

Development of a Web-Based E-Complaint Platform with WebSocket Integration Using the Prototyping Method

Jeki Adi Saputra¹, Joko Sutopo²

^{1,2}Program Studi Informatika, Universitas Teknologi Yogyakarta
Jl. Siliwangi, Jl. Jombor Lor, Mlati Krajan, Sendangadi, Kec. Mlati, Kabupaten Sleman,
Daerah Istimewa Yogyakarta 55284, Indonesia
Email coresponden penulis : 'jeacky1140@gmail.com

Abstrak. Permasalahan pengaduan masyarakat yang ditangani secara manual sering menimbulkan keterlambatan tanggapan, kurangnya transparansi, dan terbatasnya umpan balik. Penelitian ini bertujuan mengembangkan platform E-Complaint berbasis web yang terintegrasi dengan teknologi WebSocket untuk mendukung notifikasi waktu nyata (real-time) sehingga meningkatkan efisiensi dan akuntabilitas layanan publik. Metode penelitian menggunakan pendekatan Prototyping, yang meliputi tahapan penggalan kebutuhan, perancangan sistem, pembangunan prototipe, dan evaluasi fungsional bersama pengguna. Implementasi sistem mencakup fitur pelaporan pengaduan, pelacakan status, dasbor administrator, serta kanal notifikasi instan berbasis WebSocket ketika status laporan berubah. Hasil pengujian fungsional menunjukkan bahwa seluruh proses utama berjalan sesuai kebutuhan pengguna, sistem memberikan respons cepat tanpa jeda signifikan, dan komunikasi antara pengguna dan administrator menjadi lebih efisien dibandingkan proses manual. Integrasi WebSocket terbukti mampu meningkatkan kecepatan penyampaian informasi dan transparansi penanganan pengaduan masyarakat. Ke depan, pengembangan sistem difokuskan pada peningkatan fitur analitik dan optimasi performa layanan berbasis web.

Kata Kunci : kata kunci minimal 3 kata (jenis huruf, ukuran dan style sama dengan pembuatan Abstrak).

Abstract. Manual handling of public complaints often results in delayed responses, lack of transparency, and limited feedback. This study aims to develop a web-based E-Complaint platform integrated with WebSocket technology to support real-time notifications, thereby improving the efficiency and accountability of public services. The research method used a prototyping approach, which included the stages of requirements gathering, system design, prototype development, and functional evaluation with users. The system implementation included complaint reporting, status tracking, administrator dashboard, and WebSocket-based instant notification channels when the report status changed. Functional testing results show that all main processes run according to user needs, the system responds quickly without significant delays, and communication between users and administrators is more efficient than manual processes. WebSocket integration has proven to increase the speed of information delivery and transparency in handling public complaints. Going forward, system development will focus on improving analytical features and optimizing web-based service performance.

Keyword : E-Complaint, Prototyping, Public Services, Web Application, WebSocket

PENDAHULUAN

Public complaints are a crucial feedback mechanism for improving the quality of public services and strengthening government accountability. In the context of governance in the digital age, complaint channels should ideally function efficiently, transparently, and be easily accessible in order to maintain public trust. However, to date, the complaint process still relies heavily on conventional mechanisms such as suggestion boxes or direct visits, which create various obstacles such as delayed responses, difficulty in tracking the status of reports, and a lack of transparency in follow-up. This phenomenon can be seen, for example, in a case study in Wonokerso Village [1] where the use of manual complaint channels resulted in low citizen participation. More broadly, an evaluation of the national SP4N-LAPOR! platform[2]. shows that although the system has brought progress, the



average time for report verification still takes several days, emphasizing the need to strengthen the coordination of the complaint handling process to make it faster, more measurable, and more responsive at the implementation level [3].

In response to various inefficiencies in manual systems, research is now shifting towards digital transformation through the development of web-based complaint systems. A number of previous studies have focused on the design of basic features, such as administrative dashboards and online reporting, the development of complaint management platforms in specific environments such as educational institutions [4]. and the application of Geographic Information Systems (GIS) for spatial analysis of incident reports [5]. Other efforts to improve response speed include integration with external services, for example through automatic notifications sent via third-party Application Programming Interfaces (APIs) such as WhatsApp. In general, the implementation of this digital system has proven to improve the efficiency, responsiveness, and accountability of public service providers [6].

Despite a massive shift to web platforms, existing digital solutions still leave critical gaps in persistent real-time information synchronization. Most current systems still rely on inefficient mechanisms such as polling or page refreshes to update status. Even when adopting instant notifications, systems often depend on external services [7]. This approach is prone to latency, burdens server resources with repetitive overhead, and fails to create persistent two-way interactions. In fact, the need for truly real-time status updates and instant process visibility is crucial to restoring public trust and ensuring transparency in complaint handling.

In response to critical gaps that require low- latency communication and persistent connections, this research focuses on developing a web-based E- Complaint platform that adopts WebSocket technology. This technology was chosen because it provides a superior full-duplex connection compared to traditional methods such as Ajax or Polling, and ensures instant notifications and status synchronization without repetitive HTTP overhead [8]. A latency comparison analysis also shows that WebSocket performs better in real-time interaction scenarios[9]. To produce a responsive system that meets user needs, this research uses the Prototyping method, which allows user involvement from the early stages of development. This approach facilitates an iterative process in prototype creation [10]. so that the system can develop adaptively and quickly adjust to the specific needs of public service users.

This study aims to design and implement a WebSocket-based E-Complaint system that provides instant notifications and real-time status synchronization between users and administrators to improve transparency in the handling process. Its main contribution is a native real-time interaction model that operates without third-party services and demonstrates higher efficiency through low-latency technology. The findings are expected to guide government agencies in strengthening digital governance.

METODOLOGI PENELITIAN

Research Design

his study employs the Prototyping system development method. This method was chosen for its ability to produce systems that meet user needs through an iterative and participatory process. Each stage of development allows direct user involvement to provide feedback on the system design, ensuring that the final system meets the actual needs of public service processes being analyzed.



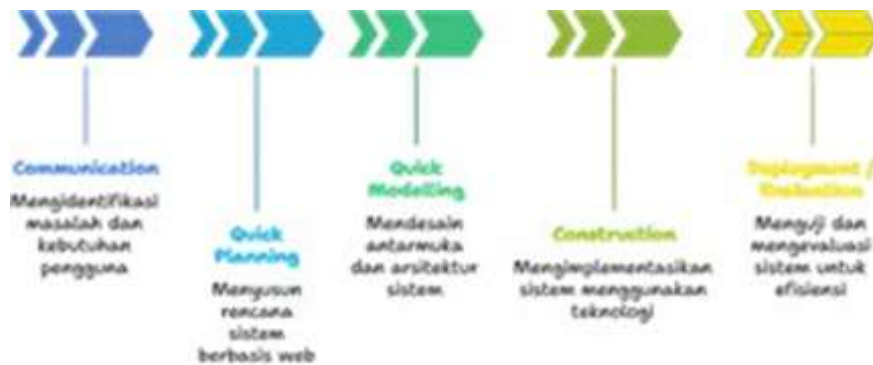


Figure 1. Stages of the prototyping method

This approach applies observation and interviews to explore user needs, challenges, and expectations in public complaint-handling processes. The analysis revealed that manual complaint management often leads to delayed responses, inconsistent record-keeping, and limited transparency. Based on these findings, a web-based public complaint system was designed and integrated with WebSocket technology to deliver real-time status updates and enhance communication efficiency between users and administrators.

During the development phase, the system was built using the Laravel framework with MySQL as the primary database. The WebSocket integration was implemented using the Laravel WebSockets and Laravel Echo packages, enabling two-way communication between users and administrators without requiring page reloads.

The WebSocket implementation utilizes Laravel events such as `ComplaintCreated` and `ComplaintStatusUpdated`, which are broadcast directly to clients to provide instant notifications whenever a complaint is submitted or its status is updated. This technology was chosen for its low latency and capability to maintain efficient data synchronization between the server and client sides.

Data Collection Technique

Data collection in this study was carried out to obtain the necessary information for designing and developing a web-based public complaint system integrated with WebSocket technology. The data were collected using three complementary approaches: observation, interviews, and literature study. Observation and interviews were conducted in one of the regional government public service offices as a reference for understanding complaint-handling workflows, while the literature study supported the formulation of the system's requirements and architecture.

Observasi

Observation data were obtained by directly reviewing the manual workflow of public complaint submission and processing. The purpose was to understand the process, identify inefficiencies, and define user needs for a faster and more transparent system. The observation techniques included:

- Monitoring the conventional reporting process to identify stages requiring digital transformation.
- Analyzing the response time of officers and communication delays between users and administrators.
- Recording user-officer interactions to determine the need for two-way communication, which was later implemented using WebSocket technology.

Interviews

Interviews were conducted to gather in-depth information regarding system requirements from both user and administrator perspectives. The interview process included:

- Interviews with complaint administrators to explore problems in manual systems and



- expectations for the new web-based platform.
- b. Interviews with citizens to understand their experiences in submitting complaints and their expectations for *real-time* notifications.
- c. Identifying additional information related to interface design, usability, and report- tracking requirements.

Literature Review

The literature review was conducted to obtain theoretical and technical references supporting the development of the web-based E- Complaint system. The review included:

- a. Studying journals, scientific articles, and books discussing public complaint information systems and web-based application development.
- b. Reviewing previous studies on the use of WebSocket technology for real-time communication in information systems.
- c. Examining theories and concepts of the Prototyping method, as well as technical documentation of Laravel, MySQL, and Laravel WebSockets frameworks.

In developing the web-based E-Complaint system, this study utilized several integrated software components designed to support an efficient, responsive, and real-time system development process. The selection of these tools was based on their compatibility, flexibility, and ability to support two-way communication using WebSocket technology

The primary framework employed is Laravel, a PHP-based framework well known for its structured workflow and efficient backend management. Laravel was chosen because it offers built-in event broadcasting capabilities and full support for Laravel Echo and Laravel WebSockets, which enable the real-time notification feature in this complaint management platform. MySQL was used as the database management system due to its reliability and seamless integration with Laravel. It stores all user data, complaint records, report statuses, and activity logs, ensuring information consistency and security.

During development, Visual Studio Code served as the main code editor for its flexibility, lightweight performance, and integration support with extensions such as Git and the built-in terminal. Additionally, Node.js and NPM (Node Package Manager) were used to manage front-end dependencies and compile JavaScript assets, including configurations for WebSocket operations.

The real-time notification feature was implemented through Laravel Echo and Laravel WebSocket, which enable instant bidirectional communication between users and administrators without page reloads. This integration plays a crucial role in enhancing process transparency and service efficiency.

HASIL DAN PEMBAHASAN

This study explores the outcomes of creating an online E-Complaint platform that incorporates WebSocket technology, built with the Prototyping approach. The central point of the discussion is the system's operation, the ways users interact with the system's functions, and how the user interface is put into action. To fully explain how the system runs, this part starts with the Flowchart, showing each step of the complaint management process, from when it's first sent to when users get live updates.



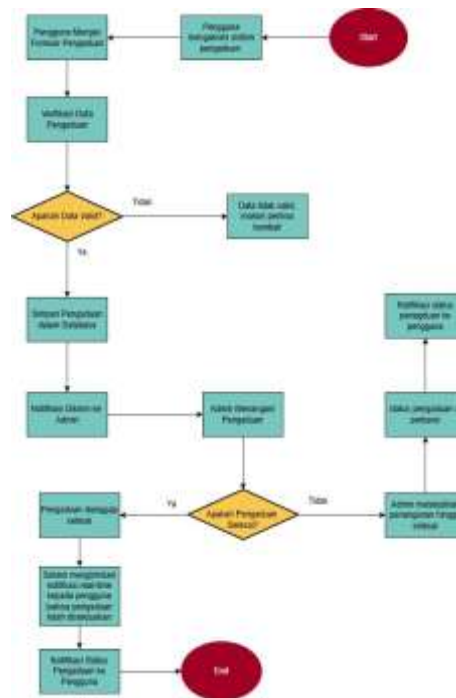


Figure 2. E-Complaint System Flowchart

The flowchart serves to describe the overall workflow of the system, starting from complaint submission to real-time notification delivery to users. This diagram helps visualize the logical process of the system and highlights how WebSocket integration enhances service efficiency and transparency.

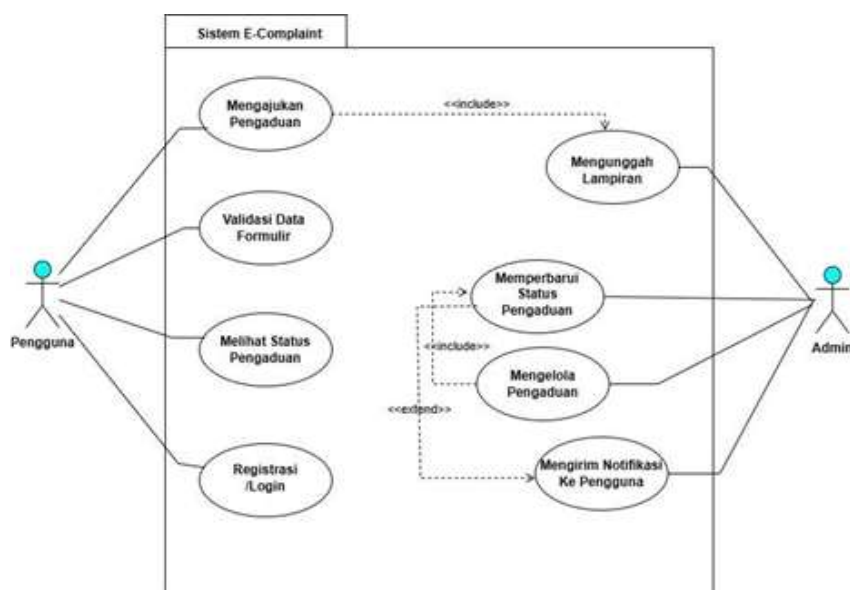


Figure 3. Use Case Diagram

This diagram of use cases visually represents how users and administrators interact within the online E-Complaint platform. Individuals have the ability to sign up, file their grievances along with supporting documents, and keep track of where their complaint is in the process. Those with administrative privileges can validate complaints, handle them effectively, and change the status of a complaint as needed. Every instance when a complaint status is modified automatically prompts immediate alerts through the incorporated WebSocket technology.

System Implementation

The implementation stage represents the realization process of the system design developed using the Prototyping method. The resulting system is a web-based E-Complaint platform integrated with WebSocket, which serves as an online and real-time public complaint medium. The system was developed using the Laravel framework and MySQL as the database, with Laravel WebSockets integration to support two-way communication between users and administrators without requiring page reloads.



Figure 4. Main Page Display

The main page of the E-Complaint system serves as the initial interface for users to access the platform. This page displays the service slogan, login and register buttons, and a link to the system demo. The interface is designed to be responsive and user-friendly, enabling users to easily access complaint services quickly and transparently.



Figure 5. Registration Page Display

To begin using the E-Complaint platform, new users must first establish an account through the registration page. They will be asked to provide their complete name, a working email address, and a secure password, which they will need to confirm. The user interface has been carefully crafted to be straightforward and adapt to different screen sizes, streamlining the registration procedure, which includes a mandatory agreement to the terms and conditions.



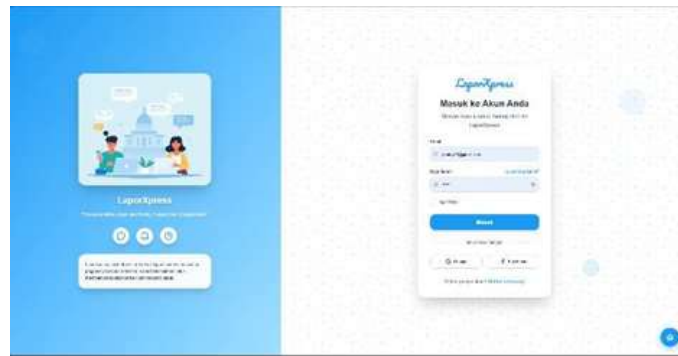


Figure 6. Login Page

The login page functions as the primary entry point where users verify their credentials to gain entry into the E-Complaint system. To log in, users must provide the email address and password they used during registration. To improve the authentication experience, features such as "remember me" and password recovery options are available.



Figure 7. Home Page

The home page is the main display after users successfully log into the system. On this page, users can access key features such as submitting new complaints, viewing report statistics, and monitoring the transparency level of public services.



Figure 8. Complaint Tracking Page

The complaint tracking page displays the current status of reports submitted by users. The system allows users to monitor complaint progress in real-time by entering the report number received during submission. Each status update ranging from received, verified, and processed to completed is presented transparently with a progress indicator. WebSocket integration ensures that status updates are delivered instantly without requiring a page reload.



Figure 9. Complaint Tracking Page

This page presents analytics and visualization of community complaints processed by the system. The data is displayed in trend and category distribution charts to illustrate completion rates and types of reports submitted.

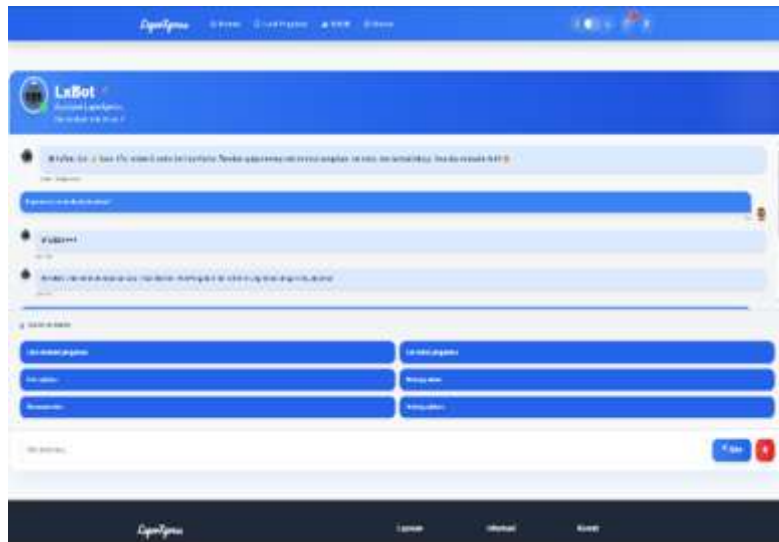


Figure 10. LxBot Chatbot Assistance Page

This chatbot feature provides interactive assistance to users in operating the system. LxBot can respond to inquiries about submitting complaints, checking report status, and providing general information about the application.

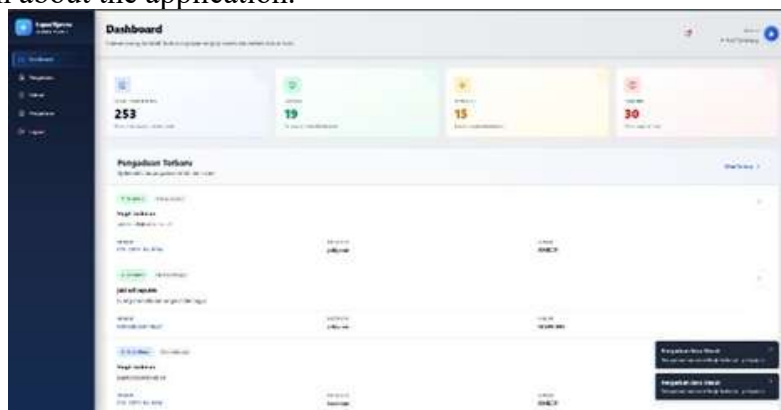
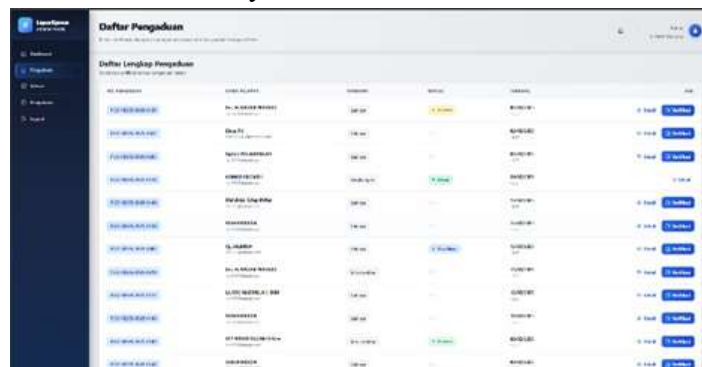


Figure 11. Admin Dashboard Page



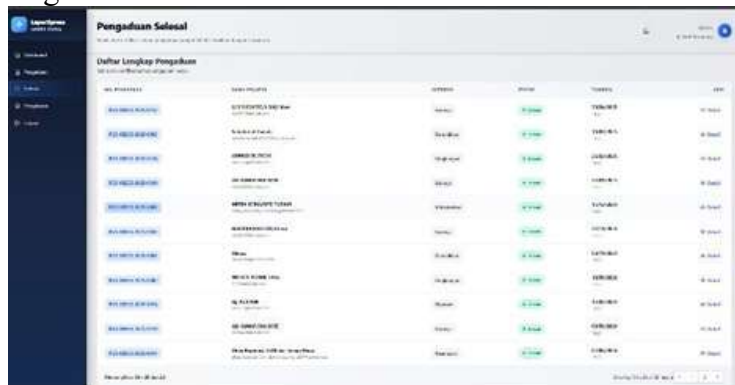
This page provides a real-time summary of complaint statistics, including total reports, completion status, and recent activity. It enables administrators to monitor complaint progress and manage follow-up actions more efficiently.



The screenshot displays the 'Daftar Pengaduan' (Complaint List) page. It features a table with columns for 'No. Laporan', 'Nama Pelapor', 'Alamat', 'Kategori', 'Status', 'Tanggal', and 'Aksi'. The table lists several complaints, each with a corresponding status (e.g., 'Belum Dikaji', 'Dikaji', 'Ditutupi') and a date. The interface includes a sidebar with navigation options like 'Dashboard', 'Pengaduan', 'Aksi', 'Pengaduan', and 'Aksi'.

Figure 12. Admin Complaint List Page

This page allows administrators to manage all incoming complaint reports. Admins can view report details, verify data, and update complaint statuses to ensure the follow-up process runs efficiently and in an organized manner.



The screenshot displays the 'Pengaduan Selesai' (Resolved Complaints) page. It features a table with columns for 'No. Laporan', 'Nama Pelapor', 'Alamat', 'Kategori', 'Status', 'Tanggal', and 'Aksi'. The table lists several resolved complaints, each with a corresponding status (e.g., 'Ditutupi', 'Ditutupi', 'Ditutupi') and a date. The interface includes a sidebar with navigation options like 'Dashboard', 'Pengaduan', 'Aksi', 'Pengaduan', and 'Aksi'.

Figure 13. Resolved Complaints Page

This page displays a list of complaints that have been fully processed by the administrator. It facilitates monitoring of final report outcomes while ensuring transparency and accountability in the complaint resolution process.

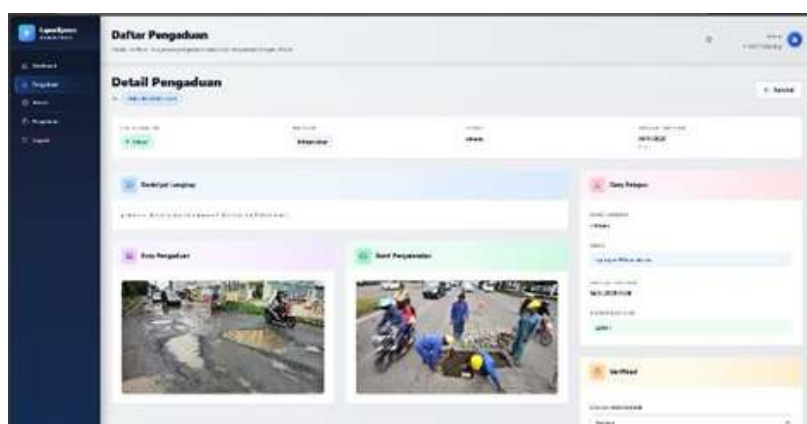


Figure 14. Admin Complaint Detail Page

This page displays complete information about a complaint, including reporter data, problem description, photo evidence, and resolution proof. Through this page, the administrator can verify and confirm that the report has been properly and transparently resolved.

The developed web-based E-Complaint system achieves end-to-end report handling. Administrative review of evidence and quick status updates significantly enhance traceability and ensure faster, transparent complaint resolution. These results confirm increased efficiency and higher citizen satisfaction found in previous online dashboard implementations [11], [12].

System Testing

To make sure every aspect of the web-based E- Complaint system that includes WebSocket worked as it should, system testing was performed, checking that it followed all set rules and designs. The tests looked at how well the system worked, checking things like sending complaints, making sure data was correct, sending alerts right away, and watching how reports were doing. Black Box Testing was used for testing, checking each function by looking at what it should do without looking at the program's inner workings.

Table 1. System Testing for User Side

No	Test Scenario	Expected Result	Status
1	Register an account using an invalid email format	The system rejects the input and displays a validation error message	Valid
2	Register an account with complete and correct data	The system accepts the data and successfully creates a new account	Valid
3	Log in with correct credentials	The system verifies the account and redirects the user to the main page	Valid
4	Log in with incorrect credentials	The system denies authentication and displays an error message	Valid
5	Submit a complaint form with complete data	The system stores the data and displays a successful submission notification	Valid
6	Submit a complaint form without filling in the description	The system rejects the submission and displays a required-field warning message	Valid
7	Track a complaint using a valid tracking number	The system displays the detailed complaint status	Valid
8	Track a complaint using an unregistered tracking number	The system displays a “data not found” message	Valid
9	Administrator updates the complaint status	The system sends a real- time notification to the user	Valid
10	User logs out from the system	The system terminates the	Valid

Based on what was found during testing using the Black Box Testing method, everything on the user's side works as it should. When someone puts in wrong info, the system turns it down, tells people clearly about mistakes, and handles info the right way when it is correct. This proves the system is reliable and does what it's supposed to.



Table 2. System Testing for Admin Side

No	Test Scenario	Expected Result	Status
1	Admin logs in with valid credentia	The system successfully grants access and displays the admin dashboard	Valid
2	Admin logs in with invalid credentials	The system rejects authentication and displays an error message	Valid
3	Admin dashboard displays complaint statistics (total, completed, in process, pending)	Statistical data are displayed accurately according to the latest system updates	Valid
4	Admin receives real-time notifications of new complaints	Notification automatically appears when a new report is submitted	Valid
5	Admin opens the complaint list page	The system displays all complaint records including name, category, status, and location	Valid
6	Admin verifies a user complaint	The system updates the status to "Verified"	Valid
7	Admin updates the complaint status to "Completed" and uploads proof of completion	The system updates the report status and displays the uploaded completion evidence	Valid
8	Admin logs out of the system	The system displays a "Logout Successful" message and redirects to the login page	Valid

Using the Black Box Testing method, system testing on the administrator's side was done to confirm that all main parts worked the way they were expected to. Tests were focused on logging in, handling complaints, managing alerts in real time, and double-checking reports. The results show that every function tested worked properly, and the system fulfilled all expected job requirements without any problems.

CONCLUSION

This research successfully developed a web- based E-Complaint system integrated with WebSocket technology to improve the transparency and efficiency of public services. The system, designed using a prototyping approach, is able to meet user needs in the reporting process, status tracking, and real-time notifications between users and administrators. Testing using the Black Box method showed that all main functions of the system ran as expected without any logical or display errors. WebSocket integration proved effective in reducing notification latency, accelerating information flow, and increasing accountability in handling public complaints. Overall, the implementation of this system shows significant potential in supporting the digital transformation of public services based on information disclosure.

FUTURE WORK

Further research is planned to add a direct communication feature between users and administrators through the integration of a real- time chat module to make the report follow-up process more interactive. In addition, the system will be developed with a more specific complaint location tracking feature using Google Maps API integration to provide accurate visualization of the



incident location. This development is expected to improve the user experience and strengthen the transparency and accountability of web-based public complaint services.

DAFTAR PUSTAKA

- [1] M. K. Khairudin, Mursalim, and T. Aprilia, "Rancang Bangun Sistem Pengaduan Masyarakat Berbasis Web di Desa Wonokerso," *Infomatek*, vol. 26, no. 2, pp. 223–230, 2024, doi: <https://doi.org/10.23969/infomatek.v26i2.19364>.
- [2] N. Made and W. Premaiswari, "The Effectiveness of the SP4N-LAPOR! as a National Public Service Complaint Management Application," *JKAP (Jurnal Kebijak. dan Adm. Publik)*, vol. 28, no. 1, pp. 36–51, 2024, [Online]. Available: <https://journal.ugm.ac.id/jkap>
- [3] I. Ibtu, I. A. P. S. Widnyani, and C. Simabura, "The Decision-Making Based on Public Aspirations and Complaints through SP4N-Lapor! in Three Provinces of Indonesia," *J. Manaj. Pelayanan Publik*, vol. 9, no. 3, pp. 633–646, 2025, doi: <https://doi.org/10.24198/jmpp.v9i3.60928>.
- [4] V. Oguntosin, M. Oluwadurotimi, A. Adoghe, A. Abdulkareem, and G. Adeyemi, "Development of a Web-Based Complaint Management Platform for a University Community," *J. Eng. Sci. Technol. Rev.*, vol. 14, no. 1, pp. 150–159, 2021, doi: <https://doi.org/10.25103/jestr.141.17>.
- [5] F. Rusdhianto and R. Imanda, "Perancangan Sistem Pengaduan Masyarakat Berbasis Web Dengan Terintegrasi GIS Studi Kasus Muara Baru Jakarta Utara," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 14, no. 2, pp. 371–382, 2025, doi: <https://doi.org/10.30591/smartcomp.v14i2.8592>.
- [6] H. J. Seta, D. G. Suharto, and K. Setyowati, "Digitalization: Innovation in Public Services (Case Study Implementation of the Complaints Service Unit in Surakarta (ULAS) and Klaten Regency Industry and Manpower Office)," *J. Gov. Civ. Soc.*, vol. 8, no. 2, p. 161, 2024, doi: <https://doi.org/10.31000/jgcs.v8i2.10884>.
- [7] A. E. Nugraha and E. Rahmawan Pramudya, "Web-Based Public Street Lighting Complaint Application with Realtime Whatsapp Notification Using Prototype Method in Pemalang Regency," *J. Appl. Intell. Syst.*, vol. 9, no. 1, 2024, doi: <https://doi.org/10.62411/jais.v9i1.10445>.
- [8] D. Sereda, "Real-time interaction on the web: how WebSockets change application performance," *Int. J. Eng. Comput. Sci.*, vol. 14, no. 5, pp. 27229–27233, 2025, [Online]. Available: <https://www.researchgate.net/publication/392604799>
- [9] M. Hlayel, H. Mahdin, and H. A. M. Adam, "Latency Analysis of WebSocket and Industrial Protocols in Real-Time Digital Twin Integration," *Int. J. Eng. Trends Technol.*, vol. 73, no. 1, pp. 120–135, 2025, doi: <https://doi.org/10.14445/22315381/IJETT-V73I1P110>.
- [10] G. Enstayn, A. Kustanto, H. P. Chernovita, and P. Korespondensi, "Perancangan Sistem Informasi Manajemen Berbasis Web (Studi Kasus: PT. Unicorn Intertranz)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 4, pp. 719–728, 2021, doi: <https://doi.org/10.25126/jtiik.202184849>.
- [11] N. Alfi, S. Purba, M. Ula, and S. Fachrurrazi, "Aplikasi Sistem Informasi Pengaduan Masyarakat Berbasis Website Pada Dinas Komunikasi Informatika Dan Persandian Kota Lhokseumawe," *J. Sist. Inf. (SISFO KP)*, vol. 4, no. 1, pp. 46–63, 2023, [Online]. Available: <https://ojs.unimal.ac.id/sisfo/article/view/19406>
- [12] A. R. Mahbub, K. Hantoro, and Y. Suyani, "Analisa dan Perancangan Layanan Pengaduan Masyarakat Desa Berbasis Web dengan Pendekatan UML," *Fakt. Exacta*, vol. 13, no. 1, p. 26, 2020, doi: <https://doi.org/10.30998/faktorexacta.v13i1.4866>.

