

Design and Development of a Web-Based Community Complaint Information System (SIPMAS)

Rivandi Ilham^{*1}, Enrico Abdillah N. P¹, Muhammad Rama Prasetyo¹, Bagus Tri Mahardika¹, Yan Sofyan Andhana Saputra¹

¹Department of Information Technology, Faculty of Engineering, Darma Persada University, East Jakarta 13450, Jakarta, Indonesia

*e-mail: rivandiilham02@gmail.com

Abstract

This study aims to design and develop a web-based public complaint information system named SIPMAS (Sistem Informasi Pengaduan Masyarakat). Previously, municipal and infrastructural grievances in target community sectors were communicated through physical paperwork or unstructured short messaging services, causing immense delays, loss of data tracking, and a lack of transparency. The system was engineered using the Laravel framework, PHP, and a MySQL relational database following the Model-View-Controller (MVC) software design pattern. System evaluation was conducted using functional Black Box testing and deployment response tracking metrics. The results indicate that the SIPMAS portal successfully provides a centralized community environment featuring structured grievance logging, automated urgency sorting, and status validation tracking logs, while offering an administrative control dashboard for comprehensive mitigation management. This digital architecture assists regional administrators in processing neighborhood infrastructure issues more transparently and efficiently, thereby enhancing community administrative accountability.

Keywords: Information System, Public Complaint, Smart Governance, MVC Pattern.

1. INTRODUCTION

The rapid advancement of information technology demands that social institutions and public service agencies adapt their delivery methods to ensure institutional accountability. Local administrative regions frequently encounter challenges in rapidly addressing public infrastructure grievances, emergency utility disruptions, and structural damages. In practice, information regarding community issues is often disseminated through direct verbal communication, physical circular letters, or short messaging applications [1]. While these conventional methods retain their operational utility, such approaches are deemed less effective when grievance data must be systematically archived, updated periodically, or accounted for transparently. Within the ecosystem of public facility management, citizens require structural validation and clear information regarding the verification status of their submitted complaint files, as well as the corresponding repair timelines [2].

Presently, regional community sectors lack a centralized digital media platform, thereby creating structural barriers for residents attempting to access neighborhood profiles, log facility disruptions, or view real-time resolution logs. Incident details and data regarding grievance urgency levels are not comprehensively documented within an easily accessible repository [3]. On the other hand, regional administrators require an optimized control panel to process complaints, partition problem subcategories, update public announcements, and systematically determine mitigation priorities [4].

An information system represents a combination of users, data, procedures, software, and hardware integrated to transform raw inputs into functional, value-added information [5]. For community service institutions, a web-based application can serve as a digital information hub where management agendas, maintenance reports, visual documentation, and public FAQs are structurally presented [6][7]. Therefore, this study aims to analyze the governance gaps in local infrastructure complaints and build the architecture of a web-based public grievance system named SIPMAS using the Laravel framework, PHP, and MySQL. The integration of a MySQL relational database with the Model-View-Controller (MVC) architecture in Laravel ensures that the development of this complaint system proceeds in an organized manner, maintains a robust security framework, and stands ready for future functional scalability [8][9].

2. METHODOLOGY

The system development methodology in this study follows a structured software engineering approach, consisting of system analysis, design, implementation, and testing phases. The development cycle began with field observations to evaluate manual workflow bottlenecks and document tracking delays within the targeted community. Concurrently, a literature review was conducted to analyze Model-View-Controller (MVC) architectural patterns, web security mechanisms, and relational database design rules.



Figure 1. Requirements gathering and consultation session with the Police management.

The application architecture establishes two primary interactive portals: a public portal for citizen engagement and an enclosed administrative panel for verified personnel. The central business routine handles complaint registration and status updating. The routine triggers when a citizen fills out the structured grievance form, declaring incident details and choosing a priority classification. The system captures the inputs, logs the entry into the database, and assigns a unique transaction state tracking token to the ticket. Administrators access the protected control board via credential verification to review the entry, assign operational priorities (low, medium, emergency), and update state tracking headers from pending to resolved, which automatically reflects on the public information dashboard. The system implementation was executed using the Laravel framework, PHP, and a MySQL relational database. System execution evaluated backend routes, data models, object controllers, database migrations, and frontend UI views. Finally, system functionality was assessed using the comprehensive Black Box approach, evaluating front-facing UI elements, operational inputs, role permission constraints, and backend routing actions against the project requirements.

3. RESULT AND DISCUSSION

The implementation results demonstrate that the SIPMAS portal successfully fulfills its core functional design requirements as a web-based public service platform. Built using the Laravel framework, PHP, and a MySQL database, the system isolates user interactions through distinct front-facing views and controller logic. The application interface is divided into a public landing page for community outreach and an enclosed user dashboard for grievance registration and progress tracking.

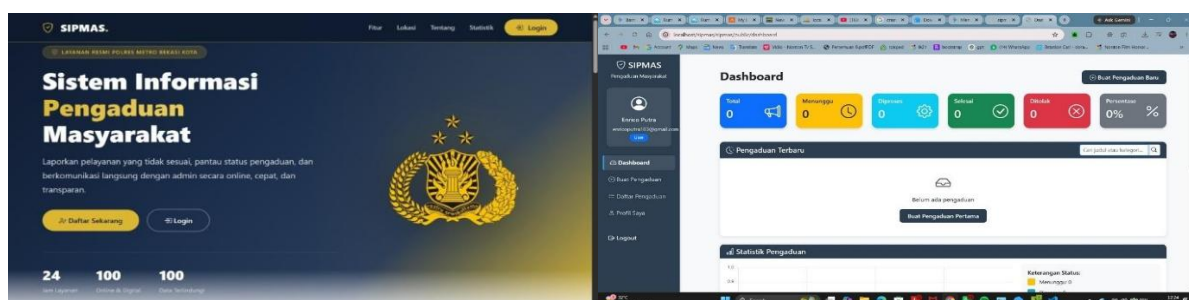


Figure 2. The public landing page interface of the SIPMAS portal & Dashboard

The home interface provides direct access buttons for user registration and authentication, alongside transparent service quality metrics such as active service hours and data protection assurances. The dashboard provides real-time tracking matrices using color-coded metrics to display ticket status categories: pending (Menunggu), in-progress (Diproses), resolved (Selesai), and rejected (Ditolak). This integrated layout allows users to submit new grievances, monitor statistical analysis charts, and review structural verification updates natively without manual data distortions. System functionality was evaluated using the Black Box testing approach across standard operational endpoints, summarized in Table 1.

Table 1. Functional Black Box Testing Results.

No	Scenario	Expected Result	Status
1	Accessing the SIPMAS homepage	Displays system descriptions, quality metrics, and authentication access.	Successful
2	User authentication and login	Redirects verified users to their dedicated system dashboard.	Successful
3	Initializing a new grievance form	Captures reporting metrics, input descriptions, and incident categories.	Successful
4	Displaying ticket status metrics	Updates operational tracking parameters dynamically based on backend logs.	Successful

The evaluation confirms that the current web implementation provides stable data persistence, centralized role authorization, and structured tracking routines that effectively eliminate the documentation delays of traditional communication models.

4. RESULT

The functional evaluations confirm that the developed web application provides significant technical advantages over traditional public report workflows. By utilizing the Model-View-Controller (MVC) architectural pattern, the system maintains strict isolation between backend data processes and user presentation views, preventing unauthorized data injection or state modifications. Centralized relational database migrations in MySQL ensure structured data storage and consistent transaction records, eliminating physical paperwork loss and tracking delays.

However, several developmental constraints remain within the current application ecosystem. The platform depends entirely on standard web server requests, lacking integrated push-notification systems or automated cross-platform messaging hooks (e.g., WhatsApp API) to alert users instantly when their complaint status changes. Furthermore, the system lacks automated analytical tools to group incoming report trends or an independent file exporter engine to instantly generate PDF or Excel summaries for administrative accounting reports. These limitations highlight critical directions for future functional software updates.

REFERENCES

- [1] G. P. Nobre, C. H. G. Ferreira, C. H. G. Ferreira, and J. M. Almeida, "Beyond Groups: Uncovering Dynamic Communities on the WhatsApp Network of Information Dissemination," vol. 12467, Springer, Cham, 2020, pp. 252–266. doi: 10.1007/978-3-030-60975-7_19.
- [2] K. Han, P. C. Shih, and J. M. Carroll, "Local News Chatter: Augmenting community news by aggregating hyperlocal microblog content in a tag cloud," *International Journal of Human-Computer Interaction*, vol. 30, no. 12, pp. 1003–1014, Sep. 2014, doi: 10.1080/10447318.2014.925773.

-
- [3] A. Lara-Millán and M. Guzman-Garcia, "Digital platforms and the maintenance of the urban order," *Social Problems*, vol. 72, no. 2, pp. 675–695, Dec. 2023, doi: 10.1093/socpro/spad053.
 - [4] T. Hatuka and G. Elhanan, "Digitization processes at the neighborhood scale: Infrastructure, governance, community, and practices," *Planning Theory*, vol. 25, no. 1, pp. 75–97, Jul. 2024, doi: 10.1177/14730952241263863.
 - [5] M. E. Cortés-Cediel, A. Segura-Tinoco, I. Cantador, and M. P. R. Bolívar, "Trends and challenges of e-government chatbots: Advances in exploring open government data and citizen participation content," *Government Information Quarterly*, vol. 40, no. 4, p. 101877, Oct. 2023, doi: 10.1016/j.giq.2023.101877.
 - [6] M. E. Mondejar et al., "Digitalization to achieve sustainable development goals: Steps towards a Smart Green Planet," *The Science of the Total Environment*, vol. 794, p. 148539, Jun. 2021, doi: 10.1016/j.scitotenv.2021.148539.
 - [7] G. A. Porumbescu, V. Trehan, and A. Edomwonyi, "Frontline Workers and Civic Tech: Bridging the responsiveness gap in digital client encounters," *Public Administration Review*, vol. 86, no. 4, pp. 1045–1059, Dec. 2025, doi: 10.1111/puar.70066.
 - [8] A. Wijaya, I. Fenriana, L. W. Kusuma, and E. Kusuma, "Web-Based Public Complaint Information System design using the LARAVeL framework," *ALGOR*, vol. 3, no. 2, pp. 35–43, Mar. 2022, doi: 10.31253/algor.v3i2.1036.
 - [9] S. S. D. L. Angeles USA and D. Novoselskyi, "MVCA: an Ultra-Lightweight PHP MVC framework for a faster, cleaner web," *The American Journal of Interdisciplinary Innovations and Research*, vol. 07, no. 12, pp. 01–07, Dec. 2025, doi: 10.37547/tajiir/volume07issue12-01.