**E-Lapor: Sistem Pelaporan dan Pengaduan Masyarakat Kota Bandar Lampung**

**👋 Selamat Datang di E-Lapor!**

E-Lapor adalah sebuah Sistem Informasi Pelaporan dan Pengaduan Masyarakat yang dirancang untuk menjadi jembatan komunikasi yang efektif antara warga Kota **Bandar Lampung** dan Pemerintah Kota (Pemkot). Sistem ini memungkinkan warga untuk melaporkan berbagai isu di lingkungan mereka dan melacak perkembangannya secara transparan.

Proyek ini dikembangkan sebagai bagian dari tugas akhir/skripsi dan mengintegrasikan teknologi *Natural Language Processing* (NLP) untuk meningkatkan efisiensi administrasi laporan.

**✨ Fitur Utama**

**Untuk Warga:**

1. **Buat Laporan Cepat:** Input laporan dengan data lokasi spesifik (jalan, kecamatan, kelurahan, titik peta) dan unggahan foto wajib.
2. **Buat Keluhan:** Sampaikan pengaduan umum secara mudah.
3. **Pelacakan Laporan *Real-time*:** Lacak status laporan/keluhan menggunakan nomor unik.
4. **Statistik Laporan:** Akses visualisasi data laporan yang masuk.
5. **Pengumuman Publik:** Melihat informasi penting dari Pemkot.

**Untuk Admin (Petugas Pemkot):**

1. **Manajemen Laporan:** Lihat, tindak lanjuti, dan kategorikan semua laporan dan keluhan yang masuk.
2. **Integrasi WhatsApp:** Tindak lanjut laporan dengan mengirimkan notifikasi ke dinas terkait via WhatsApp.
3. **Fitur NLP (Deteksi Kemiripan):** Sistem mendeteksi laporan yang berpotensi duplikat atau sangat mirip, mempermudah proses kurasi data.
4. **Ekspor Data:** Unduh laporan dalam format CSV dan PDF.
5. **Konfigurasi Sistem:** Kelola kategori laporan/keluhan dan data kontak dinas terkait.

**🛠 Teknologi yang Digunakan**

1. **Framework:** **Flask** (Python)
2. **Database:** **MySQL**
3. **Frontend:** **Bootstrap**
4. **Pemrosesan Teks:** **Natural Language Processing (NLP)** dengan library Python (misalnya, NLTK atau Scikit-learn)

## Struktur Modul dan File Project

Berikut adalah usulan struktur folder dan modul yang terorganisir untuk aplikasi Flask Anda.

### 1. Struktur Dasar Folder

| **Folder/File** | **Deskripsi** |
| --- | --- |
| app.py | File utama yang menjalankan aplikasi Flask (**Entry Point**). |
| config.py | Berisi konfigurasi aplikasi, seperti *Secret Key*, kredensial database MySQL, dan *key* API Eksternal. |
| models/ | Modul yang berisi definisi *class* untuk berinteraksi dengan database (Model). |
| routes/ | Modul yang mengatur *routing* dan logika bisnis utama (Controller/Blueprint). |
| templates/ | Berisi semua file HTML (*Jinja Templates*) untuk *user interface* (View). |
| static/ | Berisi file statis (*assets*), seperti CSS (Bootstrap), JavaScript, dan gambar. |
| nlp\_service/ | Modul khusus untuk fungsi NLP dan algoritma kesamaan teks. |
| requirements.txt | Daftar *library* Python yang dibutuhkan (Flask, Flask-MySQLdb, NLTK/Scikit-learn, dll.). |

**2. Daftar Modul/File dan Penjelasannya**

**A. Modul Utama & Konfigurasi**

| **File** | **Penjelasan Fungsi** |
| --- | --- |
| **app.py** | **Inisialisasi aplikasi Flask**, konfigurasi dari config.py, registrasi *blueprints* dari folder routes/, dan inisialisasi koneksi database. |
| **config.py** | Definisikan variabel lingkungan penting: MYSQL\_HOST, MYSQL\_USER, MYSQL\_PASSWORD, MYSQL\_DB, SECRET\_KEY, dan *key* untuk **API Peta Eksternal**. |

**B. Modul Database (models/)**

| **File** | **Penjelasan Fungsi** |
| --- | --- |
| **models/\_\_init\_\_.py** | Inisialisasi koneksi database **MySQL** (misalnya menggunakan Flask-MySQLdb atau ORM). |
| **models/laporan\_model.py** | Berisi fungsi-fungsi untuk berinteraksi dengan tabel laporan: *insert* laporan, *update* status, *fetch* laporan berdasarkan nomor, dll. |
| **models/master\_data\_model.py** | Berisi fungsi-fungsi untuk mengelola data master: kategori, data dinas terkait, data kecamatan/kelurahan. |
| **models/pengumuman\_model.py** | Berisi fungsi-fungsi untuk manajemen data pengumuman publik. |
| **models/user\_model.py** | Berisi fungsi untuk otentikasi dan manajemen data admin (login, *hashing* password). |

**C. Modul Routing & Logika Bisnis (routes/)**

*Flask menggunakan* ***Blueprint*** *untuk memecah routing agar lebih terstruktur.*

| **File** | **Penjelasan Fungsi** |
| --- | --- |
| **routes/public\_routes.py** | **Routing untuk Warga (Akses Publik)**: /, /buat-laporan, /buat-keluhan, /lacak-laporan, /statistik, /pengumuman. |
| **routes/admin\_auth\_routes.py** | **Routing Otentikasi Admin**: /admin/login, /admin/logout. |
| **routes/admin\_dashboard\_routes.py** | **Routing Area Admin**: /admin/dashboard, /admin/laporan, /admin/laporan/<id>/tindak-lanjuti. Mencakup logika **integrasi WhatsApp**. |
| **routes/admin\_config\_routes.py** | **Routing Konfigurasi Admin**: /admin/kategori, /admin/dinas-terkait, /admin/pengumuman/buat. |
| **routes/data\_export\_routes.py** | **Routing Ekspor Data**: /admin/download/csv, /admin/download/pdf. |

**D. Modul NLP dan Pelayanan Lain (nlp\_service/ & utils/)**

| **File** | **Penjelasan Fungsi** |
| --- | --- |
| **nlp\_service/duplicate\_detector.py** | **Fungsi Inti NLP**: Implementasi algoritma kesamaan teks (misalnya: *Cosine Similarity* dengan TF-IDF) untuk membandingkan deskripsi laporan yang baru masuk dengan laporan yang sudah ada. |
| **utils/whatsapp\_integrator.py** | Fungsi untuk memformat dan mengirim pesan ke nomor WhatsApp dinas terkait (bisa menggunakan **API WhatsApp Gateway** atau *library* yang sesuai). |
| **utils/map\_helper.py** | Fungsi untuk berinteraksi dengan **API Eksternal Peta** (mengambil koordinat, memusatkan peta berdasarkan kecamatan/kelurahan). |

**E. Modul Tampilan (templates/)**

| **Folder/File** | **Penjelasan Fungsi** |
| --- | --- |
| **templates/base.html** | *Template* dasar Bootstrap yang diwarisi oleh semua halaman (header, footer, navigasi). |
| **templates/public/** | Berisi semua file HTML untuk *user interface* Warga (publik): lapor.html, keluhan.html, tracking.html, statistik.html, pengumuman.html. |
| **templates/admin/** | Berisi semua file HTML untuk *user interface* Admin: login.html, dashboard.html, laporan\_list.html, tindak\_lanjut.html, master\_data.html. |

## oadmap Proyek Skripsi E-Lapor Bandar Lampung

### 1. Deskripsi Aplikasi

E-Lapor adalah Sistem Informasi Pelaporan dan Pengaduan Masyarakat yang bertujuan menjembatani komunikasi antara warga Kota Bandar Lampung dan Pemerintah Kota (Pemkot). Aplikasi ini fokus pada efisiensi pelaporan, transparansi pelacakan, dan analisis data dengan fitur khusus.

| **Pengguna** | **Fungsionalitas Utama** |
| --- | --- |
| **Warga (Publik)** | Membuat **Laporan** (wajib lokasi/foto) dan **Keluhan** (lebih ringkas), Melacak Laporan/Keluhan dengan nomor unik, Melihat Statistik, dan Melihat Pengumuman Publik. |
| **Admin (Petugas Pemkot)** | Mengelola daftar Laporan/Keluhan (termasuk *soft delete*), Menindaklanjuti dan mengirimkan notifikasi ke dinas terkait via WhatsApp, Menganalisis Statistik, Mengelola Data Master (Kategori, Dinas, Lokasi), dan membuat Pengumuman Publik. |
| **Fitur Kunci** | Integrasi Peta (Leaflet.js), Deteksi Kemiripan Laporan menggunakan **NLP**, dan Mekanisme *Soft Delete* untuk audit data. |

### 2. Tumpukan Teknologi (Tech Stack)

| **Kategori** | **Teknologi** | **Deskripsi** |
| --- | --- | --- |
| **Backend Framework** | **Flask** (Python) | Micro-framework yang ringan, cocok untuk integrasi NLP. |
| **Database** | **MySQL** | Database relasional untuk menyimpan data laporan, master, dan admin. |
| **Frontend Library** | **Bootstrap 5** | Library CSS/JS untuk tampilan yang responsif dan cepat. |
| **Database Connector** | **Flask-MySQLdb** | Ekstensi Flask untuk menghubungkan aplikasi ke MySQL. |
| **Peta** | **Leaflet.js & OpenStreetMap (OSM)** | Library JavaScript untuk menampilkan peta secara gratis, tanpa memerlukan API Key berbayar. |
| **Keamanan/Teks** | **bcrypt, NLTK/Scikit-learn** | bcrypt untuk *hashing* *password*; NLTK/Scikit-learn untuk fitur NLP (deteksi kemiripan). |

### 3. Langkah-Langkah Implementasi (Roadmap Detail)

#### FASE 1: Fondasi Proyek dan Koneksi Database (Sudah Dimulai)

| **Langkah** | **Deskripsi Aktivitas** | **File/Folder yang Dibuat** |
| --- | --- | --- |
| **1. Inisialisasi** | Siapkan lingkungan virtual dan instal dependensi (Flask, Flask-MySQLdb, bcrypt, nltk, dll.). | venv/, requirements.txt |
| **2. Konfigurasi App** | Definisikan *Secret Key* dan kredensial MySQL. | config.py |
| **3. Entry Point** | Buat instance Flask dan inisialisasi koneksi MySQL. | app.py |
| **4. Setup DB & Model** | Buat struktur folder models/ dan file inisialisasi koneksi. | models/\_\_init\_\_.py |
| **5. Data Structure** | Jalankan *query* SQL untuk membuat seluruh struktur tabel di MySQL. | e\_lapor\_db (Database) |

#### FASE 2: Sistem Admin dan Data Master

| **Langkah** | **Deskripsi Aktivitas** | **File/Folder yang Dibuat** |
| --- | --- | --- |
| **1. Model Admin** | Buat fungsi untuk mencari Admin berdasarkan email dan memverifikasi *password* (*hashing*). | models/user\_model.py |
| **2. Admin Auth Route** | Buat *Blueprint* dan logika untuk *login*, *logout*, dan pengecekan sesi Admin. | routes/admin\_auth\_routes.py |
| **3. Tampilan Auth** | Buat *template* login dan *dashboard* awal (dengan Bootstrap). | templates/admin/login.html, templates/admin/dashboard.html, templates/base.html |
| **4. Seeding Data** | Buat *script* SQL atau fungsi Python untuk menginput **Admin pertama** dan **data master awal** (kecamatan/kelurahan). | seed\_db.py (opsional) atau query SQL manual |
| **5. Model Master Data** | Buat fungsi CRUD untuk tabel m\_kategori, m\_dinas, m\_kecamatan, dan m\_kelurahan. | models/master\_data\_model.py |
| **6. Route Config Admin** | Buat *Blueprint* dan *view* untuk Admin mengelola (tambah/edit/hapus) data master. | routes/admin\_config\_routes.py, templates/admin/master\_data\_config.html |

#### FASE 3: Fitur Utama Warga (Laporan & Keluhan)

| **Langkah** | **Deskripsi Aktivitas** | **File/Folder yang Dibuat** |
| --- | --- | --- |
| **1. Model Transaksi** | Buat fungsi untuk *insert* data baru ke t\_laporan dan t\_keluhan, termasuk pembuatan nomor\_laporan unik. | models/laporan\_model.py |
| **2. Peta Helper** | Buat fungsi *helper* untuk menyediakan koordinat dan *tile URL* ke *frontend*. | utils/map\_helper.py |
| **3. Route Public** | Buat *Blueprint* dan *route* publik (/, /buat-laporan, /buat-keluhan). | routes/public\_routes.py |
| **4. Tampilan Laporan** | Buat *form* input laporan/keluhan, implementasikan **Leaflet.js** dan JS untuk memusatkan peta saat *dropdown* lokasi dipilih. | templates/public/buat\_laporan.html, templates/public/buat\_keluhan.html |
| **5. Pelacakan Laporan** | Buat fungsi di Model untuk mencari laporan/keluhan berdasarkan nomor\_laporan. | models/laporan\_model.py |
| **6. Tampilan Tracking** | Buat *form* input nomor laporan dan tampilkan status/detail laporan. | templates/public/lacak\_laporan.html |

#### FASE 4: Fitur Admin Lanjutan, NLP, dan Audit

| **Langkah** | **Deskripsi Aktivitas** | **File/Folder yang Dibuat** |
| --- | --- | --- |
| **1. Route Laporan Admin** | Buat *route* untuk menampilkan daftar t\_laporan dan t\_keluhan (Filter is\_deleted=0). | routes/admin\_dashboard\_routes.py, templates/admin/laporan\_list.html |
| **2. NLP Service** | Implementasikan algoritma kesamaan teks (misal: Cosine Similarity/TF-IDF) untuk membandingkan deskripsi laporan baru dengan yang sudah ada. | nlp\_service/duplicate\_detector.py |
| **3. Integrasi NLP** | Integrasikan duplicate\_detector ke dalam fungsi *insert* di models/laporan\_model.py untuk mengisi kolom is\_mirip. | models/laporan\_model.py (Update) |
| **4. WA Integrator** | Buat fungsi untuk mengirimkan notifikasi WA ke m\_dinas (membutuhkan API/Gateway WA eksternal). | utils/whatsapp\_integrator.py |
| **5. Tindak Lanjut** | Buat *route* dan Model untuk Admin mengubah status laporan, mencatat di t\_tindak\_lanjut, dan memanggil fungsi WA. Implementasikan **Soft Delete** di sini (Update kolom is\_deleted). | routes/admin\_dashboard\_routes.py (Update), models/tindak\_lanjut\_model.py |
| **6. Audit Log** | Pastikan setiap perubahan status dan *soft delete* dicatat di tabel t\_audit\_log. | models/audit\_model.py |
| **7. Statistik & Export** | Kembangkan *query* untuk statistik laporan dan implementasikan fitur ekspor CSV/PDF. | routes/data\_export\_routes.py, templates/public/statistik.html |

#### FASE 5: Pemeliharaan dan Finalisasi

| **Langkah** | **Deskripsi Aktivitas** | **Keterangan** |
| --- | --- | --- |
| **1. Uji Coba (Testing)** | Lakukan pengujian fungsionalitas menyeluruh (input, tracking, login, NLP, WA). | Perbaiki *bug* dan *error*. |
| **2. Dokumentasi** | Lengkapi *file* README.md dan buat dokumentasi teknis (untuk lampiran skripsi). | Dokumentasi Kode dan Sistem. |
| **3. Deploy Lokal** | Pastikan aplikasi berjalan stabil di lingkungan lokal Anda. | Siap untuk presentasi. |