

**LAPORAN APLIKASI PENGADUAN  
MASYARAKAT**

**KELOMPOK 1:**

**IHSAN NUR ALAM | 231712115**

**RIO H. P. SIREGAR | 231712121**

**WIPANDO BRANDO NAINGGOLAN | 231712114**

**FIDELIS PANDAPOTAN B. SIMANJUNTAK | 231712113**

**KOM B'23**



**PROGRAM STUDI D-3 TEKNIK INFORMATIKA  
FAKULTAS VOKASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2025**

## **KATA PENGANTAR**

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, hidayah, dan karunia-Nya yang telah dilimpahkan. Berkat anugerah tersebut, Laporan yang berjudul "Laporan Aplikasi Pengaduan Masyarakat" ini dapat diselesaikan dengan baik dan tepat waktu. Makalah ini disusun sebagai salah satu bentuk pertanggungjawaban akademis dan bagian dari pemenuhan tugas pada mata kuliah Pemrograman Visual, yang bertujuan untuk mendokumentasikan hasil perancangan aplikasi. Penulis menyadari bahwa proses penyusunan laporan ini masih jauh dari sempurna, baik dari segi kedalaman materi maupun sistematika penulisannya.

Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada Bapak Dr. Niskarto Zendrato, S.Kom., M.Kom. selaku Dosen Pengampu mata kuliah Pemrograman Visual atas bimbingan dan arahan yang telah diberikan. Penghargaan juga disampaikan kepada Ibu Nadira Natalie Kuechler selaku Asisten Laboratorium atas bantuan teknis dan praktis selama proses pengembangan aplikasi, serta kepada seluruh rekan-rekan yang telah memberikan dukungan. Penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan karya tulis ini di masa yang akan datang. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi para pembaca.

Medan, 1 Desember 2025

Kelompok 1

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR.....</b>	<b>iii</b>
<b>DAFTAR TABEL.....</b>	<b>iv</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	2
<b>BAB 2 TINJAUAN PUSTAKA .....</b>	<b>4</b>
2.1 Komponen Penelitian .....	4
2.1.1 Rekayasa Perangkat Lunak.....	4
2.1.2 Graphical User Interface (GUI) .....	4
2.1.3 Sistem Manajemen Basis Data .....	5
2.2 Metode Pengerjaan Dan Teknologi Implementasi.....	5
2.2.1 Visual Studio.....	6
2.2.2 #C.....	6
2.2.3 XAMPP.....	7
2.2.4 MySQL Connector.....	8
<b>BAB 3 METODOLOGI PENELITIAN.....</b>	<b>9</b>
3.1 Diagram Penelitian .....	12
3.1.1 Flowchart .....	9
3.2 Struktur Basis Data Relasional .....	12
3.2.1 Keterbatasan Perancangan .....	14
3.3 Jadwal Penelitian .....	15
<b>BAB 4 HASIL .....</b>	<b>17</b>
4.1 Pengertian .....	17
4.2 Penggunaan.....	17
4.3 Hasil Implementasi Aplikasi .....	18
4.3.1 Struktur Aplikasi .....	19
4.3.2 Dashboard User.....	20
4.3.3 Dashboard Admin .....	22
<b>BAB 5 PENUTUP.....</b>	<b>24</b>
5.1 Kesimpulan.....	24
5.2 Saran .....	25
<b>DAFTAR PUSTAKA.....</b>	<b>26</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Logo Visual Studio.....	7
<b>Gambar 2.2</b>	Logo XAMPP .....	7
<b>Gambar 2.3</b>	Logo MySQL .....	7
<b>Gambar 3.1</b>	Flowchart .....	8
<b>Gambar 3.2</b>	ERD Database .....	10
<b>Gambar 4.1</b>	Form Login .....	14
<b>Gambar 4.2</b>	Form Register.....	15
<b>Gambar 4.3</b>	Tampilan Buat Pengaduan.....	15
<b>Gambar 4.4</b>	Tampilan Riwayat Pengaduan.....	16
<b>Gambar 4.5</b>	Tampilan Pengaduan Masuk .....	17
<b>Gambar 4.6</b>	Tampilan Riwayat Selesai .....	17

## DAFTAR TABEL

<b>Tabel 3.1</b> Jadwal Penelitian .....	12
<b>Tabel 4.1</b> Penggunaan Aplikasi .....	14

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Proyek pengembangan aplikasi Pengaduan Masyarakat ini dilaksanakan sebagai bagian dari pemenuhan tugas akhir mata kuliah Laboratorium Pemrograman Visual. Dalam era digital, kebutuhan akan transparansi dan akuntabilitas pelayanan publik menjadi semakin mendesak. Masalah mendasar yang sering dihadapi oleh instansi pemerintah adalah mekanisme pengaduan masyarakat yang masih bersifat konvensional, seperti melalui surat atau tatap muka, yang seringkali lambat, kurang terdokumentasi, dan menghambat proses tindak lanjut yang efisien (Wahyuni & Darmawan, 2023).

Faktanya, sistem informasi pengaduan berbasis teknologi sangat krusial untuk menjembatani komunikasi antara masyarakat dan pihak berwenang. Sistem yang efisien harus mampu mencatat laporan secara terstruktur, memberikan feedback berupa status pengaduan (belum diproses, diproses, selesai), dan menyediakan saluran yang mudah diakses, termasuk pengiriman bukti pendukung seperti foto kejadian (Suryani et al., 2024).

Guna mengatasi kesenjangan ini dan mentransformasi proses pengaduan menjadi digital, pengembangan aplikasi berbasis teknologi menjadi solusi strategis. Dalam konteks pengembangan, lingkungan Integrated Development Environment (IDE) seperti Microsoft Visual Studio menawarkan platform yang komprehensif untuk merancang sistem, karena menyediakan fitur debugging yang kuat dan memfasilitasi integrasi yang mudah dengan basis data. Oleh karena itu, aplikasi Pengaduan Masyarakat dikembangkan menggunakan Visual Studio untuk menyediakan sistem digital yang valid dan andal bagi masyarakat untuk menyampaikan keluhan dan bagi administrator untuk menindaklanjuti secara terorganisir.

### **1.2 Rumusan Masalah**

Permasalahan inti dalam pengembangan aplikasi ini adalah:

1. Bagaimana merancang dan mengimplementasikan sistem otentikasi peran (User dan Admin) yang terstruktur menggunakan Visual Studio?
2. Bagaimana mengembangkan modul pelaporan User yang komprehensif (termasuk foto kejadian) dan menampilkan riwayat status pengaduan?
3. Bagaimana merancang modul workflow Admin untuk memperbarui status pengaduan (belum diproses hingga selesai) dan menyediakan fitur arsip data yang telah tuntas?

### 1.3 Batasan Masalah

Pengembangan aplikasi ini memiliki batasan fungsional sebagai berikut:

1. Otentikasi: Hanya mencakup dua peran dasar (User dan Admin), tanpa otorisasi atau peran bertingkat yang lebih kompleks.
2. Manajemen Pengaduan: Fokus pada pencatatan dan pembaruan status manual, tidak mencakup notifikasi *real-time* (Email/SMS) atau mekanisme penugasan otomatis.
3. Pengarsipan: Hanya melihat riwayat status Selesai; tidak mengimplementasikan fitur ekspor data (*PDF/Excel*) atau analisis data pengaduan.

### 1.4 Tujuan Penelitian

Tujuan dari proyek pengembangan aplikasi ini adalah:

1. Implementasi Otentikasi: Mampu merancang dan menguji sistem *login* dan registrasi yang memisahkan hak akses User dan Admin.
2. Pengembangan Modul User: Berhasil membangun fungsionalitas bagi User untuk melaporkan pengaduan secara detail dan melihat *history* status laporan mereka.
3. Perancangan *Workflow* Admin: Menyediakan dan menguji sistem bagi Admin untuk memproses, memperbarui status pengaduan, dan mengakses daftar laporan yang telah tuntas.

### 1.5 Manfaat Penelitian

Hasil dari pengembangan aplikasi ini diharapkan memberikan manfaat sebagai berikut:

1. Akademis (Penerapan Ilmu): Proyek menjadi bukti nyata penerapan konsep Pemrograman Visual dan pemahaman terhadap perancangan Sistem Informasi.
2. Praktis (Publik dan Admin): Meningkatkan efisiensi dan transparansi pelayanan publik dengan menyediakan saluran pelaporan digital yang terstruktur.
3. Mahasiswa (Kompetensi Teknis): Memberikan pengalaman praktis dalam siklus pengembangan perangkat lunak (*SDLC*) dan meningkatkan *portfolio* keterampilan teknis.



## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1     Komponen Penelitian**

Komponen penelitian ini mencakup unsur-unsur teoritis utama yang menjadi dasar perancangan dan implementasi aplikasi Pengaduan Masyarakat. Komponen ini berfokus pada konsep rekayasa perangkat lunak, antarmuka pengguna, dan teknologi basis data yang mendasari fungsionalitas sistem.

##### **2.1.1   Rekayasa Perangkat Lunak**

Rekayasa Perangkat Lunak adalah disiplin ilmu yang mempelajari semua aspek produksi perangkat lunak, mulai dari spesifikasi sistem, pengembangan, hingga pemeliharaan. Tujuannya adalah menghasilkan perangkat lunak yang berkualitas, tepat waktu, dan sesuai anggaran (IEEE, 2024).

Dalam konteks proyek ini, prinsip Rekayasa Perangkat Lunak diterapkan melalui penggunaan metodologi pengembangan terstruktur. Ini memastikan bahwa tahapan pengembangan aplikasi—mulai dari analisis kebutuhan (user dan admin), perancangan sistem, hingga pengujian fungsional—dilaksanakan secara sistematis untuk meminimalisir bug dan memenuhi semua kebutuhan fungsionalitas yang telah ditetapkan di awal.

##### **2.1.2   Graphical User Interface (GUI)**

Graphical User Interface (GUI) adalah jenis antarmuka pengguna yang memungkinkan pengguna berinteraksi dengan perangkat elektronik melalui indikator visual dan ikon, alih-alih melalui perintah berbasis teks (command line) (Techopedia, 2023).

Pengembangan aplikasi berbasis Visual Studio secara spesifik mengandalkan perancangan GUI. Ini krusial karena aplikasi Pengaduan Masyarakat harus memiliki antarmuka yang intuitif dan mudah digunakan (user-friendly), baik untuk User saat mengisi formulir laporan maupun untuk Admin saat memproses dan mengubah status

laporan. Perancangan GUI yang baik akan mengurangi error pengguna dan meningkatkan efisiensi operasional.

### **2.1.3 Sistem Manajemen Basis Data**

Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak terintegrasi yang dirancang untuk mendefinisikan, memanipulasi, mengambil, dan mengelola data dalam suatu basis data. DBMS berfungsi sebagai antarmuka antara aplikasi (front-end) dan data yang disimpan secara fisik (back-end), memastikan data dikelola secara efisien, aman, dan terstruktur (Oracle, 2024).

Peran DBMS dalam Aplikasi Pengaduan Masyarakat sangat fundamental, mencakup aspek-aspek krusial berikut:

1. **Integritas Data:** DBMS menjamin bahwa data yang dimasukkan (seperti detail laporan, nomor kontak, dan status) adalah akurat dan konsisten. Hal ini penting untuk memastikan bahwa pelaporan yang masuk tidak duplikat dan sesuai dengan format yang ditetapkan.
2. **Keamanan Data:** Dengan adanya mekanisme kontrol akses (seperti RBAC), DBMS melindungi informasi sensitif User dan Admin dari akses yang tidak sah.
3. **Akses Real-Time:** DBMS memungkinkan pengambilan data status pengaduan secara cepat, sehingga User dapat melihat riwayat laporan dan Admin dapat memproses laporan yang masuk tanpa adanya delay signifikan.

## **2.2 Metode Pengerjaan Dan Teknologi Implementasi**

Metode pengerjaan adalah tahapan sistematis dalam pengembangan aplikasi Pengaduan Masyarakat berbasis desktop, dimulai dari analisis kebutuhan, perancangan antarmuka, implementasi program, hingga uji coba sistem. Pengembangan aplikasi ini merujuk pada prinsip rekayasa perangkat lunak yang mencakup tahapan analisis, perancangan, pengkodean, dan pengujian (Pressman & Maxim, 2020). Aplikasi ini menggunakan kombinasi teknologi perangkat lunak yang dipilih secara spesifik untuk menciptakan aplikasi desktop dengan Graphical User Interface (GUI) yang terintegrasi penuh dengan sistem manajemen basis data (Database Management System - DBMS). Kombinasi teknologi ini menjadi kerangka kerja implementasi kode program, manajemen database, dan konektivitas data.

### 2.2.1 Visual Studio

Visual Studio merupakan Integrated Development Environment (IDE) andalan yang dikembangkan oleh Microsoft, berfungsi sebagai lingkungan kerja utama untuk merancang, mengkodekan, men-debug, dan menguji aplikasi desktop yang berjalan pada platform Windows. IDE ini menyediakan dukungan komprehensif untuk bahasa pemrograman modern, terutama C#, dan menawarkan fitur unggulan berupa desainer antarmuka grafis yang kuat (Albahari & Albahari, 2022).

Visual Studio diklasifikasikan sebagai Rapid Application Development tools (RAD tools) karena kemampuannya dalam mempercepat proses pengembangan. Fasilitas form designer yang mendukung mekanisme drag-and-drop pada Visual Studio memungkinkan pengembang untuk membangun antarmuka aplikasi Pengaduan Masyarakat secara visual. Ini sangat penting untuk menciptakan antarmuka yang intuitif bagi kedua peran (User dan Admin), sehingga meminimalkan waktu yang dibutuhkan untuk pembuatan form pelaporan, dashboard riwayat, dan modul pengelolaan workflow. Lingkungan terpadu ini juga menjamin semua source code dan konfigurasi proyek dikelola secara terpusat dan efisien (Jubile Enterprise, 2015).



Gambar 2.1 Logo Visual Studio

### 2.2.2 #C

C# (C Sharp) adalah bahasa pemrograman berorientasi objek yang merupakan bagian integral dari kerangka kerja .NET milik Microsoft. Bahasa ini dipilih karena kombinasi antara sintaks yang elegan dan kekuatan *framework* .NET, yang memberikan keamanan memori dan stabilitas tinggi—sifat-sifat yang krusial untuk aplikasi yang menangani data publik. C# sangat terintegrasi dengan Visual Studio, memungkinkan *debugging* dan kompilasi kode yang mulus (Albahari & Albahari, 2022).

Dalam proyek Aplikasi Pengaduan Masyarakat, C# digunakan untuk mengimplementasikan logika bisnis yang kompleks. Ini mencakup implementasi algoritma untuk:

1. Sistem Otentikasi dan Otorisasi: Mengelola proses *login* dan registrasi, serta memvalidasi peran pengguna (User atau Admin) sebelum memberikan akses ke fungsionalitas yang relevan.
2. Validasi dan Pemrosesan Input: Memastikan semua data pelaporan (teks, nomor, dan unggahan foto) memenuhi aturan validasi sebelum dikirim ke basis data.
3. Kontrol *Workflow*: Mengatur fungsi untuk mengubah data status laporan yang hanya dapat dieksekusi oleh Admin, memastikan integritas *workflow* pengaduan.

### 2.2.3 XAMPP

XAMPP adalah paket perangkat lunak bebas (open source) yang berfungsi sebagai server lokal (localhost) yang cross-platform (X), dan merupakan singkatan dari X, Apache, MySQL, PHP, dan Perl (Dwi Priyanti, 2013).

Dalam konteks pengembangan aplikasi desktop ini, XAMPP digunakan secara spesifik untuk menjalankan dan menyediakan MySQL sebagai Sistem Manajemen Basis Data (DBMS). MySQL dipilih karena kapabilitasnya dalam mengelola data relasional secara efisien dan cepat, yang penting untuk menangani penyimpanan data pengaduan yang terstruktur. Meskipun aplikasi utama dikembangkan di Visual Studio menggunakan C#, instalasi XAMPP menyediakan lingkungan yang mudah diatur untuk mengimplementasikan dan menguji kinerja MySQL, memastikan fungsionalitas database berjalan dengan baik sebelum dihubungkan ke aplikasi utama (Ramdani, 2023). MySQL bertanggung jawab menyimpan seluruh data aplikasi, termasuk registrasi User, detail pengaduan, dan history status workflow.



Gambar 2.2 Logo Xampp

#### 2.2.4 MySQL Connector

MySQL Connector adalah driver atau Application Programming Interface (API) yang berfungsi sebagai komponen esensial untuk membangun jembatan komunikasi antara aplikasi front-end (Visual Studio C#) dengan basis data back-end (MySQL). MySQL Connector memungkinkan C# untuk menginterpretasikan dan mengirimkan perintah Structured Query Language (SQL) ke database MySQL yang terpisah (Fauzi & Amin, 2012).

Tanpa konektor ini, aplikasi desktop tidak memiliki kemampuan untuk mengakses, memodifikasi, atau mengambil data dari MySQL. Secara fungsional, MySQL Connector mendukung semua operasi Database yang diperlukan oleh aplikasi, yaitu:

1. Penyimpanan (Create): Menyimpan data laporan baru dari User.
2. Pengambilan (Read): Menampilkan riwayat laporan kepada User atau menampilkan daftar antrian laporan kepada Admin.
3. Pembaruan (Update): Mengubah status laporan dari belum diproses menjadi diproses atau selesai oleh Admin.

Keberadaan konektor ini menjamin konektivitas real-time yang diperlukan agar workflow Admin dapat berjalan lancar dan informasi status laporan selalu up-to-date.



Gambar 2.3 Logo MySQL

## **BAB 3**

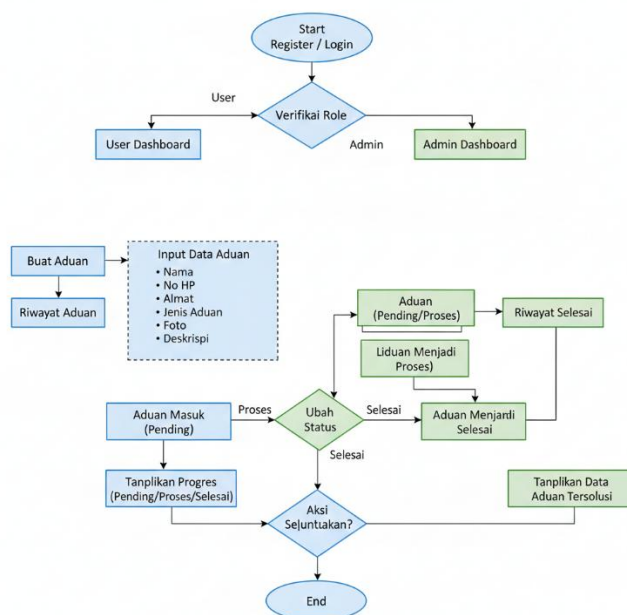
### **METODOLOGI PENELITIAN**

#### **3.1 Diagram Penelitian**

Diagram penelitian digunakan untuk menggambarkan alur proses pengembangan sistem mulai dari identifikasi masalah sampai tahap implementasi dan pelaporan. Model ini membantu peneliti memahami keterkaitan antar tahapan serta memastikan bahwa proses pengembangan mengikuti standar rekayasa perangkat lunak yang sistematis (Sommerville, 2016). Metode yang digunakan dalam pengembangan aplikasi ini adalah model Waterfall (Air Terjun) atau model sekuensial linier. Model ini dipilih karena tahapan proyek dilakukan secara berurutan dan sistematis, yang sangat cocok untuk proyek pengembangan yang memiliki spesifikasi kebutuhan fungsional yang jelas dan terdefinisi di awal, seperti pemisahan role dan workflow status pengaduan (Kendall & Kendall, 2014)

##### **3.1.1 Flowchart**

Diagram alir penelitian (Flowchart) memvisualisasikan langkah-langkah operasional dan logis yang akan ditempuh dalam aplikasi, dimulai dari otentikasi pengguna hingga pengelolaan workflow pengaduan oleh Admin. Diagram ini berfungsi sebagai representasi grafis dari logika pemrograman sistem, yang menggambarkan secara rinci alur kerja, percabangan keputusan, dan pemrosesan data, memastikan setiap fungsionalitas (seperti pengubahan status dan pelacakan riwayat) dapat diimplementasikan sesuai kebutuhan fungsionalitas sistem.



Gambar 3.1 Flowchart

Flowchart ini menggambarkan dua alur utama berdasarkan *role* pengguna: User dan Admin.

### 1. Alur Awal (Login & Verifikasi Role)

- A. Start: Titik awal dari seluruh proses sistem.
- B. Register / Login: Pengguna memulai dengan mendaftar akun baru (jika belum punya) atau masuk/login menggunakan akun yang sudah ada.
- C. Verifikasi Role: Setelah login, sistem akan memeriksa jenis akun pengguna. Apakah dia "User" atau "Admin"?
  - a) Jika User, alur akan mengarah ke User Dashboard.
  - b) Jika Admin, alur akan mengarah ke Admin Dashboard.

### 2. Alur untuk User (Dashboard User)

- A. User Dashboard: Ini adalah halaman utama yang dilihat oleh pengguna biasa. Dari sini, user memiliki dua opsi utama:

Buat Aduan: Pengguna memilih untuk membuat aduan baru.

- a) Input Data Aduan: Pada langkah ini, pengguna mengisi semua informasi yang diperlukan untuk aduan, termasuk:
  1. Nama
  2. No HP

3. Alamat
  4. Jenis Aduan
  5. Foto (sebagai lampiran)
  6. Deskripsi (detail kejadian)
- b) Aduan Masuk (Pending): Setelah user submit, aduan tersebut akan tersimpan dalam sistem dengan status awal "Pending" (belum ditindaklanjuti).

Riwayat Aduan: Pengguna memilih untuk melihat daftar aduan yang pernah ia buat sebelumnya.

- c) Tampilkan Progres (Pending/Proses/Selesai): Pada bagian ini, user dapat melihat status terkini dari setiap aduan yang diajukan. Status bisa berupa "Pending" (belum ditangani), "Proses" (sedang ditangani), atau "Selesai" (telah ditangani). Ini juga terhubung dari node "Aduan Masuk (Pending)" dan "Ubah Status" yang akan dijelaskan di alur Admin.

### 3. Alur untuk Admin (Dashboard Admin)

A. Admin Dashboard: Ini adalah halaman utama yang dilihat oleh administrator. Admin juga memiliki beberapa fungsi utama:

- a) Aduan (Pending/Proses): Admin dapat melihat daftar semua aduan yang masuk, baik yang masih "Pending" maupun yang sedang dalam status "Proses".
  - Laporan Menjadi Proses: Jika Admin mengubah status aduan dari "Pending" menjadi "Proses", aduan akan masuk ke dalam kategori ini, menandakan sedang dalam penanganan.
- b) Ubah Status: Ini adalah titik keputusan krusial bagi Admin. Dari aduan yang masuk, Admin dapat:
  - Mengubah status aduan menjadi "Proses": Ini berarti aduan sedang ditangani oleh Admin atau tim terkait. Alur akan kembali ke "Aduan (Pending/Proses)" karena aduan tersebut masih aktif ditangani.
  - Mengubah status aduan menjadi "Selesai": Ini berarti aduan telah berhasil ditangani dan diselesaikan.



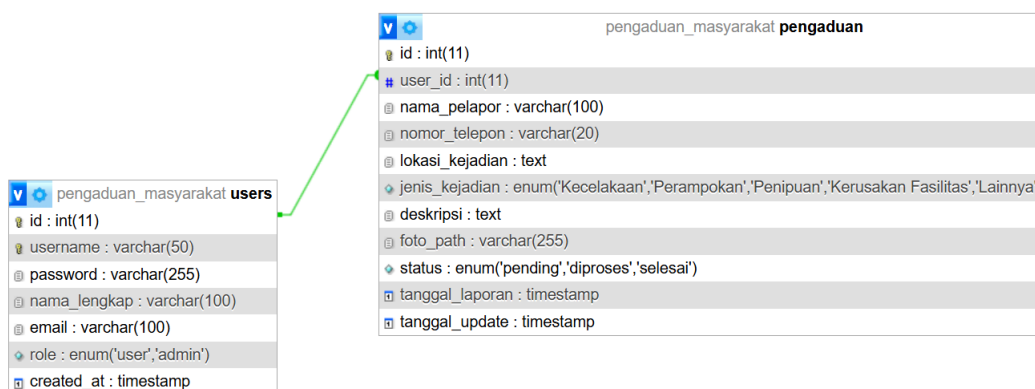
- Aduan Menjadi Selesai: Aduan yang telah diselesaikan akan masuk ke tahap ini.
- c) Riwayat Selesai: Admin dapat melihat daftar semua aduan yang statusnya sudah "Selesai".
  - Tampilkan Data Aduan Tersolusi: Di sini, Admin dapat melihat detail aduan yang telah berhasil ditangani dan solusinya.

#### 4. Alur Penutup

- A. Aksi Selanjutnya?: Setelah melakukan suatu tindakan (baik itu User melihat progres atau Admin mengubah status), sistem mungkin menanyakan apakah ada aksi lain yang ingin dilakukan.
  - Jika Ya (ada aksi lain), alur akan kembali ke dashboard yang sesuai (User atau Admin) atau ke opsi awal. (Pada flowchart ini, panah mengarah ke End jika tidak ada aksi selanjutnya).
- B. End: Titik akhir dari proses sistem.

### 3.2 Struktur Basis Data Relasional

Struktur basis data relasional ini merupakan hasil implementasi dari perancangan Entity-Relationship Diagram (ERD) yang bertujuan untuk menyimpan dan mengelola data pengaduan masyarakat secara efisien. Basis data ini terdiri dari dua entitas utama: users dan pengaduan, dengan hubungan *one-to-many* (satu pengguna dapat membuat banyak pengaduan).



Gambar 3.2 ERD Database

### 1. Tabel `users` (Pengguna)

Tabel ini berfungsi sebagai penyimpanan data seluruh pengguna yang dapat mengakses sistem. Kolom kunci di sini adalah `id` sebagai *Primary Key* (Kunci Utama) dan `role` yang bertipe data `enum('user', 'admin')`. Kolom `role` ini sangat penting karena ia adalah penentu hak akses (RBAC) yang memisahkan fungsionalitas antara masyarakat yang melapor (*user*) dan petugas yang menangani laporan (*admin*). Kolom lain seperti `username`, `password`, dan `email` mendukung proses otentikasi dan registrasi.

### 2. Tabel `pengaduan` (Laporan)

Tabel ini adalah inti dari sistem karena menyimpan semua detail laporan yang masuk. Selain `id` sebagai *Primary Key* laporan, tabel ini memiliki beberapa kolom krusial:

1. `user_id`: Ini adalah Kunci Asing (*Foreign Key*) yang menghubungkan setiap laporan kembali ke pengguna (`id` di tabel `users`) yang membuatnya. Ini memungkinkan User untuk melihat riwayat laporannya sendiri dan Admin untuk mengetahui siapa pelapornya.
2. `jenis_kejadian`: Menggunakan tipe `enum` untuk mengkategorikan laporan secara terstruktur (Kecelakaan, Perampokan, dll.).
3. `foto_path`: Menyimpan lokasi penyimpanan file foto pendukung, yang krusial sebagai bukti pelaporan.
4. `status`: Kolom ini menggunakan `enum('pending', 'diproses', 'selesai')`. Kolom inilah yang diubah oleh Admin dalam *workflow* sistem, menunjukkan progres penanganan laporan dari awal hingga terselesaikan.

### Relasi Kunci

Kedua tabel dihubungkan melalui relasi One-to-Many (Satu-ke-Banyak). Artinya, satu akun pengguna di tabel `users` dapat membuat banyak laporan yang tersimpan di tabel `pengaduan` melalui kolom `user_id`. Struktur ini memastikan integritas data terjamin dan pelacakan riwayat berjalan lancar.

### 3.2.1 Keterbatasan Perancangan

Struktur basis data yang terdiri dari tabel `users` dan `pengaduan` ini memiliki keterbatasan yang sesuai dengan batasan masalah proyek dan fungsionalitas lanjutan yang tidak diakomodir:

1. Tanpa Tabel Riwayat Proses Khusus (Audit Trail):

- a) Tabel `pengaduan` hanya mencatat status terakhir (`status` enum) dan waktu pembaruan terakhir (`tanggal_update`).
- b) Keterbatasan: Sistem tidak memiliki tabel terpisah (misalnya `riwayat_status`) yang mencatat secara detail siapa Admin yang memproses laporan, kapan status diubah dari pending ke diproses, dan catatan/tanggapan spesifik dari Admin di setiap tahapan. Hal ini menyulitkan pelacakan audit secara mendalam (siapa bertanggung jawab atas perubahan status tertentu).

2. Foto Terpisah dari *Database* (Hanya *Path*):

- a) Tabel `pengaduan` hanya menyimpan lokasi penyimpanan (`foto_path`) dari file foto, bukan file foto itu sendiri.
- b) Keterbatasan: Jika *path* file berubah atau file foto dihapus secara manual dari *server* (*folder* fisik), data foto pada laporan akan hilang dan tidak dapat diakses lagi. Integritas data foto bergantung sepenuhnya pada stabilitas sistem *file server*.

3. Tidak Ada Tabel Master Jenis Kejadian:

- a) Kolom `jenis_kejadian` pada tabel `pengaduan` menggunakan tipe data `enum` yang menyimpan kategori kejadian secara permanen dalam definisi tabel.
- b) Keterbatasan: Penambahan, pengurangan, atau perubahan kategori jenis kejadian (misalnya, menambahkan kategori "Bencana Alam") memerlukan perubahan langsung pada struktur tabel `pengaduan` di *database* (`ALTER TABLE`). Hal ini tidak fleksibel dan tidak dapat dikelola secara dinamis oleh Admin melalui antarmuka aplikasi.



proses pengolahan data, serta penerapan logika sesuai fungsi yang dibutuhkan pengguna.																			
tahap ini dilakukan evaluasi terhadap seluruh komponen sistem, perbaikan kesalahan yang ditemukan, optimalisasi fungsi agar kinerja aplikasi maksimal, dan penyusunan dokumen sebagai hasil akhir penelitian.																			

Tabel 3.1 Jadwal Penelitian

## **BAB 4**

### **HASIL**

#### **4.1 Pengertian**

Aplikasi “Pengaduan Masyarakat” adalah sistem informasi berbasis desktop yang dibuat untuk memudahkan warga dalam menyampaikan laporan atau keluhan terkait berbagai permasalahan di lingkungan mereka. Sistem ini menggantikan proses pelaporan manual menjadi pencatatan digital yang lebih rapi dan mudah dipantau. Melalui aplikasi ini, pengguna dapat melihat perkembangan status aduan mulai dari diterima, diproses, hingga diselesaikan. Dengan adanya sistem ini, penanganan pengaduan dapat berlangsung lebih cepat, teratur, dan transparan.

#### **4.2 Penggunaan**

Berikut adalah panduan singkat langkah-langkah operasional untuk modul-modul kunci dalam aplikasi:

Tabel 4.1 Penggunaan Aplikasi

Modul	Tujuan	Langkah Penggunaan
Registrasi & Login	Mengelola proses pendaftaran pengguna baru dan otentikasi akses ke sistem berdasarkan peran (User atau Admin).	1. Mulai Aplikasi: Pilih tombol Registrasi untuk pendaftaran akun baru, atau Login jika sudah memiliki akun. 2. Login: Masukkan Username dan Password. 3. Sistem memverifikasi dan mengarahkan ke Dashboard sesuai peran.
Buat Pengaduan (User)	Memberikan sarana bagi masyarakat untuk mencatat laporan kejadian secara terstruktur dan terintegrasi.	1. Akses modul Buat Pengaduan dari Dashboard User. 2. Isi data wajib (Nama, Kontak, Lokasi, Deskripsi, Jenis Kejadian). 3. Unggah Foto kejadian sebagai bukti. 4. Klik Simpan; laporan otomatis tersimpan di database dengan Status: Pending / Belum Diproses.

Riwayat Pengaduan (User)	Memungkinkan User memonitor progres penanganan laporan yang telah mereka buat.	1. Akses modul Riwayat Pengaduan dari Dashboard User. 2. Sistem menampilkan daftar semua laporan yang pernah dibuat oleh akun tersebut. 3. User dapat melihat Status terbaru dari setiap laporan: Pending, Diproses, atau Selesai.
Aduan Masuk (Admin)	Mengelola workflow laporan yang masuk (Pending dan Diproses) dan memproses tindak lanjut.	1. Akses modul Aduan Masuk dari Dashboard Admin. 2. Pilih laporan berstatus Pending untuk ditindaklanjuti. 3. Klik tombol Ubah Status ke Diproses saat verifikasi dimulai. 4. Setelah penanganan selesai, klik tombol Ubah Status ke Selesai.
Riwayat Selesai (Admin)	Menyediakan arsip data laporan yang telah tuntas ditangani sebagai referensi dan dokumentasi.	1. Akses modul Riwayat Selesai dari Dashboard Admin. 2. Sistem menampilkan semua laporan yang statusnya telah diubah menjadi Selesai (tidak termasuk laporan yang masih Pending atau Diproses). 3. Admin dapat mencari dan melihat detail laporan yang telah terselesaikan.

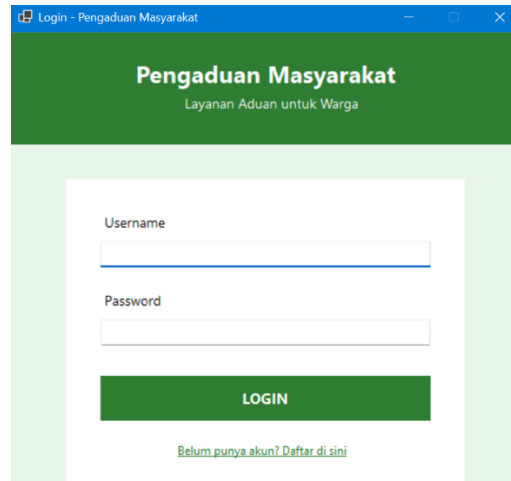
#### 4.3 Hasil Implementasi Aplikasi

Implementasi aplikasi adalah proses menterjemahkan desain sistem dan basis data ke dalam kode program C# yang berjalan di lingkungan Visual Studio.

#### 4.3.1 Struktur Aplikasi

Aplikasi Pengaduan Masyarakat dirancang dengan antarmuka yang intuitif dan fungsional untuk memudahkan User (Masyarakat) dalam membuat laporan dan Admin dalam mengelola workflow status.

1. Form Login : Fungsi utama halaman ini adalah untuk otentikasi pengguna sebelum dapat mengakses fitur sistem.
  - a. Fungsi Utama: Pengguna memasukkan Username dan Password.
  - b. Aksi: Mengklik tombol LOGIN untuk diverifikasi perannya (User atau Admin) dan diarahkan ke *dashboard*.
  - c. Aksi Alternatif: Terdapat tautan "Belum punya akun? Daftar di sini" yang mengarahkan pengguna baru ke halaman Registrasi (*Daftar*) untuk membuat akun.



Gambar 4.1 Form Login

2. Form Register: Fungsi utama halaman ini adalah memungkinkan masyarakat (User) membuat akun baru untuk mengakses sistem.
  - a. Fungsi Utama: Pengguna mengisi data yang diperlukan untuk pendaftaran: Username, Password, Nama Lengkap, dan Email.
  - b. Aksi: Mengklik tombol DAFTAR untuk menyimpan data ke tabel `users` (dengan `role` otomatis sebagai 'user') dan menyelesaikan pembuatan akun.
  - c. Aksi Alternatif: Terdapat tombol "Kembali ke Login" yang mengarahkan pengguna kembali ke halaman *login* setelah pendaftaran berhasil atau jika mereka sudah memiliki akun.



Gambar 4.2 Form Register

#### 4.3.2 Dashboard User

1. Buat Pengaduan: Fungsi utama halaman ini adalah memungkinkan pengguna (User) yang telah *login* untuk membuat dan mengirimkan laporan baru ke sistem.

##### Fungsionalitas Modul

1. Sidebar: Menyediakan navigasi utama:
  - a. Buat Pengaduan (sedang aktif).
  - b. Riwayat Pengaduan (untuk melihat status laporan yang sudah dikirim).
  - c. Logout (untuk keluar dari akun).
2. Form Input: User diminta mengisi data wajib, termasuk:
  - a. Nama Pelapor (terisi otomatis atau dapat diubah).
  - b. Nomor Telepon.
  - c. Lokasi Kejadian.
  - d. Jenis Kejadian (menggunakan *dropdown*).
  - e. Deskripsi Kejadian.
3. Bukti Pendukung: Terdapat opsi Foto Kejadian (Opsional), yang memungkinkan User mengunggah bukti visual sebelum menekan tombol KIRIM PENGADUAN.

Setelah dikirim, laporan ini akan masuk ke *database* dengan status Pending/Belum Diproses, yang kemudian dapat dilihat oleh Admin.

Gambar 4.3 Tampilan Buat Pengaduan

2. Riwayat Pengaduan : Halaman ini berfungsi sebagai pusat pelacakan bagi User untuk melihat semua laporan yang pernah mereka kirim. Daftar laporan ditampilkan dalam tabel (Data Grid), mencakup ID, Jenis kejadian, Lokasi, Tanggal, dan yang terpenting, kolom Status (Belum Ditindak, Diproses, atau Selesai). Ini memberikan transparansi real-time mengenai progres penanganan laporan oleh Admin.

ID	Jenis	Lokasi	Tanggal	Status
3	Penipuan	Padang Lintungan	30-11-2025 03:36	Belum Ditindak
2	Perampokan	Susuk Jaya	30-11-2025 14:47	Selesai
1	Kecelakaan	Tembung city	30-11-2025 13:03	Selesai

Gambar 4.4 Tampilan Riwayat Pengaduan

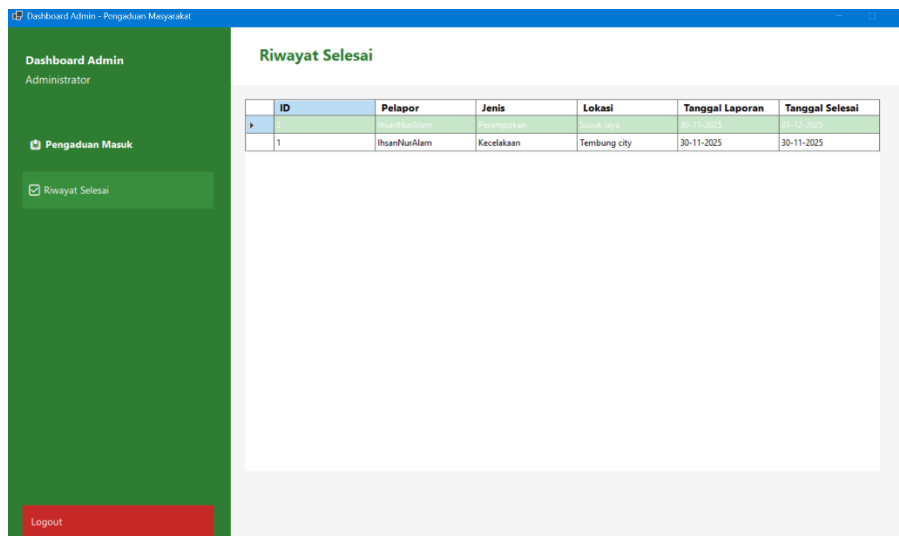
### 4.3.3 Dashboard Admin

1. Pengaduan Masuk : Tampilan ini merangkum Pengaduan Masuk di Dashboard Admin. Halaman ini berfungsi sebagai pusat kendali workflow, tempat Administrator mengelola semua laporan, baik yang masih Pending (Belum Ditindak) maupun yang sudah Diproses. Tabel utama menampilkan detail laporan (ID, Pelapor, Jenis, Status) sehingga Admin dapat cepat mengidentifikasi laporan mana yang perlu ditangani. Tombol Ubah Status adalah fungsionalitas kunci untuk memproses laporan dari Pending hingga Selesai, sekaligus mencatatnya untuk arsip di menu Riwayat Selesai.

ID	Pelapor	Jenis	Lokasi	Tanggal	Status
1	IhsanNurAlam	Perampokan	Medan Tambora	30-11-2025 10:26	Belum Ditindak
2	IhsanNurAlam	Perampokan	Susuk Jaya	30-11-2025 14:47	Sedang Diproses
1	IhsanNurAlam	Kecelakaan	Tembung city	30-11-2025 13:03	Selesai

Gambar 4.5 Tampilan Pengaduan Masuk

2. Riwayat Selesai: Tampilan ini menampilkan Riwayat Selesai di Dashboard Admin. Halaman ini berfungsi sebagai arsip permanen untuk semua laporan yang telah tuntas diproses dan diubah statusnya menjadi Selesai oleh Admin. Tabel ini mencantumkan detail penting seperti ID, Pelapor, Jenis, Lokasi, Tanggal Laporan, dan Tanggal Selesai. Modul ini memastikan Admin memiliki dokumentasi yang terpusat dan terorganisir mengenai semua kasus yang telah ditutup.



The screenshot shows a web application interface. On the left is a dark green sidebar with the text 'Dashboard Admin' and 'Administrator'. Below this are two buttons: 'Pengaduan Masuk' and 'Riwayat Selesai' (which is selected). At the bottom of the sidebar is a red 'Logout' button. The main content area has a title 'Riwayat Selesai' in green. Below the title is a table with the following data:

ID	Pelapor	Jenis	Lokasi	Tanggal Laporan	Tanggal Selesai
2	IhsanNurAlam	Perampokan	Gudak Jaya	30-11-2025	01-12-2025
1	IhsanNurAlam	Kecelakaan	Tembung city	30-11-2025	30-11-2025

Gambar 4.6 Tampilan Riwayat Selesai

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan seluruh tahapan perancangan dan implementasi yang telah dilakukan dalam pengembangan Aplikasi Pengaduan Masyarakat berbasis *desktop* ini, dapat ditarik beberapa kesimpulan utama:

1. Pemenuhan Kebutuhan Fungsional: Aplikasi ini berhasil dikembangkan menggunakan Model *Waterfall* dan memenuhi seluruh spesifikasi kebutuhan fungsional yang telah ditetapkan, terutama dalam hal pemisahan peran (User dan Admin), otentikasi, serta fitur Buat Pengaduan (termasuk unggah foto) dan Riwayat Pengaduan.
2. Integrasi Teknologi: Telah terimplementasi dengan sukses integrasi antara kode program C# yang berjalan di lingkungan Visual Studio dengan Sistem Manajemen Basis Data MySQL (dikelola melalui XAMPP). Integrasi ini didukung oleh MySQL Connector untuk menjamin konektivitas data yang stabil.
3. Implementasi *Workflow* Status: Struktur basis data relasional (*users* dan *pengaduan*) berhasil mendukung *workflow* utama sistem. Administrator (Admin) memiliki kendali penuh untuk memproses laporan dari status Pending menjadi Diproses dan Selesai, sementara Masyarakat (User) dapat melacak status tersebut secara *real-time* melalui modul Riwayat Pengaduan, sehingga menciptakan transparansi.
4. Desain Antarmuka: Perancangan antarmuka yang intuitif dan fungsional (seperti yang terlihat pada Halaman Login, Registrasi, dan Dashboard) memudahkan User dalam melaporkan dan Admin dalam mengelola ribuan laporan yang mungkin masuk.

## 5.2 Saran

Untuk pengembangan aplikasi lebih lanjut dan peningkatan kualitas sistem di masa mendatang, diajukan beberapa saran sebagai berikut:

1. Pencatatan Jejak Audit (Audit Trail): Disarankan untuk menambahkan Tabel Riwayat Status (Audit Trail) yang terpisah. Tujuannya adalah tidak hanya mencatat status terakhir, tetapi juga mencatat secara detail ID Admin yang melakukan perubahan status, waktu, serta tanggapan/catatan penindakan pada setiap tahap perubahan (Pending ke Diproses, dan Diproses ke Selesai). Hal ini akan meningkatkan akuntabilitas dan transparansi proses penanganan.
2. Manajemen Data Master yang Dinamis: Untuk meningkatkan fleksibilitas, Jenis Kejadian sebaiknya tidak menggunakan tipe data *enum* pada tabel pengaduan. Sebagai gantinya, buatlah Tabel Master `jenis_kejadian` yang baru. Ini memungkinkan Administrator untuk menambah, mengurangi, atau mengubah kategori laporan tanpa perlu mengubah struktur tabel utama (*ALTER TABLE*), sehingga sistem lebih mudah dikelola.
3. Pengembangan Platform Multi-Akses: Mengingat bahwa pelaporan sering dilakukan di lokasi kejadian, disarankan untuk mengembangkan antarmuka berbasis *web* atau *mobile* (misalnya Android/iOS) khusus untuk *Role User*. Hal ini akan memperluas aksesibilitas dan kemudahan bagi masyarakat dalam membuat laporan.
4. Penyimpanan Data Foto yang Lebih Kuat: Dalam pengembangan lanjutan, mekanisme penyimpanan foto harus diperkuat. Selain menyimpan *path* (*foto\_path*), tambahkan fitur verifikasi integritas file atau pertimbangkan integrasi dengan layanan penyimpanan *cloud* sederhana untuk mengurangi risiko kehilangan bukti foto jika terjadi kegagalan pada sistem *file server* lokal.

## DAFTAR PUSTAKA

- Albahari, J., & Albahari, B. (2022). *C# 10 in a Nutshell*. O'Reilly Media.
- Fauzi, M. S., & Amin, F. (2012). Perancangan Sistem Informasi Akademik Berbasis Desktop. *Jurnal Tekno Kompak*, 6(1), 1-8.
- IEEE. (2024). *IEEE Definition of Software Engineering*. Diakses 29 November 2025, dari <https://www.computer.org/cms/Computer.org/Publications/pdfs/ugc/Computer-Science-Engineering-Definitions.pdf>
- Jubile Enterprise. (2015). *Kupas Tuntas Visual Studio 2015: C#, Visual Basic, & Web*. Elex Media Komputindo.
- Kendall, K. E., & Kendall, J. E. (2014). *Analisis dan Perancangan Sistem*. Pearson Education.
- Oracle. (2024). *What is a Database Management System (DBMS)?*. Diakses 29 November 2025, dari <https://www.oracle.com/database/what-is-dbms/>
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
- Sommerville, I. (2016). *Software Engineering*. Pearson Education.
- Suryani, N., Setianto, R., & Permana, I. D. (2024). Analisis dan Perancangan Sistem Informasi Pengaduan Pelayanan Publik Berbasis Mobile. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 11(2), 201-210.
- Techopedia. (2023). *What is Graphical User Interface (GUI)?*. Diakses 29 November 2025, dari <https://www.techopedia.com/definition/443/graphical-user-interface-gui>
- Wahyuni, S. R., & Darmawan, D. (2023). Perancangan Aplikasi Pengaduan Masyarakat Berbasis Web Menggunakan Metode Waterfall. *Jurnal Sistem Informasi dan Teknologi Informasi (JSINTI)*, 1(1), 47-56.