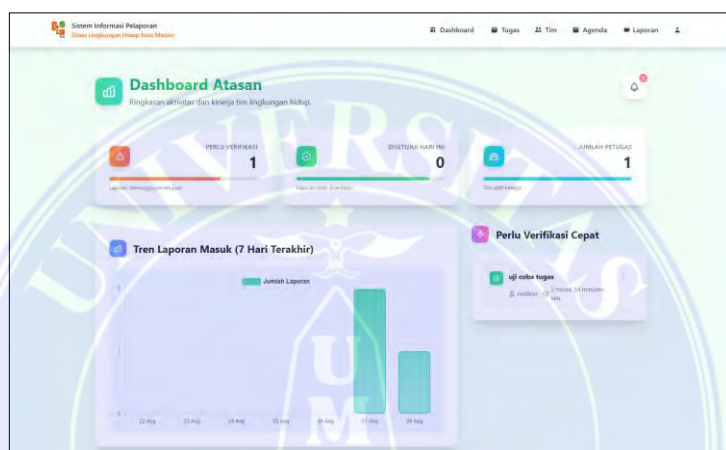


Gambar 3.14 Halaman Riwayat Kehadiran Petugas Lapangan

3.4.3 Hak Akses Atasan

1. Dashboard Atasan

Dashboard Atasan berfungsi sebagai pusat kendali utama yang menyajikan ringkasan visual dari seluruh aktivitas tim. Halaman ini menampilkan statistik kunci, seperti jumlah laporan yang perlu diverifikasi dan laporan yang disetujui hari ini. Selain itu, *dashboard* juga dilengkapi grafik tren laporan masuk dan daftar laporan terbaru yang membutuhkan verifikasi, memungkinkan Atasan untuk memantau produktivitas tim secara cepat dan efisien.



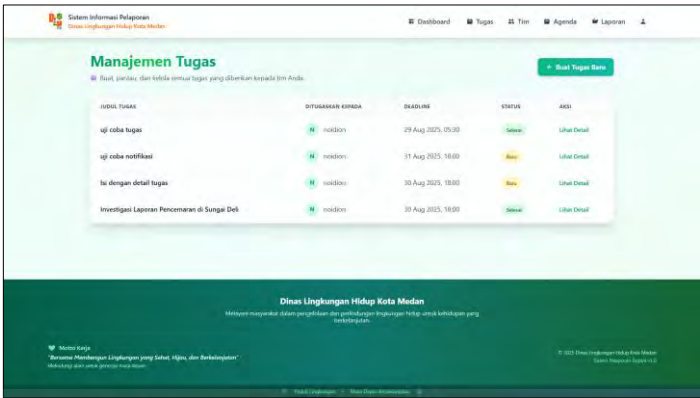
Gambar 3.15 Halaman Dashboard Atasan

Komponen pertama adalah tiga kartu statistik yang memberikan gambaran umum, yaitu jumlah laporan yang sedang menunggu persetujuan, total laporan yang telah disetujui pada hari ini, dan jumlah total petugas yang aktif bekerja. Di bawahnya, terdapat komponen tren laporan masuk yang memvisualisasikan jumlah laporan yang masuk setiap harinya selama tujuh hari terakhir dalam bentuk grafik batang, sehingga memudahkan Atasan untuk mengidentifikasi pola dan tingkat produktivitas tim. Di sisi kanan, terdapat daftar "Perlu Verifikasi Cepat" yang menampilkan laporan-laporan terbaru yang membutuhkan tindakan segera, diurutkan berdasarkan waktu pengiriman.

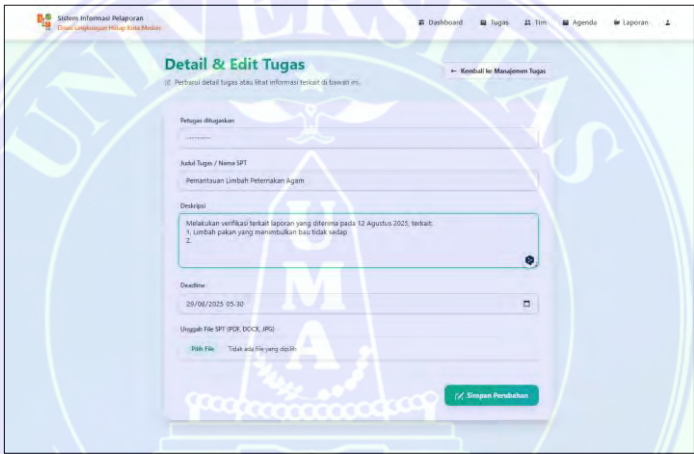
2. Halaman Manajemen Tugas oleh Atasan

Fitur ini memberikan Atasan wewenang untuk membuat dan mendelegasikan tugas kepada Petugas Lapangan. Melalui formulir khusus, Atasan dapat menetapkan judul, deskripsi, batas waktu, dan menugaskannya kepada petugas

yang relevan. Halaman ini juga menampilkan daftar semua tugas yang telah dibuat, berfungsi sebagai arsip dan alat pemantauan penugasan.



Gambar 3.16 Halaman Manajemen Tugas



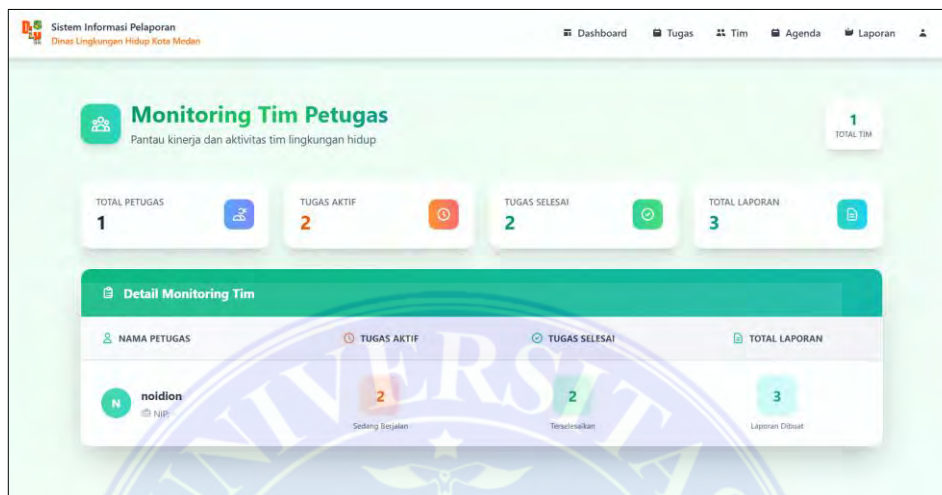
Gambar 3.17 Halaman Detail dan Edit Tugas

Selain fungsi pembuatan tugas, halaman ini juga menyajikan daftar semua tugas yang telah dibuat dalam format tabel yang terstruktur. Setiap baris pada tabel menampilkan informasi ringkas seperti judul tugas, nama petugas yang ditugaskan, batas waktu, dan status pengerjaan terkini. Pada kolom aksi, terdapat tautan lihat detail yang bersifat interaktif. Ketika tautan ini diklik, Atasan akan diarahkan ke halaman detail & edit tugas, di mana mereka dapat meninjau informasi lengkap, mengunduh SPT yang terlampir, dan melakukan perubahan atau pembaruan pada detail tugas jika diperlukan. Dengan demikian, fitur ini berfungsi sebagai arsip digital penugasan sekaligus alat pemantauan yang aktif.

3. Halaman Monitoring oleh Atasan

Halaman Monitoring Tim menyediakan pandangan menyeluruh terhadap kinerja

setiap Petugas Lapangan. Atasan dapat melihat daftar semua petugas beserta rekapitulasi jumlah tugas yang sedang aktif, tugas yang telah selesai, dan total laporan yang telah dibuat oleh masing-masing individu. Fitur ini bertujuan untuk mempermudah evaluasi kinerja dan pemantauan beban kerja tim.



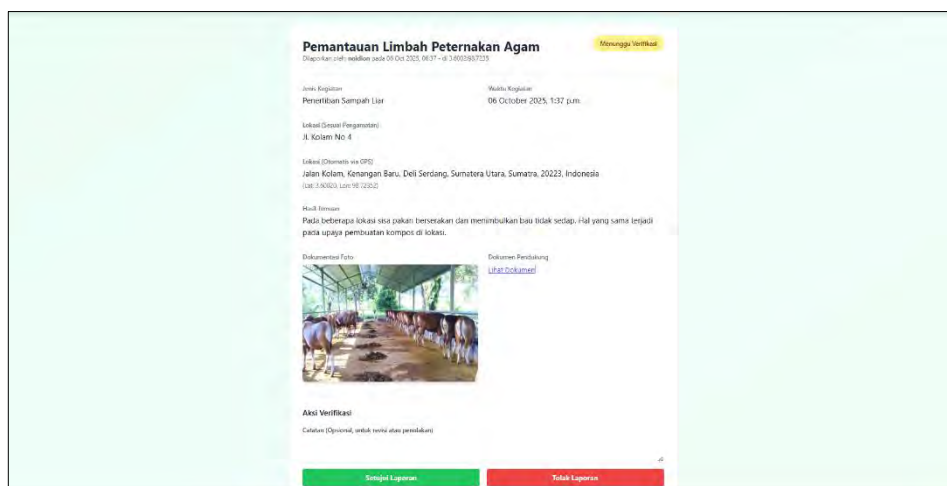
Gambar 3.18 Halaman Monitoring Tim

4. Halaman Verifikasi Laporan oleh Atasan

Fitur ini adalah fungsi inti bagi Atasan untuk melakukan kontrol kualitas terhadap laporan yang masuk. Atasan dapat melihat detail lengkap laporan yang dikirim oleh petugas, termasuk bukti foto dan dokumen. Berdasarkan pemeriksaan tersebut, Atasan memiliki wewenang untuk menyetujui atau menolak laporan, serta memberikan catatan revisi jika diperlukan.

Semua Data Laporan					
Sistem Pelaporan Dinas Lingkungan Hidup					
Filter Pencarian Petugas: Semua Petugas Status Verifikasi: Semua Status Dari Tanggal: hh/bb/tttt Sampai Tanggal: hh/bb/tttt Filter Reset					
ID	PETUGAS	WAKTU KEGIATAN	STATUS LAPORAN	SPK TERKAIT	Aksi
Pemantauan Pemukiman sapi II Masjid Kel Sari rejo Kac Medan Polonia (30 April 2025) (nv)	noidon	31 Oct 2025, 12:56	Sedang Dilaksanakan	-	Menunggu Verifikasi Detail
Pemantauan Pabrik Roti Ngetop Bakery an. Jonan di Titi Kuring (23 April 2025) (nv)	noidon	08 Oct 2025, 12:56	Sedang Dilaksanakan	-	Menunggu Verifikasi Detail
Verifikasi Pembakaran sampah yang berlokasi di Jalan Desal Kompo awaserta Kelurahan Togal 2 Mandala II (nv)	noidon	06 Oct 2025, 12:54	Sedang Dilaksanakan	-	Menunggu Verifikasi Detail
Pemantauan Limbah Pemukiman Agam	noidon	06 Oct 2025, 12:48	Sedang Dilaksanakan	-	Menunggu Verifikasi Detail

Gambar 3.19 Halaman Laporan yang Dikirim Petugas



Gambar 3.20 Halaman Detail Laporan

Antarmuka utamanya menyajikan detail lengkap dari satu laporan spesifik, mencakup semua informasi yang diinput oleh petugas seperti jenis kegiatan, waktu, lokasi pengamatan, hasil temuan, serta bukti pendukung berupa foto dan dokumen. Sebagai elemen validasi tambahan yang krusial, halaman ini juga menampilkan perbandingan antara lokasi yang diinput manual oleh petugas dengan lokasi yang terekam secara otomatis melalui GPS, lengkap dengan alamat teks hasil reverse geocoding.

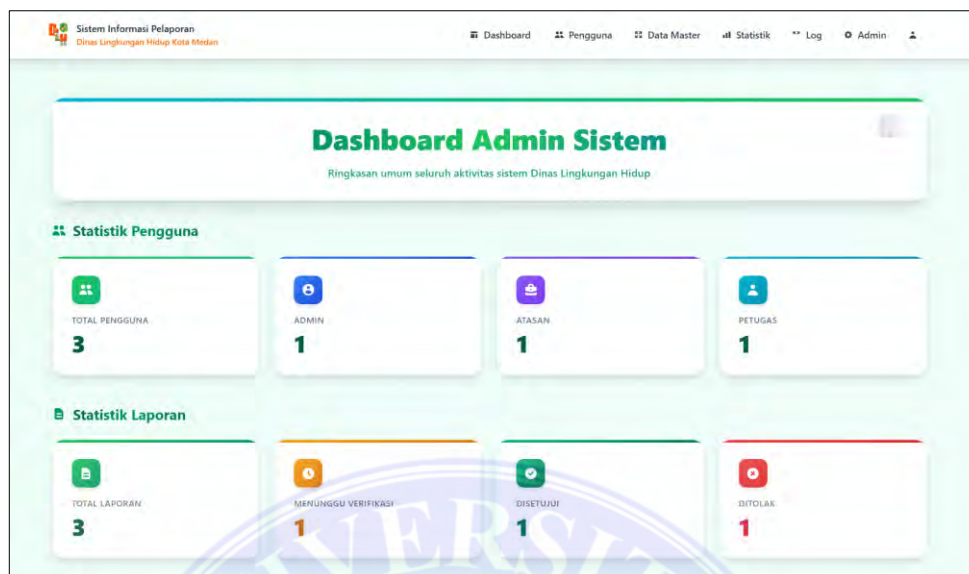
Di bagian bawah halaman, terdapat panel aksi verifikasi yang dirancang khusus untuk Atasan. Panel ini menyediakan kolom input untuk memberikan catatan yang bersifat opsional, yang sangat berguna untuk memberikan arahan revisi jika laporan ditolak. Terdapat dua tombol aksi utama, yakni setuju laporan dan tolak laporan. Ketika salah satu tombol ditekan, sistem akan memperbarui status laporan di database, menyimpan catatan yang diberikan, dan secara otomatis mengirimkan notifikasi kepada petugas yang bersangkutan mengenai hasil verifikasi tersebut.

3.4.4 Hak Akses Admin

1. Dashboard Admin

Dashboard Admin berfungsi sebagai pusat pemantauan sistem secara keseluruhan. Halaman ini menyajikan ringkasan statistik vital, seperti total pengguna berdasarkan peran, jumlah laporan berdasarkan status, serta daftar aktivitas terbaru. Tujuannya adalah untuk memberikan Admin gambaran umum yang cepat dan komprehensif mengenai kondisi dan aktivitas terkini di dalam

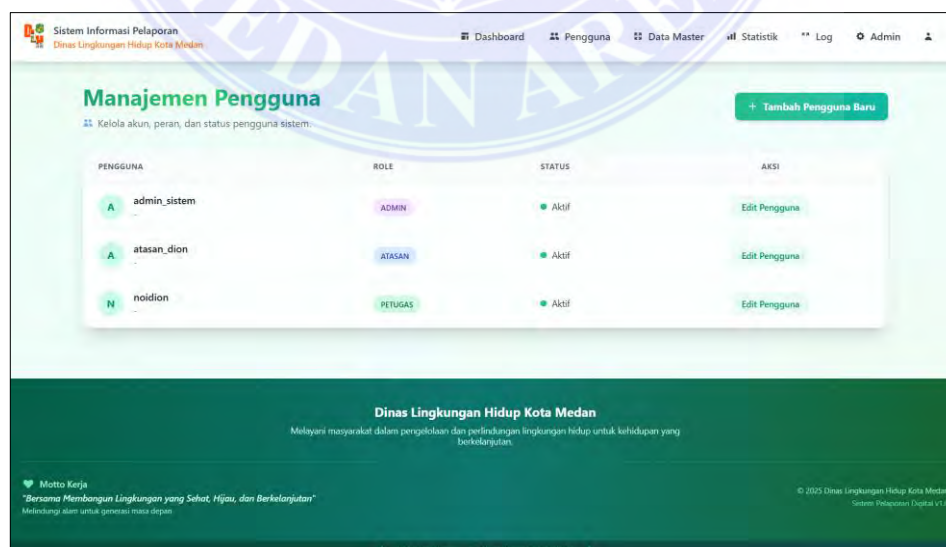
sistem.



Gambar 3.21 Halaman Dashboard Admin

2. Halaman Manajemen Pengguna

Fitur ini merupakan pusat kendali untuk semua akun pengguna di dalam sistem. Admin dapat melihat daftar seluruh pengguna, menambahkan akun baru untuk petugas atau atasan, serta mengubah informasi dan peran pengguna yang sudah ada. Fitur ini memastikan bahwa pengelolaan hak akses dan data pengguna dapat dilakukan secara terpusat dan efisien.

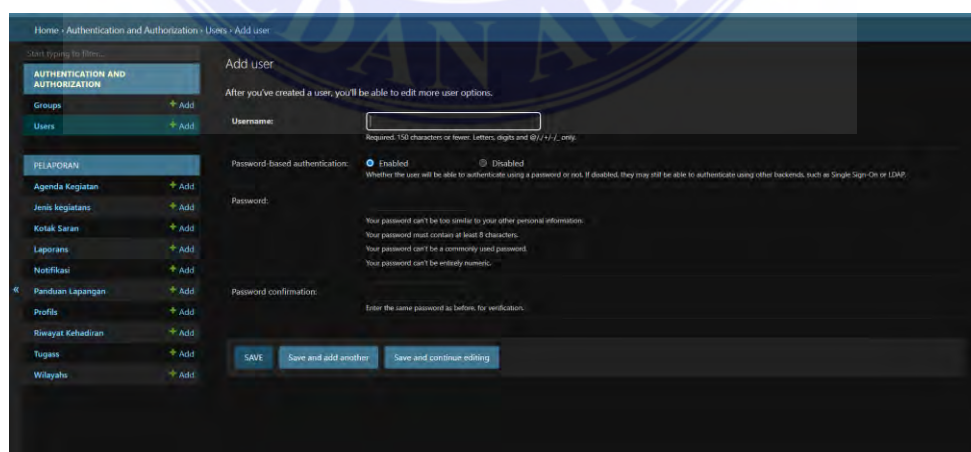


Gambar 3.22 Halaman Manajemen Pengguna

Tampilan utama fitur ini menyajikan daftar semua pengguna dalam format tabel yang informatif, menampilkan detail krusial seperti pengguna (*username*), *role* (peran), status (aktif/tidak aktif), dan kolom aksi untuk interaksi lebih lanjut. Halaman ini memiliki dua tombol interaktif utama. Tombol pertama, tambah pengguna baru, terletak di bagian kanan atas. Ketika tombol ini ditekan, Admin akan diarahkan ke antarmuka khusus di dalam Panel Admin Django untuk membuat akun pengguna baru. Di sana, Admin dapat mengisi semua informasi yang diperlukan, seperti *username*, *password*, serta menetapkan izin dan status kepegawaian. Tombol kedua, edit pengguna, tersedia di setiap baris pengguna pada kolom aksi. Menekan tombol ini akan membawa Admin ke halaman edit pengguna yang bersangkutan di dalam Panel Admin Django. Pada halaman ini, Admin memiliki wewenang untuk memperbarui informasi personal pengguna, mengubah *password*, dan yang terpenting, menyesuaikan peran (*role*) dan hak akses mereka sesuai dengan kebutuhan operasional.

3. Halaman Panel Admin

Panel Admin Django adalah antarmuka *backend* bawaan yang kuat untuk pengelolaan sistem tingkat lanjut. Melalui panel ini, Admin memiliki akses penuh untuk mengelola setiap aspek data secara mendetail, termasuk pengaturan izin yang kompleks dan konfigurasi teknis lainnya. Halaman ini berfungsi sebagai ruang mesin sistem untuk tugas-tugas administratif yang mendalam.

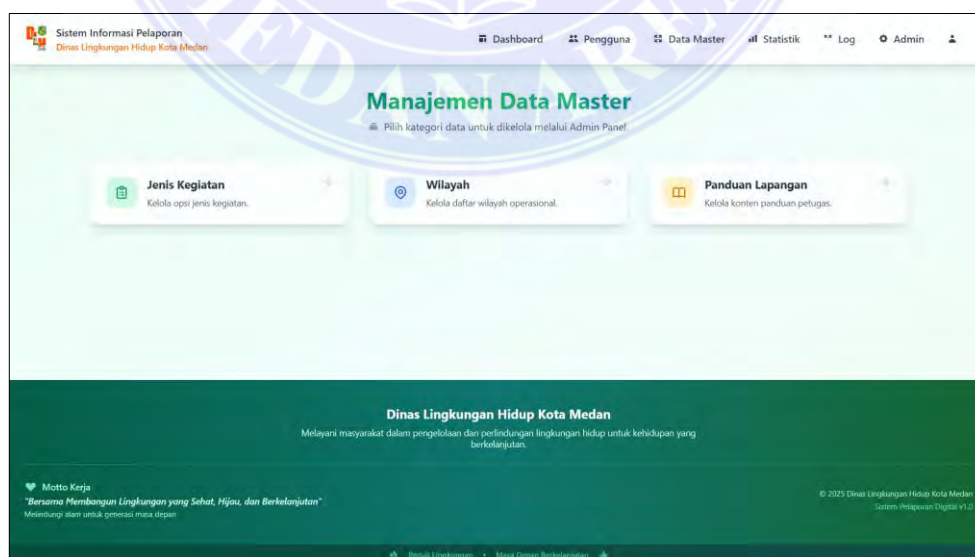


Gambar 3.23 Halaman Panel Admin

Akses ke panel ini dibatasi hanya untuk pengguna dengan status superuser atau staf, memastikan bahwa hanya administrator yang berwenang yang dapat melakukan perubahan kritis pada sistem. Seperti yang ditunjukkan pada Gambar 3.23, antarmuka Panel Admin menyediakan daftar lengkap dari semua model data yang terdaftar dalam sistem, seperti *users*, *profils*, *laporans*, dan *tugas*. Melalui halaman ini, Admin memiliki kapabilitas penuh untuk melakukan operasi *Create*, *Read*, *Update*, *Delete* (CRUD) pada setiap entri data di dalam database. Fungsionalitas ini sangat krusial untuk tugas-tugas seperti memperbaiki data yang salah input, mengelola data referensi secara massal, atau melihat riwayat perubahan pada objek tertentu. Dalam arsitektur sistem yang dikembangkan, halaman-halaman seperti manajemen pengguna dan manajemen data master secara sengaja dihubungkan ke Panel Admin ini untuk memanfaatkan fungsionalitasnya yang sudah teruji, aman, dan lengkap, sehingga mempercepat pengembangan tanpa mengorbankan keamanan atau fungsionalitas inti.

4. Halaman Manajemen Data

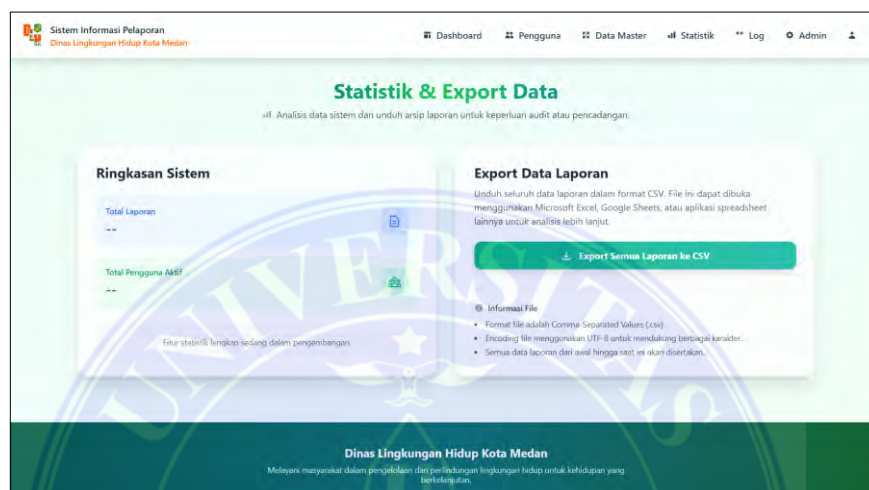
Fitur ini memungkinkan Admin untuk mengelola data referensi yang menjadi dasar pilihan di seluruh sistem. Admin dapat menambah atau mengubah opsi-opsi penting seperti jenis kegiatan dan wilayah. Tujuannya adalah untuk menjaga konsistensi dan relevansi data yang digunakan dalam operasional sehari-hari.



Gambar 3.24 Halaman Manajemen Data

5. Halaman Statistik dan Ekspor Data

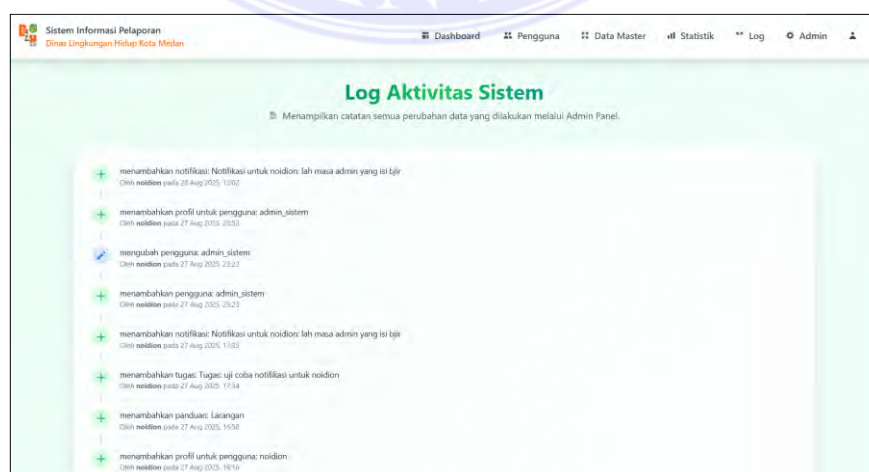
Halaman ini berfungsi sebagai pusat untuk analisis dan pengarsipan data. Admin dapat melihat statistik keseluruhan dan, yang terpenting, mengeksport semua data laporan ke dalam format file CSV. Fitur ekspor ini krusial untuk keperluan analisis data lebih lanjut, pembuatan laporan eksternal, dan pengarsipan digital.



Gambar 3.25 Halaman Statistik dan Ekspor Data

6. Halaman Log Aktivitas

Fitur Log Aktivitas berfungsi sebagai catatan audit atau "kotak hitam" sistem. Halaman ini menampilkan riwayat semua perubahan data penting yang terjadi, lengkap dengan informasi siapa yang melakukannya dan kapan. Tujuannya adalah untuk meningkatkan transparansi, keamanan, dan akuntabilitas dengan melacak semua aktivitas administratif di dalam sistem.



Gambar 3.26 Halaman Log Aktivitas

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan hasil perancangan dan pembangunan sistem, dapat disimpulkan bahwa Sistem Informasi Pelaporan Dinas Lingkungan Hidup berbasis web ini berhasil dikembangkan untuk menjawab permasalahan pada sistem pelaporan manual sebelumnya, seperti keterlambatan verifikasi laporan, pengarsipan yang tidak terstruktur, serta rendahnya transparansi dan akuntabilitas petugas dalam pelaporan hasil temuan di lapangan. Sistem yang dibangun telah berhasil mengimplementasikan tiga hak akses yang berbeda, yaitu untuk Petugas Lapangan, Atasan, dan Admin Sistem, di mana setiap peran memiliki fitur dan wewenang yang spesifik sesuai dengan alur kerja yang telah dirancang.

Sistem Informasi Pelaporan Dinas Lingkungan Hidup berbasis web ini berhasil mengatasi berbagai permasalahan mendasar pada sistem manual sebelumnya, seperti keterlambatan verifikasi, pengarsipan tidak terstruktur, dan rendahnya transparansi serta akuntabilitas pelaporan. Digitalisasi proses inti, mulai dari pembuatan laporan oleh petugas, verifikasi oleh atasan, hingga pengelolaan data dan pengguna oleh admin, telah diimplementasikan secara terpadu dan efisien. Fitur-fitur kunci seperti dashboard statistik, notifikasi real-time, pencatatan riwayat aktivitas, dan integrasi GPS mendukung monitoring yang lebih responsif, pelacakan data yang akurat, serta evaluasi kinerja yang objektif. Dengan demikian, sistem ini dapat menjadi solusi yang efektif untuk menggantikan alur kerja manual yang rentan terhadap keterlambatan, kehilangan data, dan kesulitan dalam rekapitulasi.

4.2 Saran

Meskipun sistem yang dibangun telah memenuhi tujuan awal, terdapat beberapa pengembangan lebih lanjut yang dapat diimplementasikan di masa depan untuk meningkatkan fungsionalitas dan pengalaman pengguna. Seperti peningkatan akurasi yang diperoleh melalui GPS dan unggah dan aplikasi mobile (*Mobile App*) yang akan sangat memudahkan petugas lapangan dalam membuat laporan secara langsung dari lokasi kegiatan tanpa perlu membuka *browser*.

DAFTAR PUSTAKA

- Anggarah, R., Nurma Feblia, K., Rizgita Amanda, K., Adzkiatul Mardiyah, Q., Chiara Amanda, F., Syarifatul Mursyidah, A., Villareal, Y., Valgiyos Aritonang, M. D., & Setiawan, Y. (2025). Studi Perbandingan Penerapan Pola Model-View-Controller (MVC) dalam Lima Framework Web Populer: Laravel, Django, Ruby on Rails, Asp.net MVC, Spring MVC. *Indonesian Journal of Computer Science and Engineering (IJCSE)*, 2(1), 16–22. <https://doi.org/10.70656/ijcse.v2i01.306>
- Arujisaputra, E. T. (2025). Penerapan Sistem Informasi untuk Meningkatkan Efisiensi Operasional dan Pengambilan Keputusan di Perusahaan. *Journal Scientific of Mandalika (Jsm)*, 6(3), 700–709. <https://doi.org/10.36312/10.36312/vol6iss3pp700-709>
- Daryono, D., Sutikno, A., & Widhoyoko, Y. P. (2024). Pengembangan Aplikasi Warditem Untuk Mendukung Digitalisasi Pemasaran Produk Ikm Di Desa Sijeruk. *Journal of Information System Management (JOISM)*, 6(1), 56–62. <https://doi.org/10.24076/joism.2024v6i1.1626>
- Febriyani, A., & Martanto, M. (2023). Rancang Bangun Aplikasi Penjualan Kebutuhan Pokok Berbasis Web Pada Toko Khansaa. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), 510–515. <https://doi.org/10.36040/jati.v7i1.6353>
- Hadianto, R. M. G. D., & Zubaidi, A. (2024). Pengembangan Back-End Pada Aplikasi Presensi Dan Website Pendataan UMKM NTB Mall. *Jurnal Begawe Teknologi Informasi (JBegaTI)*, 5(2), 170–181. <https://doi.org/10.29303/jbegati.v5i2.1197>
- Hartono, N. (2021). Comparison of Stored Procedures on Relational Database Management Systems. *Tech-E*, 4(2), 8–15. <https://doi.org/10.31253/te.v4i2.529>
- Kurniawan, H., Apriliah, W., Kurniawan, I., & Firmansyah, D. (2020). Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Penggajian Pada SMK Bina Karya Karawang. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 14(4), 13–23. <https://doi.org/10.35969/interkom.v14i4.58>

- Kus Indrani Listyoningrum, Danise Yunaini Fenida, & Nurhasan Hamidi. (2023). Inovasi Berkelanjutan dalam Bisnis: Manfaatkan Flowchart untuk Mengoptimalkan Nilai Limbah Perusahaan. *Jurnal Informasi Pengabdian Masyarakat*, 1(4), 100–112. <https://doi.org/10.47861/jipm-nalanda.v1i4.552>
- Muhamad Givari Ramadan, & Mardi Hardjianto. (2025). Pembuatan GH3RTZ Framework untuk Proses Reconnaissance dan Deteksi Celah Keamanan. *Jurnal Ticom: Technology of Information and Communication*, 13(2), 92–97. <https://doi.org/10.70309/ticom.v13i2.146>
- Novia Gardini, & Zulkarnaini Zulkarnaini. (2025). Peran Dinas Lingkungan Hidup pada Pengelolaan Sampah di Kota Pekanbaru. *JISPENDIORA Jurnal Ilmu Sosial Pendidikan Dan Humaniora*, 4(2), 349–360. <https://doi.org/10.56910/jispendiora.v4i2.2304>
- Perwitasari, I. D., Hendrawan, J., Panggabean, F. Y., & Raihansyah, M. (2024). Model UML Aplikasi Augmented Reality Pengenalan Desa Pertumbuhan. *Jurnal Minfo Polgan*, 13(2), 1887–1896. <https://doi.org/10.33395/jmp.v13i2.14301>
- Rasyaad, F., & Wibowo, S. A. (2025). Developing a Scalable Next.js Platform for IoT Monitoring Using Backend-as-a-Service with a Comparison of Supabase and Firebase. *2025 International Conference on Computer Sciences, Engineering, and Technology Innovation (ICoCSETI)*, 421–426. <https://doi.org/10.1109/ICoCSETI63724.2025.11018944>
- Sopriani, E., & Purwanto, H. (2023). Perancangan Sistem Informasi Persediaan Barang Berbasis Web Pada Pt. Xyz (Department It Infrastructure). *Jurnal Sistem Informasi Universitas Suryadarma*, 10(1), 127–138. <https://doi.org/10.35968/jsi.v10i1.993>
- Surachkaryadi, A., Prasetyo, A. Z., Sogen, V. F. P., Dengen, H. A., Irsyad, A., & Ibrahim, M. R. (2025). Pemetaan Sistem Informasi Geografis Daerah Rawan Banjir Di Wilayah Kota Bandung Jawa Barat Menggunakan Metode Skoring QGIS Tahun 2024. *Kreatif Teknologi Dan Sistem Informasi (KRETISI)*, 3(1), 15–22. <https://doi.org/10.30872/kretisi.v3i1.2132>

LAMPIRAN

Dokumentasi Kegiatan







Similarity Report ID: oid:29477:116585458

PAPER NAME

DIONIKXON MANURUNG.pdf

AUTHOR

DIONIKXON MANURUNG

WORD COUNT

8449 Words

CHARACTER COUNT

59230 Characters

PAGE COUNT

52 Pages

FILE SIZE

2.4MB

SUBMISSION DATE

Oct 14, 2025 8:21 AM GMT+7

REPORT DATE

Oct 14, 2025 8:22 AM GMT+7

● **7% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- 1% Publications database
- Crossref database
- Crossref Posted Content database

● **Excluded from Similarity Report**

- Submitted Works database
- Bibliographic material
- Cited material
- Abstract
- Small Matches (Less than 15 words)

Summary

	FAKULTAS TEKNIK PROGRAM STUDI TEKNIK INFORMATIKA	No. Dokumen	KP-03
	Jalan Kolam Nomor 1 Medan Estate, Medan 20223	No. Revisi	
	FORM BERITA ACARA BIMBINGAN KP	Berlaku Efektif	
		Halaman	

FORM BERITA ACARA BIMBINGAN KP

Nama Mahasiswa	:	Dionikxon Manurung
NIM	:	228160012
Judul Kegiatan KP	:	Perancangan Sistem Informasi Pelaporan Kegiatan Petugas Lapangan di Dinas Lingkungan Hidup
Tempat Pelaksanaan KP	:	Dinas Lingkungan Hidup Kota Medan
Dosen Pembimbing Akademik	:	Susilawati, S.T, M.Kom
Pembimbing Lapangan	:	Suci Lawati Yano, S.Psi, M.Psi

No	Tanggal	Uraian	Paraf Pembimbing
1	01/08/2025	Pengenalan Lingkungan Dinas dan Pembagian Tugas Ke Bidang	f
2	04/08/2025	Pengumpulan Data yang Dibutuhkan untuk sistem pelaporan.	f
3	06/08/2025	Perancangan <i>Flowchart</i>	f
4	08/08/2025	Perancangan DFD dan ERD	f
5	11/08/2025	Perancangan <i>Database</i>	f
6	13/08/2025	Perancangan UML	f
7	19/08/2025	Merancang <i>Mockup</i> Sistem Antarmuka Admin	f
8	22/08/2025	Merancang <i>Mockup</i> Sistem Antarmuka <i>User</i>	f
9	23/08/2025	Membuat Sistem Antarmuka Admin	f
10	25/08/2025	Membuat Sistem Antarmuka Admin	f
11	27/08/2025	Membuat Sistem Antarmuka <i>User</i>	f
12	29/08/2025	Membuat Sistem Antarmuka <i>User</i>	f
13	30/08/2025	Membuat Laporan Kerja Praktek	f
14	31/08/2025	Membuat Laporan Kerja Praktek	f

Medan, 28 Agustus 2025
Pembimbing Lapangan,



(Suci Lawati Yano, S.Psi, M.Psi)

	FAKULTAS TEKNIK PROGRAM STUDI TEKNIK INFORMATIKA	No. Dokumen	KP-04 B
	Jalan Kolam Nomor 1 Medan Estate, Medan 20223	No. Revisi	
	FORM PENILAIAN PEMBIMBING LAPANGAN	Berlaku Efektif	
		Halaman	

FORM PENILAIAN PEMBIMBING LAPANGAN

Sebagai Pembimbing Lapangan Kerja Praktek mahasiswa :

Nama : Dionikxon Manurung

NIM : 228160012

Setelah mengikuti pelaksanaan Kerja Praktek mahasiswa tersebut, memberikan NILAI:

ASPEK PENILAIAN	DESKRIPSI ASPEK PENILAIAN	BOBOT	SKOR (0-100)	NILAI (BOBOT * SKOR)
Komunikasi	Kemampuan untuk menyampaikan informasi, mendengarkan orang lain, berkomunikasi secara efektif, dan memberikan respon positif yang mendorong komunikasi terbuka	20%	92	18,4
Kerjasama	Kemampuan menjalin kerjasama dalam tim, peka akan kebutuhan orang lain dan memberikan kontribusi dalam aktivitas tim untuk mencapai tujuan dan hasil yang positif	15%	93	13,9
Inisiatif dan Kreativitas	Kemampuan merespon masalah secara proaktif dan gigih, menjajaki kesempatan yang ada, melakukan sesuatu tanpa disuruh guna mengatasi hambatan, yang ditampilkan secara motorik/verbal (yang berkonsekuen tindakan)	15%	93	13,9
Disiplin Kerja dan Adaptasi	Kemauan untuk mematuhi aturan yang berlaku dan dapat menyesuaikan perilaku agar dapat bekerja secara efektif dan efisien saat adanya informasi baru, perubahan situasi atau kondisi lingkungan kerja yang berbeda	20%	92	18,4
Penyelesaian Tugas	Penyelesaian setiap tugas yang diberikan oleh Pembimbing Lapangan. Penilaian berdasarkan persentase penyelesaian tugas	30%	95	28,5
TOTAL NILAI :				93,1

Pembimbing Lapangan

Nama : Suci Lawati Yano, S.Psi, M.Psi

Medan, 28 Agustus 2025

NIP : 198504042010012031

Jabatan : Pengawas Lingkungan Hidup Muda



(Suci Lawati Yano, S.Psi, M.Psi)

	FAKULTAS TEKNIK PRORGRAM STUDI TEKNIK INFORMATIKA	No. Dokumen	KP-04 A
	Jalan Kolam Nomor 1 Medan Estate, Medan 20223	No. Revisi	
	FORM PENILAIAN PEMBIMBING KP	Berlaku Efektif	
		Halaman	

FORM PENILAIAN PEMBIMBING KP

Nama : Dionikxon Manurung

NIM : 228160012

ASPEK PENILAIAN	KOMPONEN	BOBOT	SKOR (0-100)	NILAI (BOBOT*SKOR)
Buku Laporan Pelaksanaan KP	Aturan penulisan dan Tatabahasa	15%	90	13,5
	Latar Belakang dan Tujuan	15%	90	13,5
	Uraian Perumusan Masalah dan Pembahasan Hasil	40%	90	36
Presentasi Hasil KP	Kemampuan menyelesaikan pekerjaan	10%	90	9
	Kesesuaian hasil/produk dengan tujuan	10%	90	9
	Kemampuan Presentasi	10%	90	9
TOTAL NILAI				90

Nilai Akhir = (40% x Nilai Pembimbing Lapangan) + (60% x Nilai Pembimbing Akademik)

$$= \dots 40\% \times 93,1 \dots + \dots 60\% \times 90 \dots = \dots 91,2 \dots$$

Catatan untuk perhitungan nilai :

80 < NSM	A
70 < NSM ≤ 80	B+
65 < NSM ≤ 70	B
60 < NSM ≤ 65	C+
50 < NSM ≤ 60	C
40 < NSM ≤ 50	D
NSM ≤ 40	E

Medan, 14 Oktober 2025
Pembimbing Akademik



(Susilawati, S.Kom, M.Kom)
NIDN. 0126068702



UNIVERSITAS MEDAN AREA

FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan, 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan, 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 210/FT.6/01.10/VII/2025
Lamp : -
Hal : Pembimbing Kerja Praktek/T.A

21 Juli 2025

Yth. Pembimbing Kerja Praktek
Susilawati, S. Kom, M. Kom
Di
Tempat

Dengan hormat,
Sehubungan telah dipenuhinya persyaratan untuk memperoleh Kerja Praktek dari mahasiswa :

NO	NAMA MAHASISWA	NPM	JURUSAN
1	Dionikxon Manurung	228160012	Teknik Informatika

Maka dengan hormat kami mengharapkan kesediaan saudara :

Susilawati, S. Kom, M. Kom (Sebagai Pembimbing)

Dimana Kerja Praktek tersebut dengan judul :

“Perancangan Sistem Informasi Pelaporan Kegiatan Petugas Lapangan di Dinas Lingkungan Hidup”

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,

Dr. Eng. Supriatno, ST, MT



UNIVERSITAS MEDAN AREA

FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan, 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan, 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 214/FT.6/01.10/VII/2025
Lamp : -
Hal : 1Kerja Praktek

21 Juli 2025

Yth. Kepala Badan Riset dan Inovasi Daerah Kota Medan
Jln. Jenderal Besar A.H. Nasution No.32
Di
Medan

Dengan hormat,

Dengan surat ini kami mohon kesediaan Bapak/Ibu kiranya berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	N A M A	N P M	PROG. STUDI	JUDUL
1	Muhammad Lukmannulhakim	228160010	Teknik Informatika	Sistem Informasi Pengelolaan Data Pengaduan Masyarakat terhadap Masalah Lingkungan Berbasis Web
2	Dionikxon Manurung	228160012	Teknik Informatika	Perancangan Sistem Informasi Pelaporan Kegiatan Petugas Lapangan di Dinas Lingkungan Hidup
3	Nora Irawani Siregar	228160013	Teknik Informatika	Perancangan Sistem Informasi Pengarsipan Dokumen Digital Surat Masuk dan Surat Keluar
4	Puderta Sinulingga	228160068	Teknik Informatika	Analisis Sentimen Media Sosial Terkait Isu Lingkungan di Kota Medan Menggunakan NLP
5	Abdivani Malkia Dharma Girsang	228160076	Teknik Informatika	Pengembangan SIG Berbasis Web untuk Penempatan dan Pencarian TPS di Kota Medan

Untuk melaksanakan Kerja Praktek pada Dinas Lingkungan Hidup Kota Medan di Jln. Pinang Baris, No. 114

Perlu kami jelaskan bahwa Kerja Praktek tersebut adalah semata-mata untuk tujuan ilmiah. Kami mohon kiranya juga dapat diberikan kemudahan untuk terlaksananya Kerja Praktek ini.

Demikian kami sampaikan, atas kerjasama yang baik diucapkan terima kasih.

Dekan,



Tembusan :

1. Ka. BPMPP
2. Mahasiswa
3. File



PEMERINTAH KOTA MEDAN
DINAS LINGKUNGAN HIDUP

Jalan Pinang Baris No. 114, Medan Sunggal, Medan, Sumatera Utara 20127
Telepon (061) 8458994 - 8452022, Faksimile (061) 8452022,
Laman dlh.medan.go.id, Pos-el dlh@medan.go.id

SURAT KETERANGAN
Nomor: 000.9/2025

Yang bertanda tangan di bawah ini:

Nama : Melvi Marlabayana, S.T., M.Si
NIP : 197707061996022001
Pangkat/Gol.Ruang : Pembina Tingkat I(IV/b)
Jabatan : Kepala Dinas Lingkungan Hidup Kota Medan

Dengan ini menerangkan bahwa mahasiswa yang tersebut di bawah ini:

NO	NAMA	NIM	JURUSAN	FAKULTAS
1.	Muhammad Lukmanul hakim	228160010	Teknik Informatika	Teknik Universitas Medan Area
2.	Dionikxon Manurung	228160012	Teknik Informatika	Teknik Universitas Medan Area
3.	Nora Irawan Siregar	228160013	Teknik Informatika	Teknik Universitas Medan Area
4.	Abdivani Malkia Dharma Girsang	228160076	Teknik Informatika	Teknik Universitas Medan Area
5.	Puderta Sinulingga	228160068	Teknik Informatika	Teknik Universitas Medan Area

Telah selesai melaksanakan Praktek Kerja Lapangan (PKL) mulai Tanggal 1 Agustus 2025 sampai 31 Agustus 2025 pada Dinas Lingkungan Hidup Kota Medan.

Demikian surat keterangan ini di perbuat untuk dapat dipergunakan seperlunya

Medan, 03 September 2025
Kepala Dinas Lingkungan Hidup,

Melvi Marlabayana, S.T., M.Si
Pembina Tk. I(IV/b)
NIP. 197707061996022001

Code Program

Models.py

```
from django.db import models
from django.contrib.auth.models import User
from django.utils.translation import gettext_lazy as _
from django.utils import timezone

#role akses
class Profil(models.Model):
    class Role(models.TextChoices):
        ADMIN = 'ADMIN', _('Admin')
        ATASAN = 'ATASAN', _('Atasan')
        PETUGAS = 'PETUGAS', _('Petugas Lapangan')

    user = models.OneToOneField(User,
on_delete=models.CASCADE)

    role = models.CharField(max_length=50,
choices=Role.choices, default=Role.PETUGAS)

    nomor_telepon = models.CharField(max_length=20,
blank=True, null=True)
    nip = models.CharField(max_length=50,
blank=True, null=True, unique=True)

    def __str__(self):
        return f"{self.user.username} - {self.get_role_display()}"

class JenisKegiatan(models.Model):
    nama_kegiatan = models.CharField(max_length=255, unique=True)
    deskripsi = models.TextField(blank=True, null=True)

    def __str__(self):
        return self.nama_kegiatan

class Wilayah(models.Model):
    nama_wilayah = models.CharField(max_length=255, unique=True) # Contoh: Kecamatan Medan Baru
    kategori = models.CharField(max_length=100, blank=True, null=True) # Contoh: Kecamatan/Kelurahan

    def __str__(self):
        return self.nama_wilayah

#Laporan Kegiatan Petugas Lapangan
class Laporan(models.Model):
    class StatusLaporan(models.TextChoices):
        DRAFT = 'DRAFT', _('Draft')
        MENUNGGU_VERIFIKASI = 'MENUNGGU_VERIFIKASI', _('Menunggu Verifikasi')
        DISETUJUI = 'DISETUJUI', _('Disetujui')
        DITOLAK = 'DITOLAK', _('Ditolak')

    petugas = models.ForeignKey(User,
on_delete=models.PROTECT,
related_name='laporan_dibuat')

    judul_kegiatan = models.CharField(max_length=255)
    jenis_kegiatan = models.ForeignKey(JenisKegiatan,
on_delete=models.SET_NULL, null=True)

    lokasi_text = models.CharField(max_length=255, help_text="Contoh: Jl. Gatot Subroto No. 12")
    lokasi_gps_lat = models.FloatField(blank=True, null=True, verbose_name="GPS Latitude")
    lokasi_gps_lon = models.FloatField(blank=True, null=True, verbose_name="GPS Longitude")
    lokasi_alamat_otomatis = models.TextField(blank=True, null=True, help_text="Alamat yang dihasilkan secara otomatis dari koordinat GPS.")
    waktu_kegiatan = models.DateTimeField()

    hasil_temuan = models.TextField()
    dokumentasi_foto = models.ImageField(upload_to='dokumentasi/', blank=True, null=True, help_text="Upload foto bukti kegiatan")
    dokumen_pendukung = models.FileField(upload_to='dokumen_pendukung/', blank=True, null=True, help_text="Upload Berita Acara atau dokumen lain")

    status = models.CharField(max_length=50, choices=StatusLaporan.choices, default=StatusLaporan.DRAFT)
    catatan_atasan = models.TextField(blank=True, null=True, help_text="Diisi oleh atasan saat verifikasi")
    atasan_verifikator = models.ForeignKey(User, on_delete=models.SET_NULL, null=True, blank=True, related_name='laporan_diverifikasi')

    waktu_dibuat = models.DateTimeField(auto_now_add=True)
    waktu_diperbarui = models.DateTimeField(auto_now=True)

    def __str__(self):
```

Views.py

```
import csv
from django.http import HttpResponseRedirect
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth import authenticate, login as auth_login, logout as auth_logout
from django.contrib.auth.decorators import login_required
from django.utils import timezone
from datetime import timedelta
import json
from .forms import LaporanForm, SaranForm, TugasForm, AgendaForm
from .models import Laporan, Tugas, Panduan, Saran, Kehadiran, Notifikasi, Agenda
from django.db.models import Q, Count
from django.contrib import messages
from django.contrib.auth import update_session_auth_hash
from django.contrib.auth.models import User
from django.contrib.admin.models import LogEntry
from .forms import LaporanForm, UserUpdateForm, ProfilUpdateForm, CustomPasswordChangeForm

#Halaman Login ---
def login_view(request):
    if request.user.is_authenticated:
        return redirect('dashboard')

    if request.method == 'POST':
        username_input = request.POST.get('username')
        password_input = request.POST.get('password')

        user = authenticate(request, username=username_input, password=password_input)

        if user is not None:
            auth_login(request, user)
            return redirect('dashboard')
        else:
            return render(request, 'pelaporan/login.html')

    return render(request, 'pelaporan/login.html')

def logout_view(request):
    """Fungsi untuk melakukan logout pengguna."""
    auth_logout(request)
    return redirect('login')

@login_required
def buat_laporan_view(request, tugas_id=None):
    tugas = None
    initial_data = {}

    if tugas_id:
        tugas = get_object_or_404(Tugas, id=tugas_id, petugas_ditugaskan=request.user)
        initial_data = {
            'judul_kegiatan': tugas.judul_tugas,
            'hasil_temuan': tugas.deskripsi,
        }

    if request.method == 'POST':
        form = LaporanForm(request.POST, request.FILES)
        if form.is_valid():
            laporan = form.save(commit=False)
            laporan.petugas = request.user
            laporan.status = 'MENUNGGU_VERIFIKASI'
            laporan.save()

            semua_atasan = User.objects.filter(profil__role='ATASAN')
            for atasan in semua_atasan:
                Notifikasi.objects.create(
                    penerima=atasan,
                    pesan=f"Laporan baru {laporan.judul_kegiatan}" dari {laporan.petugas.username} perlu diverifikasi.",
                    link_terkait=f"/verifikasi/laporan/{laporan.id}/"
                )

            if tugas:
                tugas.laporan_terkait = laporan
                tugas.status = 'SELESAI'
                tugas.save()
                return redirect('tugas_detail', tugas_id=tugas.id)

            return redirect('dashboard')
        else:
            form = LaporanForm(initial=initial_data)

    context = {
        'form': form,
        'tugas': tugas
    }
    return render(request, 'pelaporan/buat_laporan.html', context)

#View Utama
@login_required
def dashboard_view(request):
    if hasattr(request.user, 'profil'):
        if request.user.profil.role == 'ATASAN':
            return atasan_dashboard_view(request)
```


<pre> return f"Laporan: {self.judul_kegiatan} oleh {self.petugas.username}" # Model untuk Manajemen Tugas class Tugas(models.Model): class StatusTugas(models.TextChoices): BARU = 'BARU', _('Tugas Baru') DIKERJAKAN = 'DIKERJAKAN', _('Sedang Dikerjakan') SELESAI = 'SELESAI', _('Selesai') atasan = models.ForeignKey(User, on_delete=models.PROTECT, related_name='tugas_diberikan') petugas_ditugaskan = models.ForeignKey(User, on_delete=models.PROTECT, related_name='tugas_diterima') judul_tugas = models.CharField(max_length=255) deskripsi = models.TextField() deadline = models.DateTimeField() status = models.CharField(max_length=50, choices=StatusTugas.choices, default=StatusTugas.BARU) laporan_terkait = models.OneToOneField(Laporan, on_delete=models.SET_NULL, null=True, blank=True) spt_file = models.FileField(upload_to='spt/', blank=True, null=True, verbose_name="File SPT (Opsional)") waktu_dibuat = models.DateTimeField(auto_now_add=True) def __str__(self): return f"Tugas: {self.judul_tugas} untuk {self.petugas_ditugaskan.username}" class Panduan(models.Model): judul = models.CharField(max_length=255) konten = models.TextField(help_text="Anda bisa menggunakan format HTML sederhana di sini jika perlu.") urutan = models.PositiveIntegerField(default=0, help_text="Panduan dengan urutan lebih kecil akan tampil lebih dulu.") class Meta: ordering = ['urutan'] verbose_name_plural = "Panduan Lapangan" def __str__(self): return self.judul class Saran(models.Model): pengirim = models.ForeignKey(User, on_delete=models.CASCADE, related_name='saran_dikirim') subjek = models.CharField(max_length=255) isi_saran = models.TextField() waktu_kirim = models.DateTimeField(auto_now_add=True) class Meta: ordering = ['-waktu_kirim'] verbose_name_plural = "Kotak Saran" def __str__(self): return f"Saran dari {self.pengirim.username}: {self.subjek}" class Kehadiran(models.Model): petugas = models.ForeignKey(User, on_delete=models.CASCADE, related_name='kehadiran') waktu_check_in = models.DateTimeField() waktu_check_out = models.DateTimeField(null=True, blank=True) kegiatan = models.CharField(max_length=255, help_text="Deskripsi singkat kegiatan saat check- in") class Meta: ordering = ['-waktu_check_in'] verbose_name_plural = "Riwayat Kehadiran" def __str__(self): return f"Kehadiran {self.petugas.username} pada {self.waktu_check_in.strftime('%d %b %Y')}}" @property def durasi(self): if self.waktu_check_out: return self.waktu_check_out - self.waktu_check_in return timezone.now() - self.waktu_check_in class Notifikasi(models.Model): penerima = models.ForeignKey(User, on_delete=models.CASCADE, related_name='notifikasi') pesan = models.CharField(max_length=255) sudah_dibaca = models.BooleanField(default=False) waktu_buat = models.DateTimeField(auto_now_add=True) link_terkait = models.URLField(blank=True, null=True) </pre>	<pre> return petugas_dashboard_view(request) #Dashboard Petugas @login_required def petugas_dashboard_view(request): semua_laporan = Laporan.objects.filter(petugas=request.user) laporan_terbaru = semua_laporan.order_by('- waktu_dibuat')[:5] semua_tugas = Tugas.objects.filter(petugas_ditugaskan=request.user) tugas_aktif = semua_tugas.exclude(status='SELESAI').order_by('deadlin e')[:5] jumlah_total_laporan = semua_laporan.count() jumlah_tugas_aktif = semua_tugas.exclude(status='SELESAI').count() jumlah_laporan_disetujui = semua_laporan.filter(status='DISETUIJUI').count() context = { 'laporan_terbaru': laporan_terbaru, 'tugas_aktif': tugas_aktif, 'jumlah_total_laporan': jumlah_total_laporan, 'jumlah_tugas_aktif': jumlah_tugas_aktif, 'jumlah_laporan_disetujui': jumlah_laporan_disetujui, } return render(request, 'pelaporan/dashboard.html', context) #Dashboard Atasan @login_required def atasan_dashboard_view(request): today = timezone.now().date() laporan menunggu = Laporan.objects.filter(status='MENUNGGU_VERIFIKASI') laporan_ditetujui_hari_ini = Laporan.objects.filter(status='DISETUIJUI', waktu_diperbarui__date=today).count() jumlah_petugas = User.objects.filter(profil__role='PETUGAS').count() chart_labels = [] chart_data = [] for i in range(6, -1, -1): hari = today - timedelta(days=i) jumlah_laporan_per_hari = Laporan.objects.filter(waktu_dibuat__date=hari).count() chart_labels.append(hari.strftime("%d %b")) # Format: 28 Agu chart_data.append(jumlah_laporan_per_hari) context = { 'laporan_menunggu_verifikasi': laporan_menunggu, 'laporan_menunggu_count': laporan_menunggu.count(), 'laporan_ditetujui_hari_ini': laporan_ditetujui_hari_ini, 'jumlah_petugas': jumlah_petugas, 'chart_labels': json.dumps(chart_labels), 'chart_data': json.dumps(chart_data), } return render(request, 'pelaporan/atasan_dashboard.html', context) @login_required def laporan_detail_view(request, laporan_id): laporan = get_object_or_404(Laporan, id=laporan_id, petugas=request.user) context = { 'laporan': laporan } return render(request, 'pelaporan/laporan_detail.html', context) @login_required def tugas_saya_view(request): tugas_list = Tugas.objects.filter(petugas_ditugaskan=request.user).o rder_by('deadline') context = { 'semua_tugas': tugas_list } return render(request, 'pelaporan/tugas_saya.html', context) @login_required def tugas_detail_view(request, tugas_id): tugas = get_object_or_404(Tugas, id=tugas_id, petugas_ditugaskan=request.user) context = { 'tugas': tugas } return render(request, 'pelaporan/tugas_detail.html', context) @login_required def ubah_status_tugas_view(request, tugas_id): tugas = get_object_or_404(Tugas, id=tugas_id, petugas_ditugaskan=request.user) new_status = request.POST.get('status') valid_statuses = [status[0] for status in Tugas.StatusTugas.choices] </pre>
---	---

<pre> class Meta: ordering = ['-waktu_buat'] verbose_name_plural = "Notifikasi" def __str__(self): return f"Notifikasi untuk {self.penerima.username}: {self.pesan}" class Agenda(models.Model): pembuat = models.ForeignKey(User, on_delete=models.CASCADE, related_name='agenda_dibuat') judul = models.CharField(max_length=255) deskripsi = models.TextField(blank=True, null=True) waktu_mulai = models.DateTimeField() waktu_selesai = models.DateTimeField(blank=True, null=True) lokasi = models.CharField(max_length=255, blank=True, null=True) class Meta: ordering = ['-waktu_mulai'] verbose_name_plural = "Agenda Kegiatan" def __str__(self): return self.judul </pre>	<pre> if new_status in valid_statuses: tugas.status = new_status tugas.save() return redirect('tugas_detail', tugas_id=tugas.id) @login_required def profil_view(request): if request.method == 'POST': user_form = UserUpdateForm(request.POST, instance=request.user) profil_form = ProfilUpdateForm(request.POST, instance=request.user.profil) password_form = CustomPasswordChangeForm(request.user, request.POST) password_change_attempted = request.POST.get('old_password') if user_form.is_valid() and profil_form.is_valid(): user_form.save() profil_form.save() messages.success(request, 'Perubahan profil berhasil disimpan.') if password_change_attempted: if password_form.is_valid(): user = password_form.save() update_session_auth_hash(request, user) messages.success(request, 'Password Anda juga berhasil diubah.') else: messages.error(request, 'Gagal mengubah password. Silakan periksa kembali isian Anda.') context = {'user_form': user_form, 'profil_form': profil_form, 'password_form': password_form} return render(request, 'pelaporan/profil.html', context) else: return redirect('profil') user_form = UserUpdateForm(instance=request.user) profil_form = ProfilUpdateForm(instance=request.user.profil) password_form = CustomPasswordChangeForm(request.user) context = { 'user_form': user_form, 'profil_form': profil_form, 'password_form': password_form } return render(request, 'pelaporan/profil.html', context) @login_required def panduan_view(request): semua_panduan = Panduan.objects.all() context = { 'semua_panduan': semua_panduan } return render(request, 'pelaporan/panduan.html', context) @login_required def kotak_saran_view(request): if request.method == 'POST': form = SaranForm(request.POST) if form.is_valid(): saran = form.save(commit=False) saran.pengirim = request.user saran.save() messages.success(request, 'Terima kasih! Saran Anda telah berhasil dikirim.') return redirect('kotak_saran') else: form = SaranForm() context = { 'form': form } return render(request, 'pelaporan/kotak_saran.html', context) @login_required def riwayat_kehadiran_view(request): kehadiran_aktif = Kehadiran.objects.filter(petugas=request.user, waktu_check_out__isnull=True).first() if request.method == 'POST': action = request.POST.get('action') if action == 'check_in' and not kehadiran_aktif: kegiatan = request.POST.get('kegiatan', 'Kegiatan Lapangan') kehadiran.objects.create(petugas=request.user, waktu_check_in=timezone.now(), kegiatan=kegiatan) messages.success(request, 'Anda berhasil melakukan Check-in.') elif action == 'check_out' and kehadiran_aktif: </pre>
---	--

	<pre> kehadiran_aktif.waktu_check_out = timezone.now() kehadiran_aktif.save() messages.success(request, 'Anda berhasil melakukan Check-out.') return redirect('riwayat_kehadiran') riwayat = Kehadiran.objects.filter(petugas=request.user) context = { 'riwayat_kehadiran': riwayat, 'kehadiran_aktif': kehadiran_aktif } return render(request, 'pelaporan/riwayat_kehadiran.html', context) </pre>
Buat_laporan.html <pre> {% extends 'pelaporan/base.html' %} {% load widget_tweaks %} {% block title %}Buat Laporan Baru{% endblock %} {% block content %} <div class="min-h-screen py-8 px-4"> <div class="max-w-6xl mx-auto form-container p- 8 relative"> <!-- Decorative background icons --> <div class="form-decorations"></div> <div class="form-header"> <h2 class="form-title"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"> <path fill-rule="evenodd" d="M3 4a1 1 0 011-1h3a1 1 0 011v3a1 1 0 01-1 1H4a1 1 0 01-1v4zM3 13a1 1 0 011-1h3a1 1 0 011v3a1 1 0 01-1 1H4a1 1 0 01-1v4zM12 4a1 1 0 011-1h3a1 1 0 011v3a1 1 0 01-1 1H13a1 1 0 01-1v4zM12 13a1 1 0 011-1h3a1 1 0 011v3a1 1 0 01-1 1H13a1 1 0 01- 1v3z" clip-rule="evenodd" /> </svg> Formulir Laporan Kegiatan <svg class="icon" fill="currentColor" viewBox="0 0 20 20"> <path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zM6-3a2 2 0 11-4 0 2 2 0 014 0zM-2 4a5 5 0 00-4.546 2.916A5.986 5.986 0 0010 16a5.986 5.986 0 004.546-2.084A5 5 0 010 11z" clip-rule="evenodd" /> </svg> </h2> <p class="form-subtitle">Dinas Lingkungan Hidup • Sistem Pelaporan Terintegrasi</p> </div> <form method="POST" enctype="multipart/form-data"> <!-- csrf_token --> <!-- PERUBAHAN: Gunakan Grid untuk 2 Kolom --> <div class="grid grid-cols-1 md:grid- cols-2 gap-x-8"> <!-- Kolom Kiri --> <div class="space-y-6"> <div> <div class="form-field"> <label for="{{ field.id_for_label }}" class="field-label"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M3 4a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM0 4a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM0 12a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM0 12a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z" clip- rule="evenodd"></path></svg> </label> <div class="form- input" >> <div class="error-message"><div class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zM6-3a2 2 0 11-4 0 2 2 0 014 0zM-2 4a5 5 0 00-4.546 2.916A5.986 5.986 0 0010 16a5.986 5.986 0 004.546-2.084A5 5 0 010 11z" clip- rule="evenodd"></path></div> <div class="as_text" >> </div> </div> </div> <div> <div class="form-field"> <label for="{{ field.id_for_label }}" class="field-label"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M3 4a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM0 4a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM0 12a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM0 12a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z" clip- rule="evenodd"></path></div> </label> <div class="form- input" >> <div class="error-message"><div class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zM6-3a2 2 0 11-4 0 2 2 0 014 0zM-2 4a5 5 0 00-4.546 2.916A5.986 5.986 0 0010 16a5.986 5.986 0 004.546-2.084A5 5 0 010 11z" clip- rule="evenodd"></path></div> <div class="as_text" >> </div> </div> </div> </div> </div> </form> </pre>	Buat_tugas.html <pre> {% extends 'pelaporan/base.html' %} {% load widget_tweaks %} {% block title %}Buat Tugas Baru{% endblock %} {% block content %} <div class="max-w-3xl mx-auto"> <div class="text-center mb-10"> <h1 class="text-4xl font-bold leading-tight"> Buat Tugas Baru </h1> <p class="text-gray-600 mt-2 flex items-center justify-center"> <svg class="w-4 h-4 mr-2 text-gray-400" fill="currentColor" viewBox="0 0 20 20"> <path d="M17.414 2.586a2 2 0 00-2.828 0L7 10.172V13h2.828L7.586 7.586a2 2 0 00-2.828z" /> <path fill-rule="evenodd" d="M2 6a2 2 0 012-2h4a1 1 0 012H4v10h10v-4a1 1 0 012 0v4a2 2 0 01-2 2H4a2 2 0 01-2-2V6z" clip-rule="evenodd" /> </svg> Isi formulir di bawah ini untuk mendelegasikan tugas baru kepada petugas. </p> </div> <div class="bg-white/80 backdrop-blur-sm p-8 rounded-2xl shadow-xl border border-gray-200/50"> <!-- PERBAIKAN: Tambahkan method="POST" dan enctype --> <form method="POST" enctype="multipart/form- data"> <!-- csrf_token --> <div class="space-y-6"> <div> <div class="form-field"> <label for="{{ field.id_for_label }}" class="block text-gray- 700 text-sm font-semibold mb-2 ml-1"> <div class="form- input" >> <div class="error-message"><div class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zM6-3a2 2 0 11-4 0 2 2 0 014 0zM-2 4a5 5 0 00-4.546 2.916A5.986 5.986 0 0010 16a5.986 5.986 0 004.546-2.084A5 5 0 010 11z" clip- rule="evenodd"></path></div> <div class="as_text" >> </div> </div> </div> </div> <div> <div class="form-field"> <label for="{{ field.id_for_label }}" class="block text-gray- 700 text-sm font-semibold mb-2 ml-1"> <div class="form- input" >> <div class="error-message"><div class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zM6-3a2 2 0 11-4 0 2 2 0 014 0zM-2 4a5 5 0 00-4.546 2.916A5.986 5.986 0 0010 16a5.986 5.986 0 004.546-2.084A5 5 0 010 11z" clip- rule="evenodd"></path></div> <div class="as_text" >> </div> </div> </div> </div> </div> </form> </div> </div> <div class="mt-8 pt-6 border-t border-gray- 200/80 flex flex-col-reverse sm:flex-row items-center sm:space-x-4"> Batal </pre>

<pre> 2H4a1 1 0 01-1-1z" clip- rule="evenodd"></path></svg> {{ field.label }} </label> {{ field add_class:"form- input" }} {% if field.errors %}<div class="error-message"><svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0v6a1 1 0 00-1-1z" clip- rule="evenodd" /></svg>{{ field.errors.as_text }}</div>{% endif %} </div> {% endwhile %} {% with field=form.waktu_kegiatan %} <div class="form-field"> <label for="{{ field.id_for_label }}" class="field-label"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M6 2a1 1 0 00-1 1v1H4a2 2 0 00-2 2v10a2 2 0 002 2h12a2 2 0 002-2V6a2 2 0 00-2-2h- 1V3a1 1 0 10-2 0v1H7V3a1 1 0 00-1-1zm0 5a1 1 0 00 2h8a1 1 0 100-2H6z" clip- rule="evenodd"></path></svg> {{ field.label }} </label> {{ field add_class:"form- input" }} {% if field.errors %}<div class="error-message"><svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0v6a1 1 0 00-1-1z" clip- rule="evenodd" /></svg>{{ field.errors.as_text }}</div>{% endif %} </div> {% endwhile %} {% with field=form.lokasi_text %} <div class="form-field"> <label for="{{ field.id_for_label }}" class="field-label"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M5.05 4.05a7 7 0 119.9 9.9L10 18.9L-4.95-4.95a7 7 0 10-9.92M10 11a2 2 0 100-4 2 2 0 000 4z" clip-rule="evenodd"></path></svg> {{ field.label }} </label> {{ field add_class:"form- input" }} {% if field.errors %}<div class="error-message"><svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0v6a1 1 0 00-1-1z" clip- rule="evenodd" /></svg>{{ field.errors.as_text }}</div>{% endif %} </div> {% endwhile %} <!-- Kolom Kanan --> <div class="space-y-6"> {% with field=form.hasil_temuan %} <div class="form-field"> <label for="{{ field.id_for_label }}" class="field-label"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M4 4a2 2 0 012-2h4.586A2 2 0 012 2.586L15.414 6A2 2 0 016 7.414V16a2 2 0 01-2 2H6a2 2 0 01-2-2V4zm2 6a1 1 0 011-1h6a1 1 0 110 2H7a1 1 0 01-1-1zm1 3a1 1 0 100 2h6a1 1 0 100-2H7z" clip- rule="evenodd"></path></svg> {{ field.label }} </label> {{ field add_class:"form- input" attr:"rows:5" }} {% if field.errors %}<div class="error-message"><svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0v6a1 1 0 00-1-1z" clip- rule="evenodd" /></svg>{{ field.errors.as_text }}</div>{% endif %} </div> {% endwhile %} {% with field=form.dokumentasi_foto %} <div class="form-field"> <label for="{{ field.id_for_label }}" class="field-label"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M4 3a2 2 0 00-2 2v10a2 2 0 002 2h12a2 2 0 002-2V5a2 2 0 00-2-2H4zm12 12H4l4-8 3 6 2-4 3 6z" clip-rule="evenodd"></path></svg> {{ field.label }} </label> </pre>	<pre> <button type="submit" class="w-full sm:w-auto bg-gradient-to-r from-emerald-500 to-teal-500 hover:from-emerald-600 hover:to-teal-600 text-white font-bold py-3 px-6 rounded-xl shadow-lg hover:shadow- xl transform hover:-translatey-0.5 transition-all duration-300 flex items-center justify-center"> <svg class="w-5 h-5 mr-2" fill="currentColor" viewBox="0 0 20 20"> <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000 16zm1-11a1 1 0 10-2 0v2H7a1 1 0 100 2h2v2a1 1 0 102 0v-2h2a1 1 0 100-2h- 2V7z" clip-rule="evenodd" /> </svg> Simpan Tugas </button> </div> </form> </div> {% endblock %} </pre>
	<pre> Verifikasi_laporan.html {% extends 'pelaporan/base.html' %} {% block title %}Verifikasi Detail Laporan{% endblock %} {% block content %} <div class="max-w-4xl mx-auto bg-white rounded-lg shadow-md p-8"> <div class="flex justify-between items-start mb-6"> <div> <h2 class="text-3xl font-bold text-gray- 800">{{ laporan.judul_kegiatan }}</h2> <p class="text-sm text-gray-500"> Dilaporkan oleh: {{ laporan.petugas.username }} pada {{ laporan.waktu_dibuat date:"d M Y, H:i" }} {% if laporan.lokasi_gps_lat and laporan.lokasi_gps_lon %} - di {{ laporan.lokasi_gps_lat floatformat:4 }} {{ laporan.l okasi_gps_lon floatformat:4 }} {% else %} - di (lokasi gps tidak tersedia) {% endif %} </p> </div> <div> <!-- Label Status --> Menunggu Verifikasi </div> </div> <div class="border-t border-gray-200 pt-6"> <dl class="grid grid-cols-1 md:grid-cols-2 gap- x-6 gap-y-8"> <div class="sm:col-span-1"> <dt class="text-sm font-medium text- gray-500">Jenis Kegiatan</dt> <dd class="mt-1 text-lg text-gray- 900">{{ laporan.jenis_kegiatan.nama_kegiatan }}</dd> </div> <div class="sm:col-span-1"> <dt class="text-sm font-medium text- gray-500">Waktu Kegiatan</dt> <dd class="mt-1 text-lg text-gray- 900">{{ laporan.waktu_kegiatan date:"d F Y, P" }}</dd> </div> <div class="sm:col-span-2"> <dt class="text-sm font-medium text- gray-500">Lokasi (Sesuai Pengamatan)</dt> <dd class="mt-1 text-lg text-gray- 900">{{ laporan.lokasi_text }}</dd> </div> </div> <!-- PERUBAHAN: Tambahkan bagian untuk Lokasi GPS Otomatis --> <div class="sm:col-span-2"> <dt class="text-sm font-medium text- gray-500">Lokasi (Otomatis via GPS)</dt> {% if laporan.lokasi_alamat_otomatis %} <dd class="mt-1 text-lg text-gray- 900">{{ laporan.lokasi_alamat_otomatis }}</dd> <dd class="mt-1 text-sm text-gray- 500"> (Lat: {{ laporan.lokasi_gps_lat floatformat:5 }}, Lon: {{ laporan.lokasi_gps_lon floatformat:5 }}) </dd> {% elif laporan.lokasi_gps_lat and laporan.lokasi_gps_lon %} <dd class="mt-1 text-lg text-gray- 900"> Lat: {{ laporan.lokasi_gps_lat floatformat:5 }}, Lon: {{ laporan.lokasi_gps_lon floatformat:5 }} </dd> {% else %} <dd class="mt-1 text-lg text-gray- 500 italic"> Lokasi GPS tidak tersedia </dd> {% endif %} </div> <div class="sm:col-span-2"> <dt class="text-sm font-medium text- gray-500">Hasil Temuan</dt> </pre>

<pre> input" }} {{ field add_class:"form- class="error-message">{{ if field.errors }}<div fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0V6a1 1 0 00-1-1z" clip- rule="evenodd" /></svg>{{ field.errors.as_text }}</div>{{ endif %}} {{ endif %}} {{ with field=form.dokumen_pendukung %}} <div class="form-field"> <label for="{{ field.id_for_label }}" class="field-label"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M4 3a2 2 0 00-2 2v10a2 2 0 002 2h12a2 2 0 00-2 2v5a2 2 0 00-2 2H4zm12 12H4l4-8 3 6 2-4 3 6z" clip-rule="evenodd"></path></svg> {{ field.label }} </label> {{ field add_class:"form- input" }} {{ if field.errors }}<div class="error-message"><svg class="icon" fill="currentColor" viewBox="0 0 20 20"><path fill- rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0V6a1 1 0 00-1-1z" clip- rule="evenodd" /></svg>{{ field.errors.as_text }}</div>{{ endif %}} {{ endif %}} </div> <!-- End Grid --> {{ for field in form.hidden_fields %}} {{ field }} {{ endfor %}} <div id="gps-status-container" class="mt-6 p-4 bg-emerald-50 rounded-xl flex items-center transition-all duration-300 border-2 border-emerald-100"> <svg id="gps-icon" class="w-6 h-6 mr-4 text-gray-400 animate-pulse" fill="currentColor" viewBox="0 0 20 20"> <path fill-rule="evenodd" d="M5.05 4.05a7 7 0 119.9 9.9L10 18.9l-4.95-4.95a7 7 0 010-9.9zM10 11a2 2 0 100-4 2 2 0 004 2" clip- rule="evenodd" /> </svg> <div> <p id="gps-status-text" class="text-sm font-medium text-gray- 600">Mendapatkan lokasi GPS Anda...</p> <p id="gps-status-subtext" class="text-xs text-gray-500">Mohon izinkan akses lokasi di browser Anda.</p> </div> <div class="mt-8"> <button type="submit" class="submit-button"> <svg class="icon" fill="currentColor" viewBox="0 0 20 20"> <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 0016zm3.707-8.293l- 3-3a1 1 0 00-1.414 1.414L10 5.866 9H7a1 1 0 100 2h3.586l-1.293 1.293a1 1 0 101.414 1.414l3-3a1 1 0 000-1.414z" clip-rule="evenodd" /> </svg> Kirim Laporan </button> </div> </form> </div> </div> {{ endblock %}} </pre>	<pre> <div class="mt-1 text-lg text-gray-900 whitespace-pre-wrap">{{ laporan.hasil_temuan }}</div> {{ if laporan.dokumentasi_foto %}} <div class="sm:col-span-1"> <dt class="text-sm font-medium text- gray-500">Dokumentasi Foto</dt> <div class="mt-2"> </div> </div> {{ endif %}} {{ if laporan.dokumen_pendukung %}} <div class="sm:col-span-1"> <dt class="text-sm font-medium text- gray-500">Dokumen Pendukung</dt> <div class="mt-2"> Lihat Dokumen </div> </div> {{ endif %}} </div> </div> <!-- Panel Aksi Verifikasi untuk Atasan --> <div class="border-t border-gray-200 mt-8 pt-6"> <h3 class="text-lg font-medium text-gray-900 mb-4">Aksi Verifikasi</h3> <form method="POST" class="space-y-4"> {{ csrf_token %}} <div> <label for="catatan_atasan" class="block text-sm font-medium text-gray-700">Catatan (Opsional, untuk revisi atau penolakan)</label> <textarea name="catatan_atasan" id="catatan_atasan" rows="3" class="mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border- indigo-500 focus:ring-indigo-500 sm:text- sm"></textarea> </div> <div class="flex items-center space-x-4"> <button type="submit" name="aksi" value="setuju" class="w-full bg-green-500 hover:bg- green-700 text-white font-bold py-2 px-4 rounded"> Setujui Laporan </button> <button type="submit" name="aksi" value="tolak" class="w-full bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-4 rounded"> Tolak Laporan </button> </div> </form> </div> </div> {{ endblock %}} </pre>
---	---