



Development of a Web-Based Campus Facility Complaint Information System Prototype Using Rapid Application Development

Muh. Akhyarul Insan^{1*}, Fajar Octanouva², Nur Vitriati³, Sabila Wahyu Pratama⁴, Aditya Sendy Ardiansyah⁵

¹⁻⁵ Universitas Islam Sultan Agung, Semarang, Indonesia

email: akhyar2442@gmail.com

Article Info :

Received:
20-02-2026
Revised:
01-03-2026
Accepted:
10-03-2026

Abstract

Campus facility complaint management requires a structured mechanism for recording, monitoring, and tracing reported problems to support transparent service processes. This study developed a web-based Campus Facility Complaint Information System (SIPFAS) prototype using the Rapid Application Development (RAD) method. The development process comprised requirement planning, user design, construction, and cutover and verification stages, with the prototype designed to support service information access, authentication, complaint submission, status monitoring, and complaint-detail retrieval. Evaluation focused on the public page, login interface, user dashboard, and complaint-detail page containing the status history of complaint PGD-002. Interface consistency verification showed that the dashboard summary was arithmetically consistent, with 3 complaints in Waiting status, 4 in In Process status, and 5 in Completed status, resulting in a total of 12 complaints. The PGD-002 identity was also traceable across the dashboard and complaint-detail interfaces. However, two inconsistencies were identified, namely a discrepancy between the complaint date of 19 August 2026 and the timeline submission timestamp of 20 August 2026 at 09:14, and a mismatch between the administrator-verification timestamp and its visual state. The prototype establishes a structured complaint-monitoring workflow, although data consistency, functional testing, authorization, backend persistence, and usability evaluation require further verification before full implementation.

Keywords: Complaint Management, Monitoring, Prototype, Rapid Application Development, Web Application.



©2026 Authors.. This work is licensed under a Creative Commons Attribution-Non Commercial 4.0 International License.
(<https://creativecommons.org/licenses/by-nc/4.0/>)

INTRODUCTION

Campus facilities and infrastructure constitute an important part of the learning environment because they support academic activities, administrative processes, and student services, while their reliability increasingly depends on the availability of effective mechanisms for reporting and managing facility-related problems. When facilities such as air-conditioning systems, projectors, Wi-Fi networks, lighting, chairs, and other equipment experience damage, reporting mechanisms that remain informal or undocumented can make subsequent handling difficult to monitor and weaken the traceability of corrective actions. Nurhasan et al. (2022) demonstrated that verbal reporting of campus facility damage makes it difficult for reporters to determine the status of handling and limits the availability of evidence regarding the current condition of reported facilities. Widiatry et al. (2026) similarly identified informal reporting as a source of delayed responses, inadequate documentation, and limited transparency in campus facility management. The problem extends beyond the act of submitting a complaint because an unstructured reporting process can disconnect the initial information from verification, follow-up, and completion of the repair process. As higher education institutions increasingly adopt web-based information services, facility complaint management requires a mechanism capable of transforming fragmented reports into structured information that can be accessed, monitored, and updated throughout the handling process.

Within higher education environments, the need for a structured complaint channel concerns not only physical facility damage but also the broader student experience of institutional services, because students need a clear mechanism through which problems can be communicated and their handling can be monitored. Alfawaz et al. (2022) demonstrated the relevance of web-based student complaint management systems in tertiary institutions for organizing complaint submission and

institutional responses through a digital platform. Aulia et al. (2025) developed a web-based student complaint information system to replace a manual suggestion-box mechanism and emphasized the potential of digitalization to improve the efficiency and responsiveness of complaint services. Sitanggang et al. (2024) further demonstrated that a web-based complaint system can facilitate complaint submission, status monitoring, and access to complaint history within a university environment. These findings indicate that digital complaint systems can shift institutional complaint handling from an isolated reporting activity toward a traceable information process in which users can follow the development of their reports. The effectiveness of such systems, however, also depends on whether users can identify the appropriate complaint channel and understand how the submitted information will be processed. Hidhayah et al. (2024) found that unclear complaint channels at the faculty level can create difficulties for students in determining where and how problems should be submitted, indicating that accessibility and clarity are integral components of digital complaint management.

Digitalization of facility complaints has been implemented in various institutional contexts, with previous studies demonstrating that structured information attributes and clearly defined workflows are necessary to support effective complaint management. Zuliyanto and Asriningtias (2025) emphasized the importance of location, description, photographic evidence, and status-monitoring features in applications for public-facility complaints because these attributes provide more complete information for subsequent handling. Anisa et al. (2026) developed a public-facility complaint information system involving report submission, verification, disposition, follow-up, and monitoring, illustrating the need to connect user reports with administrative handling processes. Phratama et al. (2024) similarly positioned a web-based student feedback system as a mechanism for improving the efficiency of complaint processing within a higher-education environment. These studies collectively demonstrate that the value of a web-based complaint system lies not merely in replacing manual reporting but in establishing centralized information, traceable processes, and visible complaint-status development. However, the breadth of functionality implemented in previous systems varies according to institutional needs, meaning that a campus facility complaint prototype does not necessarily require every advanced processing mechanism from the outset. A focused prototype can instead prioritize the coherence of essential reporting, monitoring, and status-history functions before additional automation and decision-support capabilities are introduced.

From the software-development perspective, Rapid Application Development (RAD) is relevant to the construction of a campus facility complaint prototype because its iterative development orientation enables requirements and interface components to be adjusted rapidly according to identified user needs. Wicaksono et al. (2025) applied RAD to a web-based school facilities and infrastructure complaint application to address delays in responses and limitations associated with manual data management. Yanuardi et al. (2024) used RAD to develop a public-service complaint system supporting complaint submission, status monitoring, and officer responses, demonstrating the compatibility between iterative development and complaint-management requirements. Suni and Sihombing (2023) also showed that iterative RAD development can support the production of functional complaint applications that are refined according to user input. At the broader information-system level, Nilawati and Martin (2023) demonstrated that RAD can connect requirements analysis, system design, construction, and implementation within a relatively short development cycle. Hidayatulloh and Patyani (2024) similarly confirmed the applicability of RAD to web-based information-system development, while Maulana et al. (2026) provided recent evidence of RAD application in the development of institutional web-based information systems. The relevance of RAD in this study therefore lies not only in accelerating system construction but also in providing an iterative framework through which the structure and consistency of the prototype can be examined before full-scale implementation.

Table 1. Comparison of Previous Studies

Research	Context/Method	Main Focus	Position toward SIPFAS
Wicaksono et al. (2025)	Campus facilities and infrastructure complaints; RAD	Web-based complaint system to reduce manual management	Same use of RAD; SIPFAS focuses on campus facilities

Yanuardi et al. (2024)	Public-service complaints; RAD	Complaint submission, status monitoring, and officer responses	Strengthens status monitoring and iterative development
Nurhasan et al. (2022)	Campus facility damage	Monitoring and prioritization of repairs	SIPFAS is simpler and focuses on reporting/status flow
Sitanggang et al. (2024)	Web-based student complaints	Complaint submission, status, and history	Supports the need for a centralized complaint channel on campus
Widiatry et al. (2026)	Campus facility damage	Prioritization, SLA, and Black Box Testing	SIPFAS does not yet implement priority/SLA; functional verification is prepared
Khansa et al. (2024)	UI/UX for facility complaint services	Ease of user interaction	Reinforces the need for a simple and consistent interface

Although research on complaint systems and facility reporting has developed substantially, the emphasis of previous studies remains diverse, creating a gap concerning the integration of basic complaint functionality with explicit consistency checking across interconnected interfaces. Widiatry et al. (2026) introduced prioritization and automatic Service Level Agreement mechanisms for campus facility damage, whereas Nurhasan et al. (2022) emphasized monitoring and prioritization of facility repairs, showing that previous research has frequently focused on improving operational decision-making after complaints are received. Saputra and Sutopo (2025) extended complaint-system functionality through WebSocket integration, while Fakhriyah and Kurniawan (2024) employed rule-based processing to structure web-based complaint management. These approaches demonstrate that complaint systems can be expanded through automation, real-time communication, and rule-based processing, yet such extensions can also increase system complexity beyond the immediate requirements of an initial campus facility complaint prototype. From the interaction perspective, Khansa et al. (2024) highlighted the importance of user-oriented interface design in public-facility complaint services, particularly because interaction difficulties can emerge during essential processes such as authentication and complaint submission. The literature consequently indicates that technical functionality, development methodology, and interface usability have been addressed across different studies, while the explicit relationship between complaint-flow functionality and consistency of information across the main interfaces of a campus facility complaint prototype remains insufficiently emphasized.

This study aims to develop a web-based Sistem Informasi Pengaduan Fasilitas (SIPFAS) prototype using the Rapid Application Development methodology and to evaluate the consistency of information presented across its main interfaces. The research focuses on establishing a structured complaint flow connecting the public page, login interface, dashboard, complaint-detail page, and status-history information within a unified prototype. The prototype is designed to provide a more organized mechanism for documenting campus facility problems while allowing users to identify the progress and current status of submitted complaints. The evaluation specifically examines the relationship and consistency of information presented between interconnected interface components and establishes a basis for functional verification before the system proceeds toward broader implementation. Methodologically, the study contributes an application of RAD that emphasizes iterative realization of a complaint-system prototype together with explicit examination of information consistency across the principal user interfaces. The study does not position SIPFAS as the first campus facility complaint system, but rather as a focused prototype that addresses the practical need to connect reporting, status visibility, and complaint history within a coherent user journey. The resulting prototype is expected to provide a methodological and practical foundation for subsequent development involving more advanced features such as complaint prioritization, Service Level Agreement mechanisms, automated notifications, and broader integration with campus facility-management processes.

RESEARCH METHODS

This study is classified as a technical system-development study with empirical functional verification, because the research develops a web-based SIPFAS prototype and prepares systematic verification of its observable functions without involving human respondents or experimental treatment.

The system architecture consists of four principal interface components, namely the public SIPFAS page, login interface, user dashboard, and complaint-detail page containing the status history of complaint PGD-002. The development process employs Rapid Application Development (RAD) because its iterative structure supports rapid refinement of system requirements, interface design, and functional components. Wicaksono et al. (2025) demonstrated the applicability of RAD in developing web-based facilities and infrastructure complaint systems within an educational environment. The requirement-planning stage establishes five core functions, namely access to service information, authentication, complaint submission, status monitoring, and complaint-detail access, which are subsequently translated during the user-design stage into navigation structures, dashboard summaries, status cards, complaint tables, and handling timelines. The construction stage implements the approved interface structure into the web-based prototype, while the cutover stage prepares the resulting prototype for interface consistency verification and functional Black Box Testing, with the complete development sequence illustrated in Figure 1.

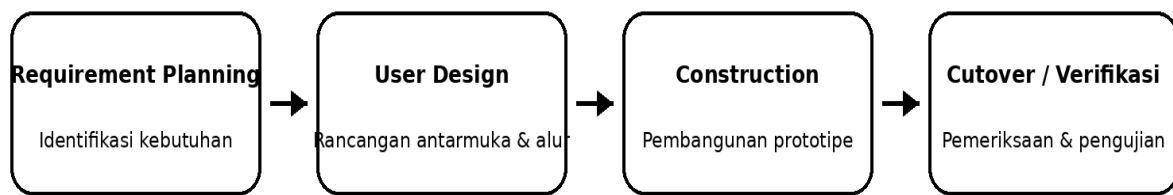


Figure 1. Rapid Application Development Stages for SIPFAS

The verification procedure combines descriptive interface-consistency analysis with Black Box Testing to determine whether the prototype presents coherent information and whether its specified functions produce the expected observable outputs. The verification objects comprise the public SIPFAS page, login page, user dashboard, and PGD-002 complaint-detail page, while the examined information includes complaint identity, category, location, date, status, summary counts, and status-history timestamps. Zidan et al. (2022) showed that Black Box Testing can verify application functions through observable inputs and outputs without requiring examination of the internal program structure. The functional scenarios cover valid and invalid login, empty authentication fields, complaint creation, image upload, complaint-list access, complaint-detail retrieval, status transitions, complaint completion, and authorization restrictions, as specified in Table 2. The validation criterion is based on whether each test input generates the predefined expected output, whereas interface consistency is assessed descriptively by comparing identical complaint attributes across interconnected pages and recording discrepancies in values, labels, status representations, or timestamps. Anisa et al. (2026) applied Black Box Testing to evaluate functional conformity in a web-based public-facility complaint system, supporting its use as an appropriate verification approach for SIPFAS, with performance summarized using the number and percentage of scenarios that satisfy their expected outputs rather than respondent-based usability or inferential statistical measures.

RESULTS AND DISCUSSION

Prototype Interface Implementation

The construction stage of the Rapid Application Development (RAD) process produced a web-based SIPFAS prototype designed to organize campus facility complaints through a structured and traceable information flow. The prototype integrates four principal interfaces, namely the public page, login page, user dashboard, and complaint-detail page, which collectively represent the main interaction sequence from accessing service information to monitoring an existing complaint. The use of a web-based complaint mechanism is consistent with previous studies showing that digital complaint systems can replace fragmented manual reporting and provide a more structured channel for complaint submission and monitoring (Alfawaz et al., 2022). The interface structure was developed around the functional requirements identified during requirement planning, while the construction process translated these requirements into interconnected visual components. Rather than emphasizing extensive administrative features, the prototype prioritizes information accessibility, complaint visibility, and continuity of status information across pages. This design orientation provides a

functional foundation for subsequent verification because each major interaction is represented through an identifiable interface component.

The public interface functions as the initial information gateway through which users can understand the purpose and basic operation of SIPFAS before authentication. Figure 2 presents the SIPFAS public page with the system identity, navigation options consisting of Home, Complaint Procedure, About the System, and Login, together with facility categories such as projectors, air conditioning/cooling systems, WiFi networks, and lighting. The inclusion of facility categories provides an early classification structure that can help users associate a reported problem with the appropriate facility type. A summary of active complaints is also presented to provide contextual information about the current complaint environment without requiring users to enter the authenticated area. This arrangement reflects the principle that complaint systems should provide an accessible and clearly defined channel through which users can identify where and how a problem should be reported (Hidhayah et al., 2024). The public page therefore operates not merely as a visual landing page but as an orientation layer that connects general service information with the subsequent authentication and reporting process.

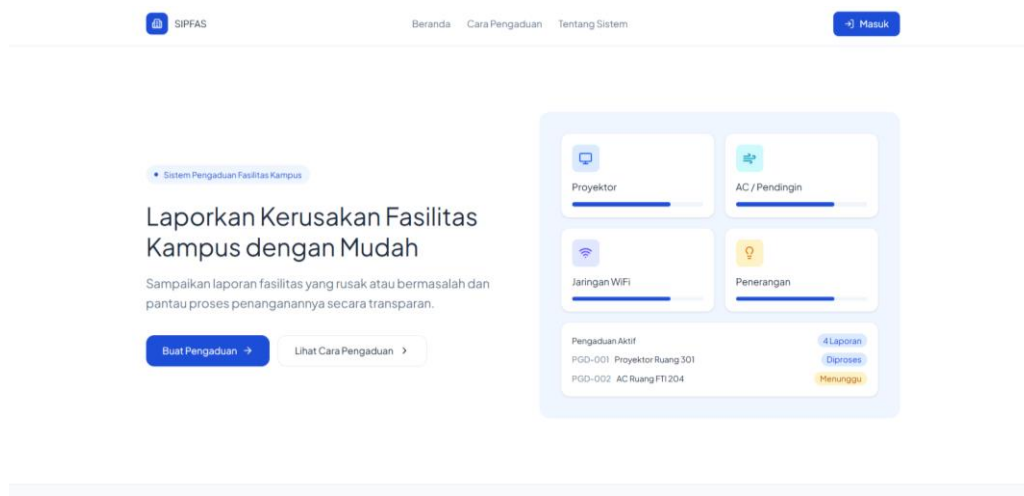


Figure 2. SIPFAS Public Page

The login interface provides the authentication gateway between the publicly accessible information and the personalized complaint-management functions of the system. As shown in Figure 3, the page adopts a two-column layout in which the left section communicates the purpose and function of SIPFAS, while the right section contains the email field, password field, Remember Me option, and Login button. This separation creates a visual distinction between explanatory information and the primary authentication action, allowing the user to identify the required interaction without navigating through unnecessary elements. A clear authentication interface is particularly relevant in complaint-management systems because subsequent information, including personal complaint history and report details, requires controlled access. Previous research on web-based complaint systems has demonstrated the importance of integrating user access with complaint submission and monitoring functions within a unified information environment (Sitanggang et al., 2024). The prototype, however, does not use the visual appearance of the login page as evidence of authentication security, since password protection, session management, encryption, and authorization must be verified through application and server-side testing.

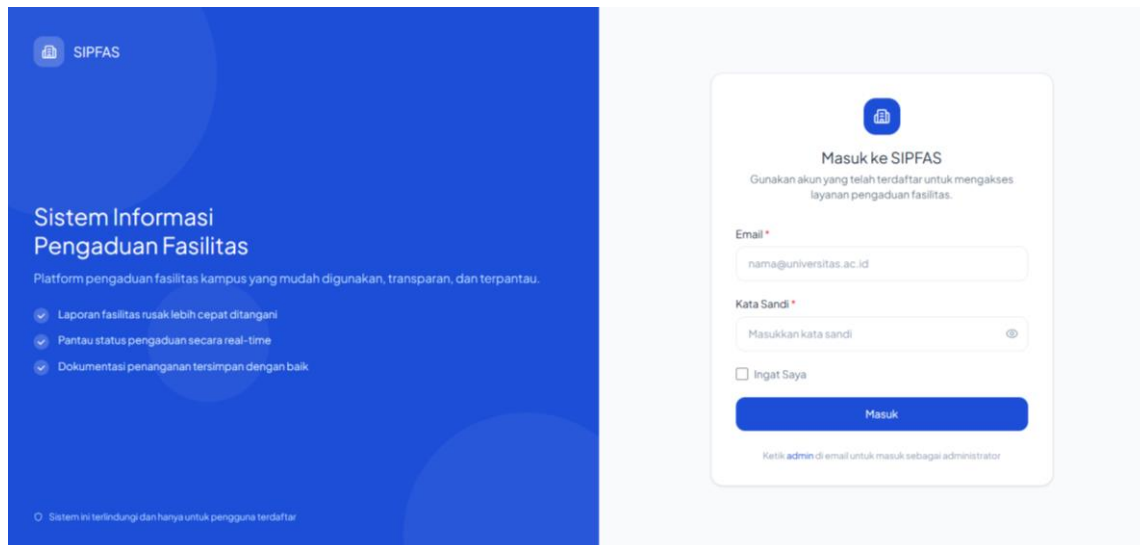


Figure 3. SIPFAS Login Page

Following authentication, the user dashboard consolidates complaint information into a compact monitoring interface that enables users to identify the current distribution of their reports without opening each record individually. Figure 4 shows four statistical cards representing Total Complaints, Waiting, In Process, and Completed, with displayed values of 12, 3, 4, and 5, respectively. The dashboard also provides a Create Complaint action and a Recent Complaints table containing the report number, facility, location, date, status, and detail action. The numerical summary establishes a direct relationship between aggregate complaint information and individual records, while the table provides a transition from overview information to detailed inspection. This structure corresponds to the functional orientation of complaint systems in which users require both complaint submission and status monitoring within the same digital environment (Phratama et al., 2024). The dashboard consequently serves as the central information-control point for users, reducing the need to repeatedly search for individual complaints when the immediate requirement is to determine the overall status of reported facility problems.

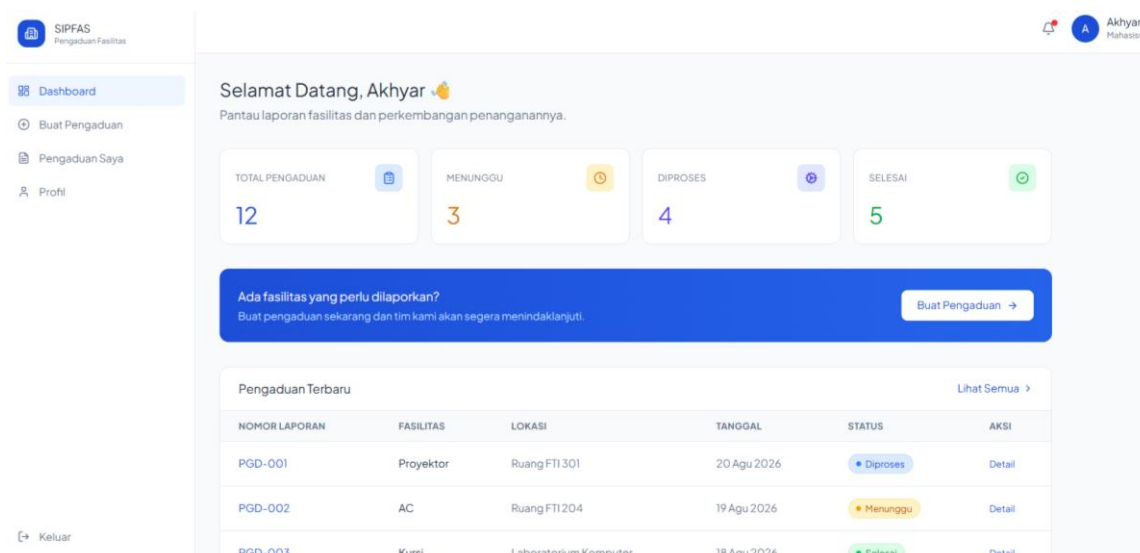


Figure 4. SIPFAS User Dashboard

The complaint-detail interface extends the dashboard information by presenting the complete record and chronological status history of a selected report. Figure 5 displays complaint PGD-002 with the title “AC Not Functioning,” category AC, location FTI Building Room 204, reporter Akhyar, complaint date of 19 August 2026, current status Waiting, and a description of the reported facility

problem. The page also provides a status timeline consisting of Complaint Submitted, Verified by Admin, Being Processed, and Completed, allowing the complaint to be interpreted as a process rather than as a static record. Such chronological representation strengthens traceability because users can associate a complaint with successive stages of administrative handling, an aspect also emphasized in web-based facility complaint systems developed for institutional environments (Nurhasan et al., 2022). The prototype therefore connects descriptive complaint data with process information, creating a more informative representation of facility-service activity. At the prototype stage, the timeline primarily demonstrates the intended information architecture, while the validity of its timestamps and state transitions requires separate verification.

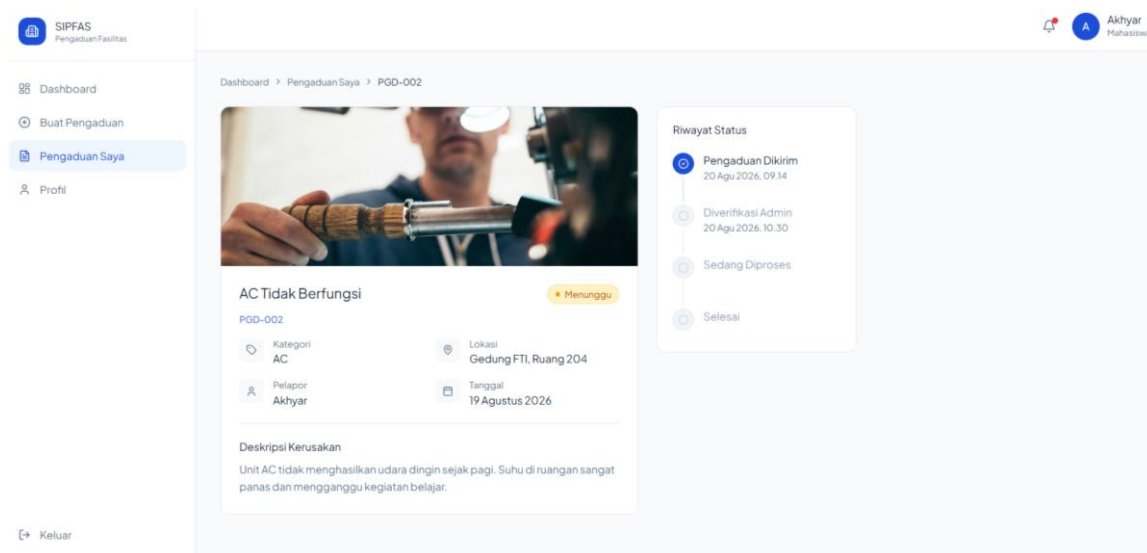


Figure 5. Complaint Detail and Status History for PGD-002

Taken as an integrated interface structure, the four prototype components establish a sequential interaction model consisting of public information access, authentication, complaint monitoring, and detailed status tracking. The progression from Figure 2 to Figure 5 indicates that SIFPAS does not treat complaint submission as an isolated transaction, but organizes it within a continuous information pathway in which users can move from general service information toward personalized complaint records. This approach is compatible with the iterative characteristics of RAD, where interface components can be progressively refined according to identified functional requirements and consistency findings during development (Sun & Sihombing, 2023). The resulting prototype remains intentionally focused on the core complaint workflow rather than incorporating advanced mechanisms such as automatic priority determination or Service Level Agreement management. Such prioritization of core interaction is relevant because interface complexity can increase the possibility of user errors during fundamental complaint activities, particularly authentication and complaint submission (Khansa et al., 2024). The implementation therefore provides a controlled prototype foundation for examining information consistency and functional behavior before broader system deployment.

Interface Information Consistency Verification

Interface information consistency was verified by comparing complaint attributes displayed across the public page, user dashboard, and complaint-detail interface, with particular attention to identifiers, status categories, locations, dates, and status-history information. This verification is important because a complaint-management system does not only need to display information, but also needs to preserve the meaning of the same data when users move between interconnected interfaces. The dashboard presented 12 total complaints consisting of 3 Waiting, 4 In Process, and 5 Completed reports, producing an arithmetic correspondence between the category values and the displayed total. The identity of complaint PGD-002 was also traceable from the complaint list to its detail page, allowing the selected record to be associated with its facility category, location, reporting date, and current status. This traceability reflects the information-monitoring function emphasized in previous campus

complaint systems, where users require access to complaint records and their handling progress through an integrated interface (Sitanggang et al., 2024). The verification therefore indicates that the prototype has established a basic information structure in which summary-level and record-level information can be connected.

The quantitative relationship between the dashboard summary and its status categories provides the clearest evidence of internal consistency within the prototype. The values displayed in the four statistical cards were examined by reconstructing the total from the three complaint-status categories rather than treating the dashboard representation as inherently correct. Table 3 summarizes the verification results and shows that the relationship between Waiting, In Process, and Completed complaints is consistent with the displayed total of 12, while PGD-002 can also be traced across the dashboard and detail interface. The consistency of these values is important because an aggregate dashboard becomes misleading when its statistical indicators do not correspond to the records represented in the system. A similar emphasis on monitoring and structured complaint information can be observed in the development of campus infrastructure reporting systems, where complaint records are organized to support monitoring activities (Nurhasan et al., 2022). The verification consequently confirms that the prototype's principal summary information has a coherent numerical relationship, although this finding does not yet establish the correctness of underlying database transactions.

Table 2. Interface Information Consistency Verification Results

Verification Object	Evidence on Interface	Status	Notes
Dashboard summary	Waiting 3 + In Process 4 + Completed 5 = Total 12	Consistent	Status counts correspond to the total
PGD-002 identity	PGD-002 appears on dashboard and detail page	Consistent	Complaint identity can be traced
Category/location/status	AC; FTI Room 204; Waiting	Relatively consistent	Detail specifies the location more completely: FTI Building, Room 204
Complaint date vs. Timeline	Detail: 19 August 2026; Complaint Submitted: 20 August 2026, 09:14	Inconsistent	Date definition should be clarified or data corrected
Administrator verification state	Timestamp 20 August 2026, 10:30 is available but indicator remains grey	Inconsistent	Visual state should follow the process history

The comparison in Table 2 identifies a second level of consistency in which the same complaint is represented with slightly different location descriptions across interfaces. PGD-002 is associated with the AC facility and FTI Room 204, while the detail page provides the more complete expression “FTI Building, Room 204,” meaning that the underlying location concept remains identifiable even though the textual representation is not completely standardized. This condition can be classified as relative rather than absolute inconsistency because the additional building information does not contradict the room designation. Standardized metadata labels would nevertheless improve information retrieval and reduce ambiguity when a campus contains multiple buildings with similarly numbered rooms. The importance of clear information presentation is consistent with findings from facility complaint interface research, which emphasizes the role of understandable interaction structures in helping users communicate facility problems accurately (Khansa et al., 2024). For SIPFAS, this finding suggests that interface consistency should be treated not only as a visual design issue but also as a data-representation issue connecting labels, metadata, and user interpretation.

A more substantive discrepancy was identified between the complaint date shown in the detail interface and the date recorded in the status timeline. The detail page identifies PGD-002 as having a complaint date of 19 August 2026, whereas the first timeline event, Complaint Submitted, is recorded as 20 August 2026 at 09:14. The one-day difference cannot be interpreted as an error with certainty because the two fields may represent different concepts, such as the date associated with the reported incident and the timestamp associated with the submission transaction. If both fields are intended to represent the same event, however, the difference constitutes a temporal inconsistency that may weaken

users' ability to reconstruct the actual complaint sequence. Systems for complaint processing require chronological information to remain sufficiently explicit because monitoring depends on users being able to determine when a report entered the service workflow (Phratama et al., 2024). The appropriate correction is therefore not necessarily to force the two values to become identical, but to define the semantic meaning of each date and represent that meaning through explicit labels.

The verification also detected an inconsistency between the available administrator-verification timestamp and its corresponding visual state in the timeline. The interface records an administrator verification event at 20 August 2026, 10:30, yet the associated timeline indicator remains visually grey, creating a discrepancy between the textual process history and the graphical representation of that state. This discrepancy is more consequential than a minor formatting difference because users generally interpret timeline indicators as direct representations of process completion. A status-monitoring system can lose informational value when textual metadata indicate that an event has occurred while visual state indicators suggest that the event remains inactive. Previous development of complaint systems has incorporated status monitoring as a central mechanism for communicating the progress of submitted reports (Yanuardi et al., 2024). The finding indicates that SIPFAS requires a state-rendering rule in which the visual status of each timeline stage is generated consistently from the corresponding process state or validated timestamp.

The consistency verification positions interface validation as an essential quality-control activity within the RAD development cycle rather than as a cosmetic assessment conducted after implementation. RAD emphasizes iterative refinement, making discrepancies identified during construction particularly valuable because they can be corrected before the prototype proceeds to broader functional implementation and testing (Hidayatulloh & Patyani, 2024). The SIPFAS findings demonstrate that numerical consistency can already be achieved at the dashboard level, while semantic and temporal consistency requires further refinement at the complaint-detail level. These distinctions are important because a prototype may appear visually coherent while still containing inconsistencies that affect interpretation of operational information. The present verification therefore establishes two concrete refinement requirements: standardization of complaint metadata and synchronization between timeline states and their underlying event information. The resulting prototype provides a stronger basis for subsequent Black Box Testing because the functions to be tested can be evaluated against an interface whose expected information states have been explicitly identified.

Functional Verification and Black Box Testing

Functional verification was formulated to examine whether the principal SIPFAS functions correspond to the requirements established during the requirement-planning and construction stages. The verification scope covers authentication, input validation, complaint submission, image upload, complaint retrieval, complaint-detail access, status transitions, complaint completion, and role-based access control. Black Box Testing was selected because the assessment focuses on observable system responses to defined inputs without requiring inspection of the internal program structure (Zidan et al., 2022). The approach provides a structured mechanism for examining whether each functional requirement has a corresponding and observable system response. This testing orientation is consistent with previous complaint-management system development in which functional conformity is examined through predefined input-output scenarios (Anisa et al., 2026). The resulting scenarios were used as the verification basis for the SIPFAS prototype, while the distinction between expected behavior and executed test results was maintained to prevent unsupported performance claims.

The authentication functions constitute the initial verification component because users must be appropriately identified before accessing personalized complaint information. The valid-login scenario is designed to direct an authenticated user to the dashboard, whereas an incorrect password should result in rejected access and an informative error message. Empty email or password fields are treated as separate validation conditions because incomplete inputs should be identified before authentication is processed. The expected responses for these conditions are defined together with the complaint-management scenarios in Table 4, which provides a reproducible mapping between test conditions and expected system behavior. The inclusion of authentication and validation scenarios reflects the importance of controlled access in web-based tertiary-institution complaint systems (Alfawaz et al., 2022). At the prototype stage, the presence of login components should not be

interpreted as evidence of security effectiveness because session management, password protection, encryption, and server-side authorization require execution-based verification.

Table 3. Black Box Testing Scenario Design

No.	Function	Scenario	Expected Output
1	Login	Valid account credentials are entered	User dashboard is displayed
2	Login	Incorrect password is entered	System rejects access and displays an error message
3	Login Validation	Email/password fields are left empty	Field validation message is displayed
4	Create Complaint	Complete complaint data are submitted	Complaint is stored and assigned an identity
5	Photo Upload	Valid image file is uploaded	Preview/file is accepted by the system
6	Complaint List	User opens My Complaints	User's complaint list is displayed
7	Complaint Detail	User opens PGD-002	Complaint details and status are displayed consistently
8	Change Status	Administrator changes Waiting to In Process	Status and status history are updated
9	Complete Complaint	Administrator changes In Process to Completed	Final status and timestamp are stored
10	Access Control	Regular user attempts to access administrator page	Access is denied according to authorization rules

The complaint-management scenarios extend verification from authentication toward the continuity of information generated through the reporting process. A complete complaint submission is expected to create an identifiable report, while a valid image file should be accepted as supporting evidence for the reported facility condition. The complaint-list function is expected to retrieve records belonging to the authenticated user, while the complaint-detail function should display the record corresponding to the selected identifier, particularly PGD-002 in the prototype. This relationship is essential because complaint information must remain traceable from initial submission to subsequent monitoring rather than being represented as disconnected transactions. Web-based student complaint systems similarly emphasize integrated complaint submission, monitoring, and access to complaint histories as core service functions (Phratama et al., 2024). The SIPFAS design adopts this principle through a relatively compact workflow that prioritizes complaint identity, record retrieval, and continuity of status information.

Status management constitutes the next functional dimension because the value of a complaint system depends partly on its ability to represent changes in the handling process. The transition from Waiting to In Process is expected to modify the current complaint state while adding or updating the corresponding status history. A subsequent transition from In Process to Completed should establish the final state and record its associated timestamp. The relationship between current state and historical events is particularly important for SIPFAS because the complaint-detail interface uses a timeline to communicate the progression of administrative handling. Digital complaint systems have previously employed status monitoring to improve the visibility of complaint-processing activities and service responses (Yanuardi et al., 2024). SIPFAS currently limits this mechanism to the fundamental status sequence rather than incorporating automated priority determination or Service Level Agreement calculations, which are features addressed in more advanced campus facility reporting systems (Widiatry et al., 2026).

Access control provides the final functional boundary in the verification design because ordinary users and administrators have different operational responsibilities. A regular user attempting to access an administrator page is expected to receive an access-denied response, while authorized administrators should be able to perform designated actions such as changing complaint status. This distinction is necessary because removing administrative navigation from the user interface does not

demonstrate that unauthorized requests are technically restricted. Role separation is particularly relevant to complaint-management applications because administrative functions involve modification of records and process states that should not be available to ordinary users (Fakhriyah & Kurniawan, 2024). The scenario in Table 4 therefore treats authorization as an observable functional requirement rather than merely a user-interface characteristic. Full verification should subsequently include both normal navigation and direct access attempts against restricted resources once the complete backend and authorization mechanisms are operational.

The Black Box Testing scenarios presented in Table 4 should be interpreted as a structured verification framework rather than as evidence of completed pass-or-fail testing. The available prototype evidence establishes the interface structure and expected functional behavior, but it does not provide documented execution records for each scenario that would justify a numerical success rate. This distinction is important because assigning successful outcomes to unexecuted test cases would introduce empirical claims that are not supported by the collected evidence. The iterative character of RAD allows these scenarios to be executed and refined as the system components progress from prototype construction toward functional implementation (Sun & Sihombing, 2023). Once the backend, database persistence, authentication, image handling, status management, and authorization mechanisms are operational, each scenario can be executed using controlled inputs and classified according to its predefined expected output. The resulting observations can then support calculation of successful and failed cases and provide an empirical basis for evaluating the functional conformity of the implemented SIPFAS system.

CONCLUSION

This study developed a web-based Campus Facility Complaint Information System (SIPFAS) prototype using the Rapid Application Development (RAD) approach through requirement planning, user design, construction, and cutover and verification stages. The resulting prototype provides an integrated interface structure comprising a public information page, login interface, user dashboard, and complaint-detail page with status history. The dashboard presents a concise overview of complaint conditions through the Waiting, In Process, and Completed categories, with 3, 4, and 5 complaints respectively, producing a consistent total of 12 complaints. The PGD-002 complaint can also be traced across the dashboard and detail interface through its complaint identity, category, location, date, and current status. These interface components establish a structured reporting and monitoring flow that can improve the visibility of facility complaints compared with fragmented reporting practices. The development process demonstrates that RAD can support iterative refinement of the SIPFAS interface and accommodate verification of information consistency during prototype development. The prototype therefore provides a functional design foundation for campus facility complaint management while remaining at an early stage of system implementation.

The interface verification identified two issues that require resolution before SIPFAS can proceed to full operational implementation. The first concerns a temporal inconsistency between the complaint-detail date of 19 August 2026 and the timeline entry indicating that the complaint was submitted on 20 August 2026 at 09:14. The second concerns the administrator-verification state, where a timestamp of 20 August 2026 at 10:30 is displayed while the corresponding visual indicator remains inactive. These findings indicate that information synchronization between complaint metadata and status-history representations remains an important technical requirement for maintaining reliable system information. The prototype should not yet be considered fully operational because functional execution of the Black Box Testing scenarios, backend data persistence, authentication and authorization mechanisms, file-upload processing, and usability evaluation have not been empirically completed. Future implementation should prioritize synchronization of date definitions and status states, followed by systematic execution of the predefined functional test scenarios across user and administrator roles. Subsequent evaluation involving actual campus users can further assess usability, interaction efficiency, and practical acceptance before deployment in a full institutional environment.

REFERENCES

Alfawaz, K. M., Anusiuba, R. C. O., Mani, P., & Bharathiraja, N. (2022). *Design and Implementation of an Online Tertiary Institution Web-Based Student Complaint Management System*. Shineeks Publishers.

- Anisa, N., Syarifuddin, S., & Mustafidah, H. (2026). Rancang bangun sistem informasi pengaduan fasilitas publik pada Dinas PUPR Kabupaten Bima menggunakan prototyping dan Black Box Testing. *Journal of Information System Research (JOSH)*, 7(4), 1325–1339. <https://doi.org/10.47065/josh.v7i4.10714>
- Aulia, R., Nugroho, B. A., & Hasibuan, N. A. (2025). Rancang bangun sistem informasi pengaduan mahasiswa berbasis web menggunakan metode DevOps. *Jurnal Rekayasa Sistem Informasi dan Teknologi*, 2(4), 1468–1483. <https://doi.org/10.70248/jrsit.v2i4.2417>
- Fakhriyah, M., & Kurniawan, R. (2024). Design and Development of A Web-Based Internet Complaint Information System using A Rule-Based Method at PT Integra Kreasitama Solusindo. *IJISTECH (International Journal of Information System and Technology)*, 8(2), 117–129. <https://doi.org/10.30645/ijistech.v8i2.355>
- Hidayah, R. N. L., Agitha, N., & Anjarwani, S. E. (2024). Sistem informasi pengaduan mahasiswa Fakultas Teknik Universitas Mataram berbasis website. *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTika)*, 6(2), 461–468. <https://doi.org/10.29303/jtika.v6i2.398>
- Hidayatulloh, S., & Patyani, E. (2024). Metode Rapid Application Development dalam pengembangan sistem informasi perpustakaan berbasis web. *Computer Science (CO-SCIENCE)*, 4(2), 109–118. <https://doi.org/10.31294/coscience.v4i2.3332>
- Khansa, S. S., Rahman, A. S., Ramadhan, D. N., & Huda, M. Q. (2024). UI/UX design for public facility complaint services using mobile-based design thinking methodology. *Journal of System and Computer Engineering*, 5(1), 128–140. <https://doi.org/10.61628/jsce.v5i1.1070>
- Maulana, D. N., Wirapraja, A., & Soedargo, D. S. O. (2026). The Design of Website-Based Mosque Information System: A Case Study of Al-Muhajirin Mosque In Balongsari Using The Method of Rapid Application Development. *International Journal of Science and Environment (IJSE)*, 6(1), 385–393. <https://doi.org/10.51601/ijse.v6i1.357>
- Nilawati, L., & Martin, M. (2023). Penerapan metode RAD pada perancangan sistem informasi permohonan data aduan Smartmaps berbasis web. *JURIKOM (Jurnal Riset Komputer)*, 10(2), 648–658. <https://doi.org/10.30865/jurikom.v10i2.6041>
- Nurhasan, U., Arianto, R., Vista, C. B., Lestari, E. D., & Sari, M. K. (2022). Implementasi aplikasi E-Reporting Infrastructure (ERI) sebagai media monitoring pengaduan kerusakan fasilitas kampus. *PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika*, 15(1), 76–90. <https://doi.org/10.33322/petir.v15i1.1532>
- Phratama, O. M., Handhayani, T., & Perdana, N. J. (2024). Student Feedback Systems: Developing a Web-Based Solution for Efficient Complaint Processing at Faculty of Information Technology Tarumanagara University. *Internet of Things and Artificial Intelligence Journal*, 4(4), 803–821. <https://doi.org/10.31763/iota.v4i4.834>
- Rachelita, B. A., & Istiyowati, L. S. (2026, June). Development of a Web-Based School Academic Information System Using the Rapid Application Development (RAD) Method. In *Perbanas International Seminar on Economics, Business, Management, Accounting and IT* (Vol. 4, pp. 780–792). <https://journal.perbanas.id/index.php/Proficient/article/view/1300>
- Razzaq, N. S., & Dewi, R. S. (2025). Design of a Web Based Workshop Management Information System Application Using Rapid Application Development Techniques. *Journal of Digital Business and Innovation Management*, 4(2). <https://doi.org/10.26740/jdbim.v4i2.71678>
- Saputra, J. A., & Sutopo, J. (2025). Development of a web-based e-complaint platform with websocket integration using the prototyping method. *Jurnal Ilmiah Komputasi*, 24(1), 7. <https://doi.org/10.53842/juki.v7i2.1832>
- Sitanggang, R. I. K., Jamaluddin, & Simamora, R. J. (2024). Perancangan sistem informasi pengaduan mahasiswa Universitas Methodist Indonesia berbasis web. *TAMIKA: Jurnal Tugas Akhir Manajemen Informatika & Komputerisasi Akuntansi*, 4(2), 136–142. [https://doi.org/10.46880/tamika.Vol4No2\(SEMNASTIK\).pp136-142](https://doi.org/10.46880/tamika.Vol4No2(SEMNASTIK).pp136-142)
- Suni, E. K., & Sihombing, D. J. C. (2023). Analysis and development of Sipemas: Community complaint applications using the Rapid Application Development method. *Jurnal Mantik*, 7(3), 2693–2702. <https://doi.org/10.35335/mantik.v7i3.4484>
- Wicaksono, R., Rakryan, R., & Faizah, N. M. (2025). Aplikasi pengaduan sarana dan prasarana: Studi kasus di SMK Bhayangkari Delog berbasis web dengan metode Rapid Application

- Development menggunakan Sublime Text dan MySQL. *Jurnal Ilmu Komputer dan Teknologi Informasi*, 2(1), 1–11. <https://doi.org/10.35870/jikti.v2i1.1335>
- Widiatry, Nyahun, J. A., Novareldanio, R., Supratman, M. F. A., Geraldly, D. A., & Sasongko, R. S. (2026). Perancangan sistem pelaporan kerusakan fasilitas kampus berbasis web dengan penentuan prioritas dan SLA otomatis. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 10(3), 5500–5510. <https://doi.org/10.36040/jati.v10i3.18380>
- Yanuardi, Y., Azhari, L., Sinlae, A. A. J., & Alexander, A. D. (2024). Development of public service complaint system using Rapid Application Development (RAD) methodology. *J-INTECH: Journal of Information and Technology*, 12(1), 36–48. <https://doi.org/10.32664/j-intech.v12i1.1201>
- Zidan, M., Nur'aini, S., Wibowo, N. C. H., & Ulinuha, M. A. (2022). Black Box Testing pada aplikasi Single Sign On (SSO) di Diskominfostandi menggunakan teknik equivalence partitions. *Walisono Journal of Information Technology*, 4(2), 127–137. <https://doi.org/10.21580/wjit.2022.4.2.12135>
- Zuliyanto, A. F., & Asriningtias, Y. (2025). Pengembangan aplikasi pengaduan dan monitoring fasilitas umum. *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, 14(3), 1780–1791. <https://doi.org/10.35889/jutisi.v14i3.3290>