Journal of Applied Informatics Research Vol 2 No 2 (2026) pp. 1~12

Journal of Applied Informatics Research
Fakultas Vokasi, Kampus Unesa 1, Ketintang, Surabaya, Indonesia
Website: https://journal.unesa.ac.id/index.php/jair/ | E-mail: jair@unesa.ac.id

Implementation of Role-Based Access Control in the Design and Development of the Surabaya City Police (Polrestabes Surabaya) Website

Dara Lembayung Kinasih[1], I Gde Agung Sri Sidhimantra[2]

[1,2]Manajemen Informatika, Universitas Negeri Surabaya, Indonesia
[1] dara.22050@mhs.unesa.ac.id, [2] igdesidhimantra@unesa.ac.id

ARTICLE INFORMATION

Article history:

Received October 28, 2026
Revised November 01, 2026
Accepted November 02, 2026

Keywords:

Role-Based Access Control;
Public Complaint System;
Web Application;
Rapid Application Development;
System Usability Scale

ABSTRACT

The digitalization of public services requires structured complaint processing and controlled access to sensitive internal functions. This study develops and evaluates a web-based information and public complaint system for Surabaya City Police (Polrestabes Surabaya) using Rapid Application Development (RAD), Laravel, and MySQL. Role-Based Access Control (RBAC) separates Administrator, Sub-administrator, and Officer functions, while assignment checks restrict Officers to their assigned complaints. IP allowlisting and operating-hour restrictions were implemented as additional access controls, and VPN access was evaluated through approved exit IPs. The evaluation comprised nine functional black-box tests, 12 authorization scenarios, 14 network, time, and VPN scenarios, and a controlled comparison of 37 internal role-route combinations. All test scenarios produced their expected outcomes. In the controlled comparison, all 37 combinations were accessible with RBAC disabled, whereas 19 were permitted and 18 were denied with RBAC enabled according to the defined access policies. Usability evaluation involving 50 personnel from Polrestabes Surabaya yielded a mean System Usability Scale (SUS) score of 82.70. These findings demonstrate the implementation of role-based and contextual access restrictions under the tested conditions, but do not establish comprehensive resistance to cyberattacks or operational effectiveness in a production deployment.

1. INTRODUCTION

Digital transformation has enabled public institutions to deliver information and services through integrated web platforms. Public complaint channels are especially important because citizens need accessible mechanisms to submit reports and monitor their handling. A study of Indonesian e-government portals identified considerable variation in the maturity of online complaint facilities [1]. Previous web-based complaint systems have also addressed the digitization of reporting and information access [2], [3]. Nevertheless, an online reporting interface must be accompanied by controls governing who may inspect, assign, and resolve reports.

Surabaya City Police (Polrestabes Surabaya) requires integrated dissemination of institutional information and a structured complaint-management process. The requirements analysis conducted for this study identified a need to consolidate information on services, news, personnel, and local police stations and to provide digital complaint submission, tracking codes, and status updates. The system developed here addresses those operational requirements as an application prototype evaluated with test data; its successful testing should not be interpreted as evidence of deployment across the police organization.

Complaint management involves distinct responsibilities. Administrators manage user accounts and website content, Sub-administrators verify reports and assign cases, and Officers investigate and document only their assigned complaints. Members of the public submit and track complaints without authenticating as internal users.

These different responsibilities make explicit authorization necessary, since a logged-in account should not automatically receive access to every internal route or complaint record.

Role-Based Access Control (RBAC) associates permissions with organizational roles rather than granting every permission directly to individual accounts [4], [5]. RBAC can help implement least-privilege access, although practical implementations must also consider application-specific constraints and object ownership [6]. In this study, route-level role checks are combined with assignment-aware authorization and rules governing permissible complaint-status transitions. Public information, submission, and tracking remain accessible without an internal login.

Previous studies on web-based complaint-management systems have primarily focused on digitizing complaint submissions, improving public service accessibility, and facilitating complaint documentation and status monitoring [2], [3], [7]. However, the present study extends this functional focus by implementing Role-Based Access Control (RBAC) to regulate access to internal system functions based on organizational responsibilities. Unlike a conventional complaint-management system that primarily provides reporting and tracking features, this study introduces three distinct internal roles: Administrator, Sub-administrator, and Officer, each with specific access permissions. Furthermore, the implementation combines route-level role authorization with complaint-assignment restrictions and status-transition validation. The contribution of this study is therefore an applied implementation and evaluation of multi-level authorization within a web-based complaint-management workflow, rather than the development of a new RBAC algorithm. The evaluation also compares 37 internal role-route combinations with RBAC disabled and enabled to demonstrate how the implemented authorization mechanism enforces role-specific access restrictions. Previous studies have also examined the validation of implemented RBAC policies in web applications and the testing of role-based authorization controls on websites [14], [15].

Network and time restrictions were implemented as complementary controls rather than as components of core RBAC. Access decisions can incorporate contextual attributes such as location and time [8], [10]; in the implemented system, requests to internal pages are checked against an allowed IP list and operating-hour rules. A VPN was tested as an access path using an allowed exit IP. These tests evaluate the specified configuration, not comprehensive resistance to network attacks.

The website was developed with Rapid Application Development (RAD), using requirements analysis, design, construction, and evaluation to support iterative refinement [9]. Laravel provides routing, middleware, and controller logic; MySQL stores complaint and content records; and Chart.js visualizes monthly complaint counts. The system covers public information, complaint submission and tracking, internal management, and authorization; it does not provide independent native mobile applications or advanced security functions such as penetration testing.

This study therefore asks how a role-based and context-restricted complaint-management website can be designed and tested and how usable the resulting interface is perceived to be. Evidence is drawn from functional tests, 12 access-control scenarios, controlled disabled/enabled RBAC route comparisons, 14 additional network/time/VPN scenarios, and a System Usability Scale (SUS) evaluation involving 50 respondents.

2. RESEARCH METHODS

The development process followed RAD, beginning with problem identification and data collection through observation, interviews, and literature review, followed by requirements analysis, UML-based design, implementation, and testing. The workflow is summarized in Fig. 1. The authorization evaluation used predefined roles and test scenarios to compare expected and observed access outcomes. For safety and reproducibility within the project scope, functional testing employed fictitious complaint data, as specified in the thesis.

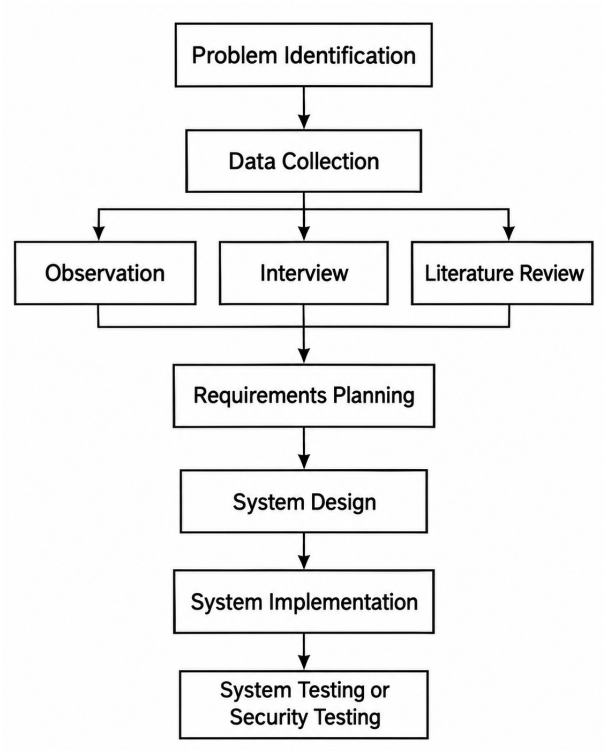


Figure 1. Research and development workflow

2.1 Problem Identification

Observation and interviews identified a need to consolidate public-service information and organize complaint submissions into records whose progress could be tracked. The intended users comprise citizens seeking information and internal staff responsible for managing website content and processing complaints. The requirements emphasize digital recording, case status visibility, and a division of internal responsibilities.

The security requirement was to prevent internal users from executing functions outside their authority. Content administration, complaint verification, case assignment, investigation, and closure require distinct access rules. In addition to authenticated role checks, an approved-IP rule and operating-hour rule were specified for internal endpoints.

Accordingly, the implementation and evaluation have two connected objectives: providing an integrated information and complaint website and enforcing permission boundaries at the application route, complaint record, and workflow-transition levels.

2.2 Data Collection

Data were collected through observation of information and complaint procedures, interviews with relevant police personnel regarding workflow and responsibilities, and a literature review of web-based public services, RAD, and access-control models. These activities informed user functions, complaint status flow, and internal access restrictions.

The requirements obtained from this process were mapped to four actors: Citizen, Administrator, Sub-administrator, and Officer. The first uses open public services; the other three authenticate to their assigned internal functions. The thesis does not document the number or demographic composition of interview participants; therefore, this article makes no claims about interview representativeness.

2.3 System Requirements

Functional requirements include public institutional information, complaint submission with supporting evidence, validation and tracking codes, report verification, Officer assignment, investigation updates, closure, content management, and monthly complaint statistics. Role-specific permissions and complaint ownership constraints are treated as essential functional requirements rather than optional interface features.

Non-functional requirements include responsive web presentation, usability, structured database storage, and controlled access. The security scope comprises authentication, RBAC, allowed-IP checking, and operating-hour restrictions. The study does not report independent penetration testing, comprehensive cryptographic assessment, or production-level availability testing.

2.4 System and Authorization Design

The design includes the research workflow, user interactions, a data model, complaint-processing rules, and an authorization pipeline. Figure 2 provides a simplified English overview of actors and capabilities, while Fig. 3 presents the implementation data model. Table 1 specifies the role-based permissions, and Table 2 describes the permitted complaint-status transitions and their associated constraints.

1. Use Case Diagram: The Use Case Diagram illustrates the interactions among four actors-the Public, Admin, Sub-Admin, and Officer-each possessing distinct functions and authorities within the Surabaya City Police (Polrestabes Surabaya) website. The Public can access public information, view complaint statistics, submit reports, and track the status of their complaints without needing to log in. The Admin is authorized to manage user accounts, update website content, monitor complaints, and view report statistics. Meanwhile, the Sub-Admin is responsible for verifying complaints, rejecting invalid reports, updating the status to "In Process," and assigning reports to Officers. Officers are tasked with handling assigned complaints, conducting investigations, documenting follow-up actions, uploading evidence of the resolution, and updating the report status to "Completed." All three internal user roles are required to authenticate before accessing system functions permitted by their respective authorities. This functional division serves as the foundation for implementing Role-Based Access Control (RBAC), ensuring that each user can only access features aligned with their assigned role.



Figure 2. Use Case Diagram

2. Class Diagram: A class diagram is used to illustrate the structure and relationships between classes that support data management for the Polrestabes Surabaya (Surabaya City Police) website. The design encompasses user data, informational content, public complaints, and the assignment of personnel. The 'User' class represents internal users-comprising Admins, Sub-Admins, and Officers-with access rights determined by a 'role' attribute. Meanwhile, the 'Complaint' class stores information regarding public reports, tracking numbers, complaint status, as well as the outcomes and evidence of follow-up actions. The relationships between these classes enable an integrated approach to authentication, report storage, status updates, and data access. This structure serves as the foundation for data management and the implementation of Role-Based Access Control (RBAC), ensuring that each user can perform functions in accordance with their authorized permissions.

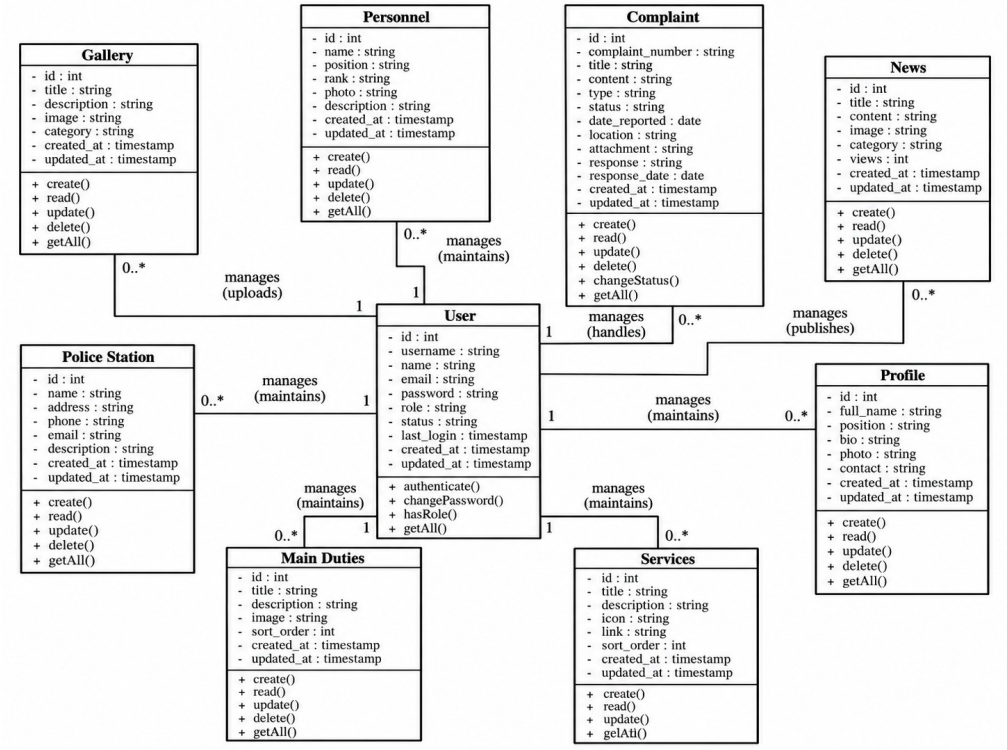


Figure 3. Class Diagram

3. Role-permission specification: Three authenticated internal roles are configured: Administrator, Sub-administrator, and Officer. Citizens interact through public endpoints without an internal account. Table 1 states the corresponding functions, and Table 2 sets out valid status transitions and their constraints.

Table 1. Role-Based Access Permissions

User role	Main permissions
Administrator	Manage internal accounts and website content; monitor complaints
Sub-administrator	Verify or reject complaints, set In Process status, and assign cases
Officer	Investigate assigned cases, upload evidence, and close assigned complaints
Citizen	Browse public content, submit complaints, and track reports

Status transitions are additionally checked against the current complaint state. A role may therefore be permitted to open a complaint page yet still be unable to move a record to an invalid state. These business-process checks are distinct from route-level RBAC.

Table 2. Permitted Complaint-Status Transitions

Initial status	Next status	Authorized role	Condition
Awaiting Verification	Rejected	Sub-administrator	Rejection decision; reason is optional in the documented workflow
Awaiting Verification	In Process	Sub-administrator	Change status; Officer assignment is performed through a separate function
In Process	Under Investigation	Officer	The complaint must be assigned to the acting Officer
Under Investigation	Completed	Officer	Follow-up outcome is required; supporting evidence may be uploaded

The permission model combines route checks with complaint-specific checks. For example, an Officer cannot inspect or update a complaint assigned to a different Officer even when the general route permits the Officer role.

4. Authorization pipeline: For internal requests, the internal.network middleware first checks the source IP. Authentication and the internal-user check then run before CheckRole evaluates route permissions. The internal.hours middleware applies time restrictions to Administrators and Sub-administrators; Officers are exempted to support after-hours field duties. Controller and ComplaintService rules subsequently check assignment ownership and allowed state transitions.
   Algorithm 1 describes the decision process in procedural form, while Fig. 4 shows its main decision points. A disallowed IP or an out-of-hours Administrator/Sub-administrator request produces HTTP 403; an unauthenticated request is redirected to sign-in, and a mismatched role is redirected to the corresponding dashboard according to the configuration evaluated.

Algorithm 1. Internal access decision procedure
Input: request R for an internal route and user u. Output: allow, deny, or redirect.
1. Check originating IP (internal.network); if disallowed, return 403.
2. Verify authentication and internal-user role (admin); if absent, redirect to login.
3. Apply CheckRole route policy; if role is not permitted, redirect to role dashboard.
4. Apply internal.hours to Administrators and Sub-administrators; reject out-of-hours requests (403). Officers are exempt.
5. For Officer actions on complaint p, confirm that p is assigned to u.
6. For status updates, verify the acting role and permitted prior state (Table 2).
7. If all applicable checks pass, continue to the requested controller operation.

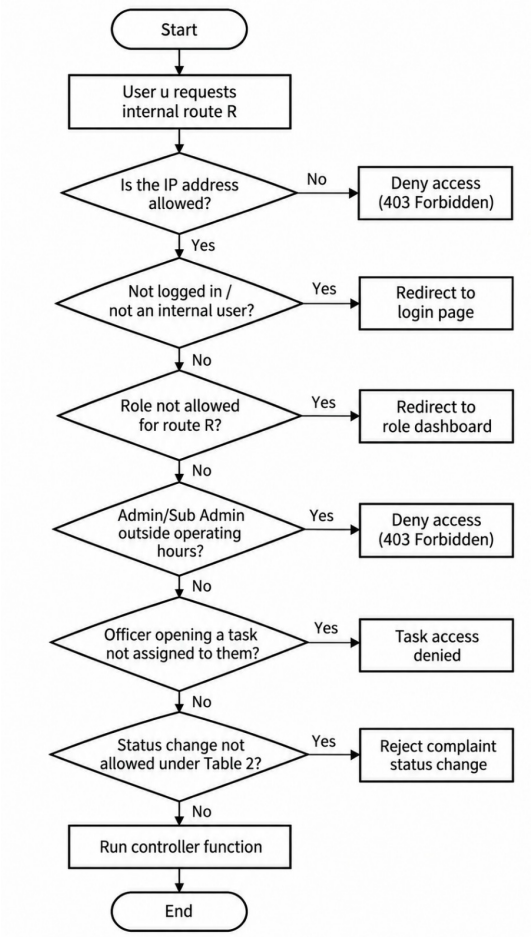


Figure 4. Flowchart of the Internal Page Access Verification Mechanism

5. Network and time restrictions: The allowed-IP policy acts as a contextual boundary for internal endpoints; public routes remain available from public networks. Operating-hour checks apply to Administrators and Sub-administrators, whereas Officers may access assigned duties outside those hours. VPN access was evaluated by verifying whether its exit IP appeared in the approved list.

These restrictions complement, but do not replace, role authorization. The study tests authorized versus unauthorized network conditions and operational schedules; it does not claim that an IP allowlist or VPN alone protects against all attack scenarios.

6. Monthly complaint statistics: Complaint timestamps are grouped by month and year, counted, and visualized using Chart.js. This is calendar-period aggregation (time-series grouping), not a predictive time-series model. Four test scenarios checked display, record counts, period grouping, and agreement between the plotted values and stored data.

2.5 System Implementation

The prototype was implemented with Laravel and MySQL. Public pages provide institutional information, service details, complaint submission, and complaint tracking. Internal dashboards offer account and content management, report verification and assignment, investigation updates, and monthly summaries. The User model stores role information; middleware and controllers enforce the role, network, time, ownership, and state-transition restrictions.

2.6 Evaluation Procedures

Functional black-box tests examined valid and invalid login inputs, complaint validation, evidence uploads, report storage, tracking-code generation, and status lookup. Access-control tests covered 12 role, network, time, and public-access scenarios, followed by 13 role-feature comparisons with RBAC disabled and enabled. A separate route-level comparison considered 14 internal GET routes across three internal roles, resulting in 42 possible role-route combinations. Five combinations were excluded because they had additional authorization checks at the controller level. These combinations were not included in the calculation because their access decisions were also affected by controller-level authorization and therefore could not be used to measure the direct effect of the RBAC middleware. Consequently, 37 comparable role-route combinations were evaluated under two conditions: RBAC disabled and RBAC enabled. Separate tests evaluated IP restrictions (four scenarios), operating hours (five scenarios), and VPN routing (five scenarios). Four additional tests examined monthly complaint statistics. Usability was measured using the ten-item System Usability Scale (SUS), completed by 50 personnel from Polrestabes Surabaya after they had tried the website. The standard odd- and even-item scoring adjustments and the 2.5 multiplier were applied [11], [12].

3. RESULTS AND DISCUSSION

3.1 Implemented Website

The implemented application integrates a public website with role-specific internal dashboards. The representative screens in Figs. 5-9 document the sign-in interface, public homepage, complaint-submission page, administrator dashboard, and monthly complaint statistics. The screens show the actual Indonesian-language interface used in the project; the academic descriptions and captions are presented in English.

1. Internal sign-in: Registered users submit credentials through the login page and, after authentication, are directed to the appropriate role dashboard. Subsequent route checks determine which features are available rather than relying solely on menus hidden from view.

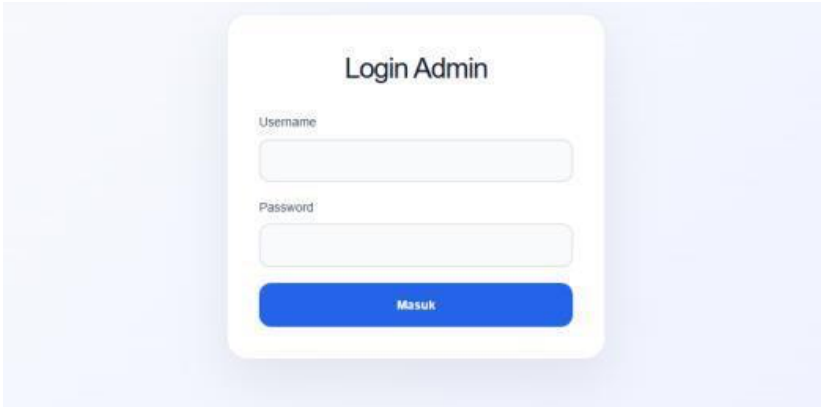


Figure 5. Internal user login page

2. Public homepage: The homepage presents information about police services, institutional activities, and navigation to the complaint service. It is accessible without internal authentication.

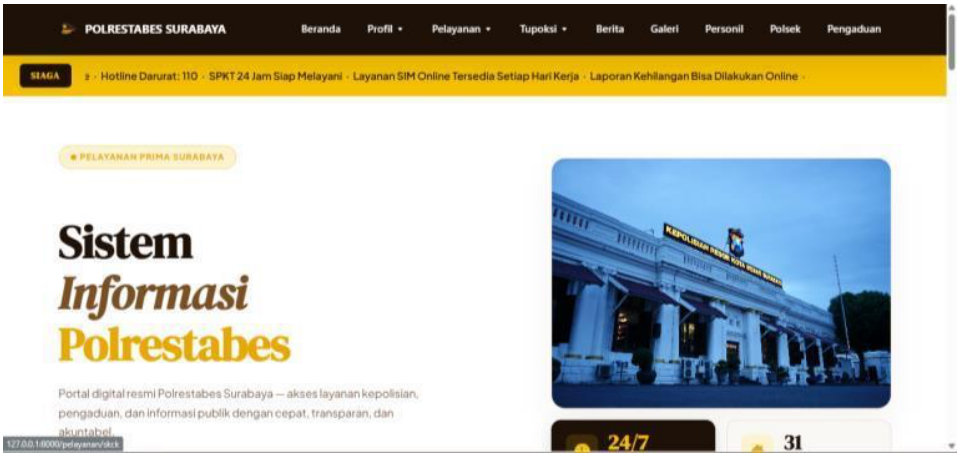


Figure 6. Public homepage

3. Complaint submission and tracking: A citizen enters complaint details and uploads supporting evidence. A valid submission receives the initial Awaiting Verification status and a tracking code, allowing the citizen to monitor progress without an internal account.

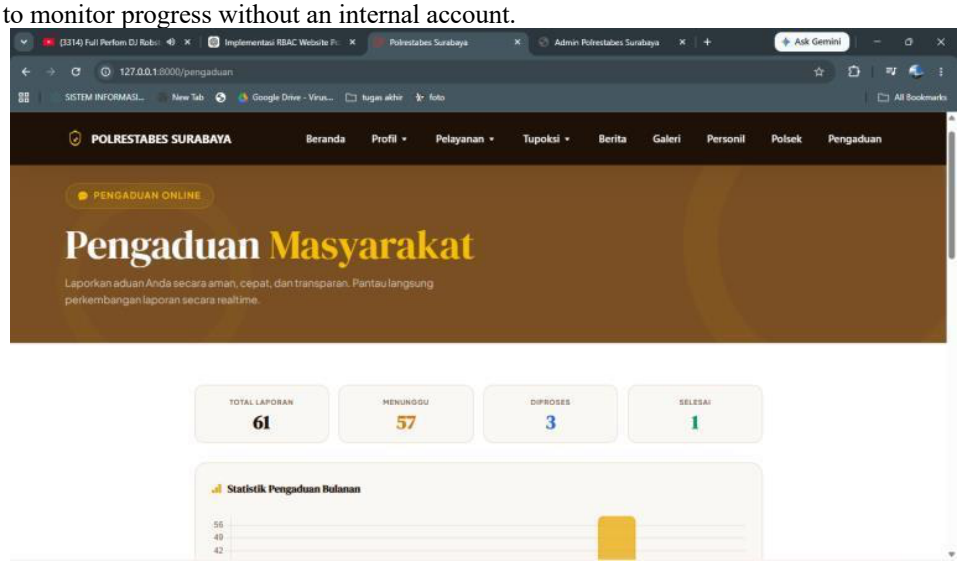


Figure 7. Public complaint submission page

4. Administration dashboard: The dashboard provides summaries and navigation for authorized content and user-management tasks. Public information and complaint records are displayed and managed within each role's configured permissions.

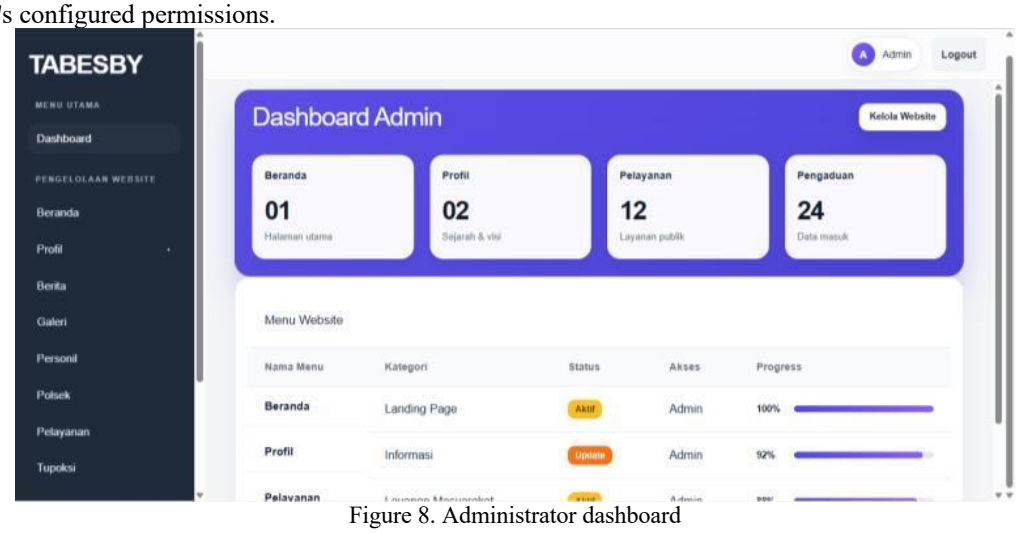


Figure 8. Administrator dashboard

5. Monthly complaint chart: Complaint counts are aggregated by month and displayed with Chart.js. The descriptive chart supports monitoring; it does not forecast future complaint volumes.

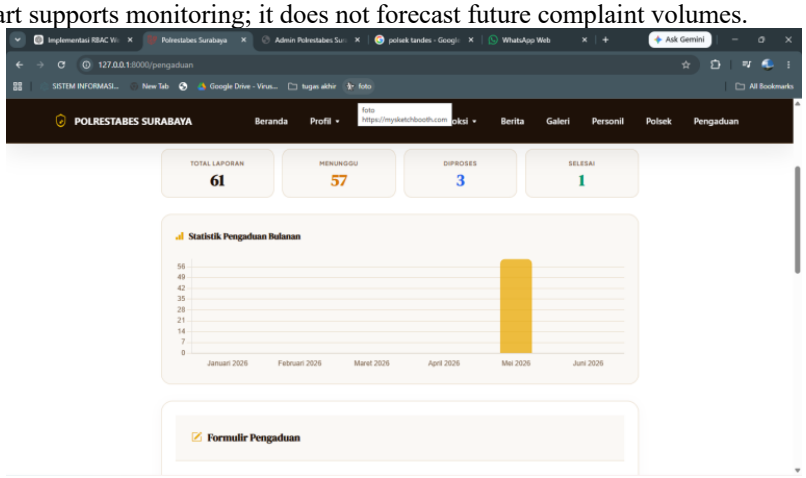


Figure 9. Complaint statistics by month

3.2 Role-Based Authorization Implementation

The application uses the User.role attribute and Laravel CheckRole middleware to restrict internal routes. The Administrator manages users and content; the Sub-administrator handles verification, rejection, initial status changes, and assignments; and the Officer investigates assigned reports and records closure. Citizens use public functionality without an internal account. This design follows the role-to-permission mapping proposed in classical RBAC models [4], [5].

Assignment-based checking is enforced in controller/service logic. This is a material distinction: route-level RBAC can authorize an Officer to open a case-management endpoint, but case-level logic must still verify whether the requested complaint was assigned to that Officer. Likewise, the status transition rules in Table 2 prevent authorized users from skipping workflow steps. Contextual IP and operating-hour checks remain separate from these role and ownership decisions.

Table 3. Implemented Access-Control Components

Component	Implementation
User role	Role attribute on User model (Administrator, Sub-administrator, Officer)
Middleware	CheckRole verifies authorization for requested routes
Route permissions	Account administration, verification, and Officer tasks restricted by role
Complaint authorization	Controller checks whether a complaint is assigned to the Officer
Status transitions	Controller rejects transitions that violate role or prior-state rules
Network and time (outside RBAC)	Approved-IP policy and operating-hour checks

The resulting implementation expresses least privilege at multiple levels: role permissions limit feature access, assignment checks limit individual complaint records, and transition rules limit actions on records. These mechanisms were evaluated using fixed functional scenarios, rather than inferred from screen visibility alone.

3.3 Functional and Access-Control Tests

The test plan assessed public-facing forms, role restrictions, direct route access, network and time policies, and the monthly statistics feature. Test cases used fictitious complaint records, as stated in the thesis; therefore, the results describe functional and authorization behavior under the tested conditions, not live police operational performance.

1. Black-box testing: Four login cases examined valid credentials, incorrect passwords, unregistered usernames, and blank fields. Five complaint-related cases covered data validation, evidence uploads, record storage, tracking-code generation, and status lookup. All nine tested cases produced their expected outcomes, as summarized in Table 4.

Table 4. Black-Box Functional Testing Summary

Feature	Scenarios	Passed
Login	4	4
Complaint submission and tracking	5	5
Total	9	9

2. Role and middleware tests: All 12 prescribed access scenarios returned the expected outcomes, including permitted and cross-role requests, direct URL access, allowed and disallowed networks, operating-hour limits, and public access without sign-in. In a separate 13-feature comparison, all requests were permitted when RBAC was disabled. After RBAC was enabled, six requests were permitted and seven were denied in accordance with the assigned permissions. This feature-level comparison is distinct from the route-level comparison described below. These results represent deterministic authorization tests rather than a security-attack detection benchmark.
   The controlled route-level comparison evaluated 14 internal GET routes across three internal roles, resulting in 42 theoretical role-route combinations. Five combinations were excluded because they had additional authorization checks at the controller level. These combinations were not included in the calculation because their access decisions were also affected by controller-level authorization and therefore could not be used to measure the direct effect of the RBAC middleware. After excluding these five combinations, 37 comparable role-route combinations remained for the controlled comparison of RBAC-disabled and RBAC-enabled conditions. Table 5 presents the results for each internal role.

Table 5. Internal Route Access with RBAC Disabled and Enabled

Role	Compared	RBAC off: allowed	RBAC on: allowed	RBAC on: denied
Administrator	12	12	10	2
Sub-administrator	12	12	5	7
Officer	13	13	4	9
Total	37	37	19	18

   With RBAC disabled, all 37 comparable role-route combinations were accessible. After RBAC was enabled, 19 combinations (51.35%) were permitted and 18 (48.65%) were denied according to the defined access policies. By role, the number of permitted routes was 10 out of 12 for Administrators, five out of 12 for Sub-administrators, and four out of 13 for Officers. These results demonstrate that the middleware enforced different access decisions according to the configured roles. The 48.65% figure represents the proportion of combinations denied under the tested policy, not a percentage improvement in overall cybersecurity. The findings also indicate the importance of server-side route authorization, since hiding administrative menus alone would not prevent users from directly requesting restricted internal routes [14], [15].

3. IP, time, and VPN tests: Requests from approved IP addresses were allowed, while disallowed internal requests returned HTTP 403. Public pages remained accessible from other networks. Operating-hour restrictions were enforced for Administrators and Sub-administrators, with Officers exempted for assigned duties outside office hours.
   When the VPN connection used an approved exit IP, internal sign-in and dashboard access followed the configured access policy. Connections that did not use an approved IP were denied. All 14 contextual security test cases produced their expected outcomes: four IP restriction scenarios, five operating-hour scenarios, and five VPN access scenarios, as summarized in Table 6. Because these tests depend on the configured network and access conditions, their results should not be generalized as evidence of comprehensive network security or resistance to penetration attacks [8], [10].

Table 6. Additional network, time, and VPN test results

Security control	Scenarios	Passed
IP restriction	4	4
Operating hours	5	5
VPN access	5	5
Total	14	14

4. Monthly statistics tests: All four test cases passed. The chart was displayed, monthly counts matched the database, grouping used the correct month, and plotted outputs matched the aggregated values. These tests support correctness of the implemented monthly reporting function under the tested data.

3.4 System Usability Scale Results

The usability evaluation involved 50 personnel from Polrestabes Surabaya who had tried the website before completing the ten-item System Usability Scale (SUS) questionnaire. Each response to an odd-numbered item was adjusted by subtracting one, while each response to an even-numbered item was calculated by subtracting the response value from five. The adjusted scores were summed and multiplied by 2.5 to obtain the SUS score for each respondent [11]. The cumulative SUS score was 4,135, resulting in a mean score of 82.70 across the 50 respondents.

Table 7. SUS evaluation results

Measure	Result
Participants	50
SUS questionnaire items	10
Sum of participant scores	4,135
Mean SUS score	82.70
Comparative benchmark	68 (approx.)
Interpretation	Favorable perceived usability

The mean score of 82.70 exceeds the commonly used comparative SUS benchmark of approximately 68 and indicates a favorable overall perception of the website's usability [12], [13]. The respondents were personnel from Polrestabes Surabaya. However, the available data do not specify the number of respondents in each internal user role, namely Administrator, Sub-administrator, and Officer. Therefore, the SUS result represents an overall usability assessment among Polrestabes Surabaya personnel and cannot be interpreted as a separate evaluation of each role-specific interface. Future evaluations should record respondent characteristics and role distributions so that the usability of each internal user interface can be assessed separately.

4. CONCLUSION

A web-based public information and complaint-management system for Polrestabes Surabaya was developed using the Rapid Application Development (RAD) method with Laravel and MySQL. The system integrates public information and complaint services with internal functions governed by role-based route permissions, complaint-assignment authorization, workflow-status restrictions, and additional IP and operating-hour controls. Public users can submit and track complaints without an internal account, while internal responsibilities are separated among Administrators, Sub-administrators, and Officers.

The evaluation results show that all nine black-box functional tests, 12 access-control scenarios, and 14 network, time, and VPN scenarios produced their expected outcomes. In the controlled comparison of 37 comparable internal role-route combinations, all combinations were accessible when RBAC was disabled, whereas 19 were permitted and 18 were denied when RBAC was enabled. Five combinations from the initial 42 were excluded because they had additional controller-level authorization checks and were therefore not suitable for measuring the direct effect of the RBAC middleware. The usability evaluation involving 50 personnel from Polrestabes Surabaya yielded a mean SUS score of 82.70, indicating favorable overall perceived usability. These findings demonstrate the implementation and scenario-level evaluation of role-based and contextual access restrictions under the tested conditions. However, they do not establish comprehensive resistance to cyberattacks or operational effectiveness in a production deployment.

Further evaluation should include independent penetration testing, secure reverse-proxy and client-IP handling, login rate limiting, comprehensive authorization and audit logging, and testing with operational stakeholders under an approved data-governance procedure. Role-specific usability studies and security assessments across different deployment environments would also strengthen the generalizability of the findings. The results should therefore be interpreted within the scope of the prototype, fictitious test data, and tested configuration.

ACKNOWLEDGEMENTS

The authors thank the D4 Informatics Management Study Program, Faculty of Vocational Studies, Universitas Negeri Surabaya, for facilities and support during the project.

REFERENCES

[1] N. P. Yunita, R. S. Selina, and G. Anggriawan, "A portrait of e-government implementation in public complaints services in Indonesia," The Indonesian Journal of Computer Science, vol. 13, no. 4, 2024. https://doi.org/10.33022/ijcs.v13i4.4184

[2] A. Erlangga, A. Hadi, and M. Syahputra, "Sistem informasi layanan pengaduan masyarakat berbasis web dalam peningkatan pelayanan publik di Kantor Dinas Sosial Kabupaten Pasaman Barat," Jurnal Sains dan Teknologi Informatika, vol. 1, no. 1, pp. 59-68, 2023. https://doi.org/10.38204/jsti.v1i1.1409

[3] S. Z. Ainiah, M. T. A. Bangsa, and F. Felawati, "Perancangan sistem informasi pengaduan masyarakat berbasis web pada Dinas Pekerjaan Umum dan Penataan Ruang Tanjung Jabung Timur," Jurnal Komputer Teknologi Informasi Sistem Komputer, vol. 4, no. 2, pp. 1310-1317, 2025. https://doi.org/10.62712/juktisi.v4i2.616

[4] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," Computer, vol. 29, no. 2, pp. 38-47, 1996. https://doi.org/10.1109/2.485845

[5] D. F. Ferraiolo, R. S. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," ACM Transactions on Information and System Security, vol. 4, no. 3, pp. 224-274, 2001. https://doi.org/10.1145/501978.501980

[6] M. U. Aftab et al., "Traditional and hybrid access control models: A detailed survey," Security and Communication Networks, vol. 2022, article 1560885, 2022. https://doi.org/10.1155/2022/1560885

[7] A. B. Nasution, P. Safitri, and N. Adittra, "Rancang bangun sistem pengaduan masyarakat berbasis web di Kesbangpol Tebing Tinggi," SATESI: Jurnal Sains Teknologi dan Sistem Informasi, vol. 5, no. 1, pp. 99-104, 2025. https://doi.org/10.54259/satesi.v5i1.4143

[8] A. Kumar, K. Abhishek, C. Chakraborty, and J. J. P. C. Rodrigues, "Real geo-time-based secured access computation model for e-Health systems," Computational Intelligence, vol. 39, no. 1, pp. 18-35, 2023. https://doi.org/10.1111/coin.12523

[9] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York: McGraw-Hill, 2020.

[10] V. C. Hu et al., Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST SP 800-162, updated 2019. https://doi.org/10.6028/NIST.SP.800-162

[11] J. Brooke, "SUS: A quick and dirty usability scale," in Usability Evaluation in Industry, P. W. Jordan et al., Eds. London: Taylor & Francis, 1996, pp. 189-194.

[12] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the System Usability Scale," International Journal of Human-Computer Interaction, vol. 24, no. 6, pp. 574-594, 2008. https://doi.org/10.1080/10447310802205776

[13] J. Sauro and J. R. Lewis, Quantifying the User Experience: Practical Statistics for User Research. Morgan Kaufmann, 2012. https://doi.org/10.1016/C2010-0-65192-3

[14] H. T. Le, L. K. Shar, D. Bianculli, L. C. Briand, and C. D. Nguyen, "Automated reverse engineering of role-based access control policies of web applications," Journal of Systems and Software, vol. 184, art. 111109, 2022. https://doi.org/10.1016/j.jss.2021.111109

[15] V. Sharma, "Testing role-based authorization controls in websites," Information Security Journal: A Global Perspective, vol. 18, no. 4, pp. 170-177, 2009. https://doi.org/10.1080/19393550903074127

AUTHOR BIOGRAPHY

	**Dara Lembayung Kinasih** was born in Malang on July 3, 2003. She graduated from SMA Negeri 1 Singosari and studied in the Applied Bachelor Program in Informatics Management, Faculty of Vocational Studies, Universitas Negeri Surabaya. Email: dara.22050@mhs.unesa.ac.id
	**I Gde Agung Sri Sidhimantra** is a lecturer in the Applied Bachelor Program in Informatics Management, Faculty of Vocational Studies, Universitas Negeri Surabaya, with the academic rank of Assistant Professor (Asisten Ahli). His areas of interest include web and mobile application development, decision-support systems, large language models, and agentic AI. Email: igdesidhimantra@unesa.ac.id