Jurnal Mandiri IT
Vol. 14 No. 1, July (2025), pp. 119-128

ISSN 2301-8984 (Print), 2809-1884 (Online)
Published by Institute of Computer Science

Implementation of role-based access control, multi tenancy and audit logging in a single sign-on system

Putranta Aswintama[1], Eri Haryanto[2], Ryan Ari Setyawan[3]
[1,2,3]Department of Informatics, Faculty of Engineering, Universitas Janabadra, Yogyakarta, Indonesia

ARTICLE INFO

Article history:

Received Jul 1, 2025
Revised Jul 10, 2025
Accepted Jul 19, 2025

Keywords:

Audit Logging;
JWT;
Laravel;
Role-Based Access Control;
Single Sign-On.

ABSTRACT

As enterprises increasingly require centralized, secure, and efficient authentication mechanisms, Single Sign-On (SSO) has emerged as a strategic approach to managing user access. This study discusses the implementation of an SSO system based on Laravel Livewire with support from JSON Web Token (JWT) and OAuth, developed for PT Radiator Springs Indonesia. The system integrates three main components: Role-Based Access Control (RBAC) for access rights management, a Multi-Tenancy architecture for separating users across organizational units, and Audit Logging to track user activities. The analysis shows significant improvements in security, with 87.5% fewer unauthorized access attempts and enhanced user management efficiency, evidenced by a 71.43% reduction in time to onboard new users. Additionally, the system generates over 300+ audit log entries per day, improving monitoring and compliance capabilities.

Corresponding Author:

Putranta Aswintama,
Department of Informatics, Faculty of Engineering,
Universitas Janabadra,
Jl. Tentara Rakyat Mataram No. 55-57, Yogyakarta, 55231, Indonesia
Email: putrantaaswintama21@gmail.com

1. INTRODUCTION

Information system digitalization in enterprise environments demands efficient and secure access control mechanisms. Single Sign-On (SSO) serves as a centralized solution allowing users to access multiple applications with a single authentication session. However, SSO systems face challenges in maintaining secure access, supporting scalability across departments, and providing user activity visibility (MARTIN et al., 2022).

In today's rapidly evolving digital landscape, organizations face increasing demands for secure, efficient, and scalable authentication mechanisms. As enterprises expand their use of diverse applications and cloud services, managing user identities and access permissions becomes a critical challenge. Single Sign-On (SSO) systems have emerged as a fundamental solution, enabling users to authenticate once and gain seamless access to multiple interconnected applications (Fugkeaw et al., 2024). This approach not only enhances user convenience but also strengthens security by centralizing authentication control.

The adoption of cloud computing, mobile workforces, and remote access technologies has accelerated the reliance on SSO solutions. Modern enterprises require authentication systems that are flexible, interoperable, and compliant with stringent regulations such as GDPR, HIPAA, and PCI-DSS. In response, the market has seen significant innovations including federated identity management, OAuth 2.0, and OpenID Connect protocols, which allow seamless integration across platforms and identity providers

Despite these advantages, deploying SSO in complex enterprise environments introduces significant security and operational challenges (Vasudevan, 2023). Chief among these are ensuring fine-grained access control, maintaining logical separation of users across multiple organizational units or tenants, and enabling comprehensive monitoring and auditing of user activities. Without addressing these concerns, enterprises risk unauthorized access, data breaches, and compliance violations, which can lead to severe financial and reputational damage (Farhadighalati et al., 2025).

The evolving threat landscape has further complicated SSO deployment. Cyberattacks are increasingly sophisticated, exploiting weak access controls and insufficient monitoring to gain unauthorized privileges. Furthermore, the growth of multi-tenant SaaS environments increases the risk of data leakage across organizational boundaries if tenant isolation is not rigorously enforced (Adewale, 2024). Consequently, modern SSO implementations must incorporate advanced security models such as Role-Based Access Control (RBAC), multi-tenancy architectures, and robust audit logging to mitigate these risks effectively (Chatterjee, 2022).

An organizational model that supports multiple tenants (such as branches or departments) within a single system instance while ensuring strict data and access isolation. Implementing multi-tenancy is essential for enterprises with distributed operations, enabling scalable user management and data segregation (Olabanji et al., 2023). The continuous recording of user actions and system events to provide visibility, traceability, and accountability. Audit logs are crucial for detecting suspicious behavior, investigating incidents, and demonstrating compliance with regulatory requirements (Yu et al., 2021).

This research is grounded in a real-world application at PT Radiator Springs Indonesia, a company characterized by diverse operational units and extensive internal systems. The implemented SSO solution leverages Laravel Livewire for a dynamic user interface, JSON Web Token (JWT) and OAuth protocols for secure authentication and authorization, and enforces multi-tenancy through the use of an organizational identifier. Audit logging is integrated to provide thorough visibility of user activities (PYROH et al., 2025).

By examining this implementation, the study aims to demonstrate how the integration of RBAC, multi-tenancy, and audit logging within an SSO framework can significantly improve security and operational efficiency in an enterprise context (Adewale, 2024). The findings are expected to provide valuable insights and best practices for organizations seeking to strengthen their authentication infrastructure and regulatory demands (AlMaqousi, 2024).

This research presents a technical solution to secure access and manage user identities in complex environments. The proposed Single Sign-On (SSO) system includes key security features like Role-Based Access Control (RBAC), multi-tenancy, and audit logging, aimed at improving operational efficiency and security within enterprises. While the system is adaptable to various sectors, it does not specifically address the regulatory needs of industries like finance or healthcare, nor does it focus on national cybersecurity regulations such as GDPR, HIPAA, or PCI-DSS (Owen et al., 2022). Instead, it serves as a flexible foundation that can be further customized to meet the specific security needs of these sectors.

2. RESEARCH METHOD

This study employs an Agile development methodology, utilizing the Scrum framework to guide the incremental design, implementation, and evaluation of the Single Sign-On (SSO) system integrated with Role-Based Access Control (RBAC), multi-tenancy, and audit logging for PT Radiator Springs Indonesia.

Development Approach

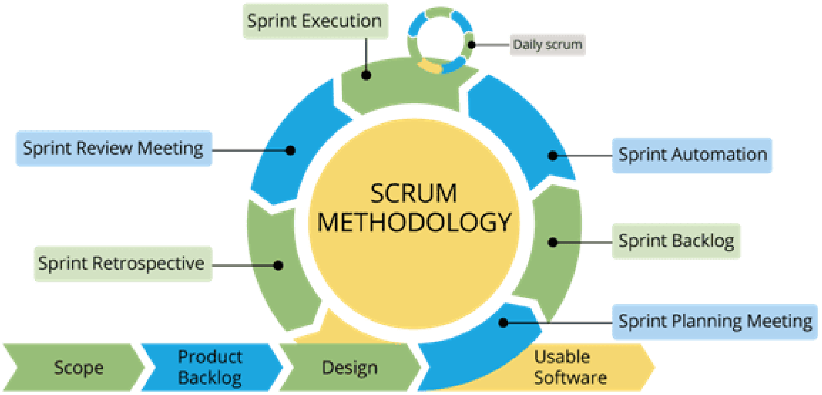


Figure 1. Agile Scrum methodology

The Agile Scrum approach was selected to accommodate the dynamic and evolving requirements inherent in enterprise identity management systems (Nyembe et al., 2023). Scrum's iterative cycles allow for early and continuous delivery of functional components, enabling rapid feedback from stakeholders and adaptive refinement. This method is especially suitable for complex systems where security, compliance, and scalability must be balanced carefully (Akhtar et al., n.d.).

By organizing the development into discrete sprints, each focusing on core features or integration points, the system was built in a modular way (Hron & Obwegeser, 2022). This enables quick adaptation to new security threats, changes in user role requirements, or increases in user load as the system scales. Each sprint provides the opportunity to address emerging security concerns, such as the introduction of new protocols or threat vectors, and also to optimize for scalability as more users, tenants, or organizational units are added.

The regular sprint reviews and retrospectives ensured that the team remained aligned with organizational needs and could address any evolving security or scalability challenges promptly. These reviews also provided a platform for collaborative problem-solving, ensuring that any adjustments needed to enhance security or meet scalability requirements were implemented in the following sprint. In this way, the Agile Scrum methodology ensured that the system could continuously evolve to meet the ever-changing demands of both security and scalability, aligning with the needs of the organization and the threat landscape.

System Architecture

The system backend is built on the Laravel framework, leveraging its robust MVC architecture and ecosystem. MySQL as the relational database management system, providing a reliable and scalable foundation for structured data storage and integrity. JSON Web Token (JWT) is used for secure, stateless authentication, while OAuth 2.0 protocol manages delegated authorization and identity federation with external identity providers (Zhang & Pan, 2022).

Laravel Livewire was employed to develop a reactive, single-page application (SPA) interface that supports dynamic user interactions without excessive client-side JavaScript, ensuring smooth user experiences (Yussuff et al., 2024).

Context Diagram

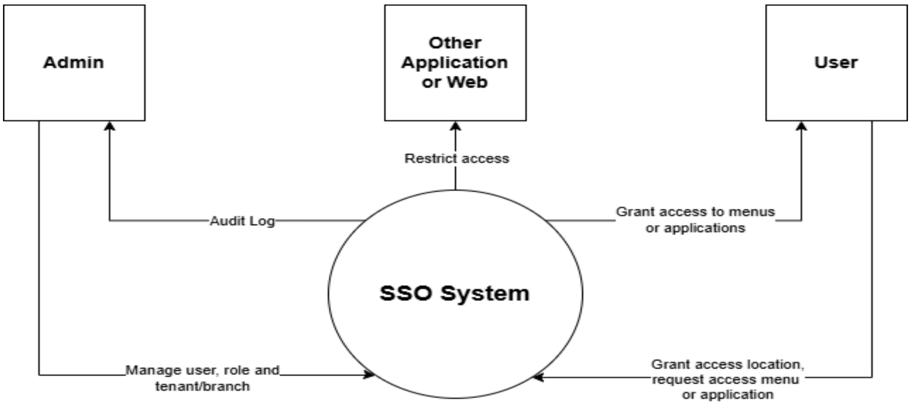

Admin
Other Application or Web
User
Restrict access
Audit Log
Grant access to menus or applications
SSO System
Manage user, role and tenant/branch
Grant access location, request access menu or application

Figure 2. Context diagram SSO system

The context diagram illustrates the interaction between the Single Sign-On (SSO) system and three main external entities, Admin, User, and External Applications. The Admin manages users, roles, and tenant or branch data, while all administrative activities are recorded through an audit logging mechanism. Users can request access to specific menus or applications, which are granted based on their assigned roles and tenant context. The SSO System also acts as a central authorization provider for external applications, restricting or allowing access through secure token-based validation. This architecture supports multi-tenancy, centralized authentication, role-based access control (RBAC), and audit logging.

Flowchart

A flowchart detailing the user authentication and authorization process was designed, illustrating the sequential steps from login page access, location permission request, credential verification, JWT token generation, middleware tenant and role checks, access granting or denial, to audit logging and logout (Ghadge, 2024).

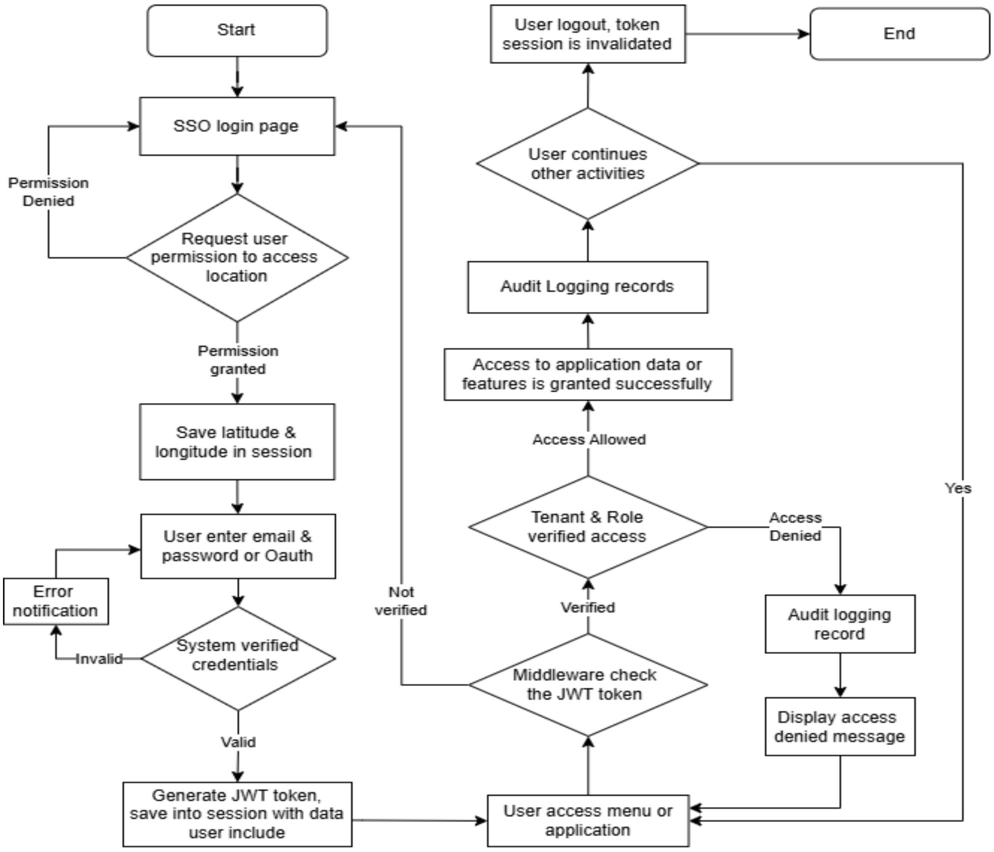

Start
SSO login page
Permission Denied
Request user permission to access location
Permission granted
Save latitude & longitude in session
User enter email & password or Oauth
Error notification
Invalid
System verified credentials
Valid
Generate JWT token, save into session with data user include
Not verified
User access menu or application
Middleware check the JWT token
Verified
Tenant & Role verified access
Access Denied
Audit logging record
Display access denied message
Access Allowed
Access to application data or features is granted successfully
Audit Logging records
User continues other activities
Yes
User logout, token session is invalidated
End

Figure 3. System flowchart

Figure 3 illustrates the flow of user authentication and access control using a combination of Single Sign-On (SSO) and Role-Based Access Control (RBAC). The process begins when the user accesses the SSO login page and is prompted to grant permission for location access. If granted, the latitude and longitude are stored in the session.

The user then proceeds to log in using email/password or OAuth. The system verifies the credentials and, upon success, generates a JWT (JSON Web Token) which includes user data and stores it in the session. Once authenticated, the user attempts to access the application. The middleware checks the JWT token, and access control is enforced by verifying the user's tenant and assigned role. If access is granted, the system logs the activity and the user is allowed to interact with application features. Otherwise, access is denied and an appropriate message is shown, with the attempt logged for auditing purposes. The flow also includes a logout mechanism, which invalidates the session and ends the user’s interaction securely.

Entity Relationship Diagram (ERD)

An ERD was developed to model the relationships between users, roles, permissions, tenants/branch, and audit logs. The diagram highlights the multi-tenancy data segregation, role assignments per tenant, and linkage of audit entries to respective user actions, supporting secure and scalable access control (Putu et al., 2025).

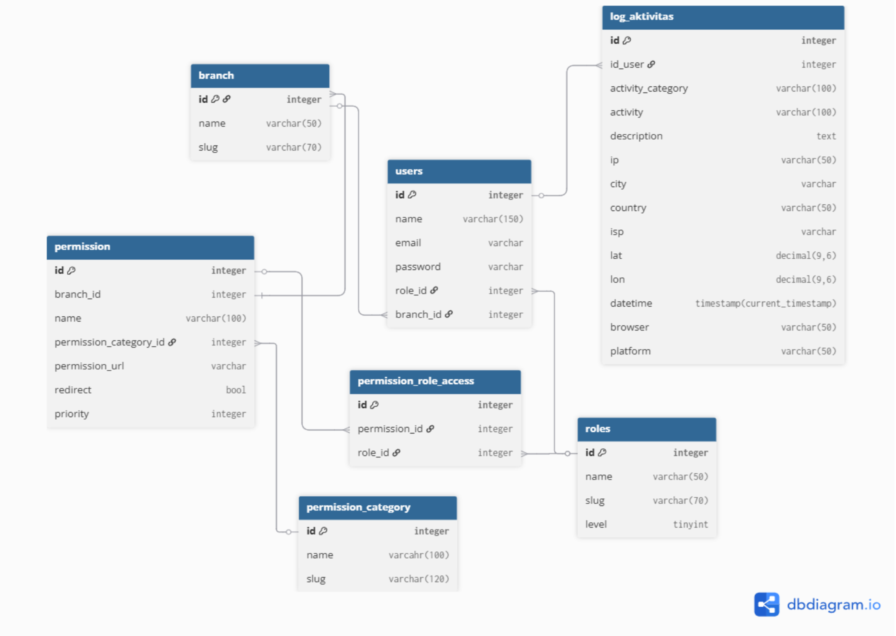


Figure 4. Entity relationship diagram (ERD) of SSO system

Figure 4 presents the database schema designed to support Single Sign-On (SSO) with Role-Based Access Control (RBAC). The schema ensures secure authentication, structured permission management, and comprehensive activity logging.

The users table stores essential user information, including credentials, role assignments (role_id), and branch affiliations (branch_id). The roles table defines access levels, which are linked to specific permissions via the permission_role_access junction table. This allows flexible assignment of multiple permissions to each role. The permission table holds detailed access control information, including URLs, categories (permission_category_id), and redirection flags, with optional branch-specific settings. Permissions are organized into logical groups through the permission_category table, supporting modular and scalable access management. branch allows user grouping and permission control across organizational divisions or physical locations. The log_aktivitas table functions as an audit log, capturing detailed user activity such as IP address, location (latitude, longitude), device/browser info, and timestamps—supporting accountability and traceability of system usage. This structure enables fine-grained access control and centralized auditing, making it suitable for multi-tenant applications requiring secure and transparent user authorization mechanisms.

Testing and Validation

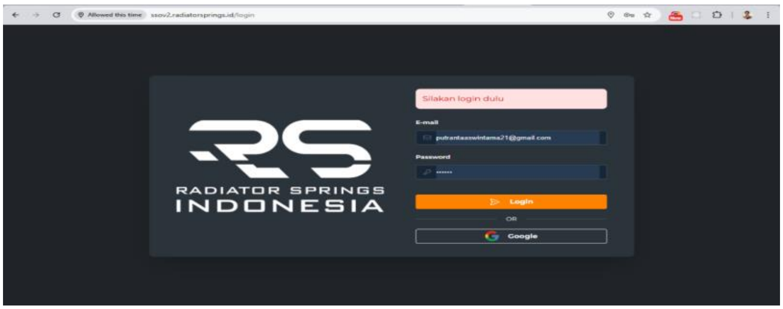
Comprehensive testing was conducted to ensure the effectiveness and robustness of the implemented system (Dona & Ciuffo, 2022). The testing was conducted including functional tests of role-based access and tenant isolation, security tests simulating unauthorized access and privilege escalation.Data was collected from system logs and performance metrics, and. Quantitative data such as login latency, access violation attempts, and user role assignment times were analyzed using descriptive statistics (Dimitrijević et al., 2024).

In this study, comprehensive testing focused on critical aspects such as role-based access, tenant isolation, and security through simulated tests for unauthorized access and privilege escalation. Additionally, performance metrics like login latency and role assignment times were analyzed. These tests provided a strong technical foundation for the system's functionality and security. In real life scenarios, the system effectively manages role conflicts and access needs across tenants by utilizing a multi-tenancy architecture combined with Role-Based Access Control (RBAC). Each tenant (such as departments or branches) operates in an isolated environment where users are assigned specific roles within their tenant. To handle role conflicts where the same user may have conflicting roles across multiple tenants the system uses tenant specific role assignments. This ensures that roles are localized to the tenant context, and conflicts are avoided by enforcing clear separation between the roles and permissions in each tenant. A user's role in one tenant does not automatically carry over to another tenant, minimizing potential conflicts and ensuring the correct access per tenant.

3. RESULTS AND DISCUSSIONS

Visual Evidence of System Behavior

Visual Evidence of System Behavior refers to screenshots that show how the system behaves in various scenarios, such as login attempts, error messages, or access denials. These images provide clear proof of the system's functionality and performance (Ody et al., 2023).



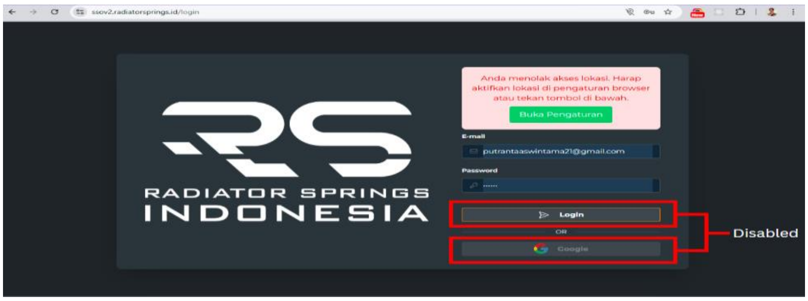
Silakan login dulu

E-mail: putrantaaswintama21@gmail.com

Password: ••••••

Figure 5. User interface of login screen

Figure 5 shows the initial login interface of the SSO system, demonstrating a clean, user-friendly design powered by Laravel Livewire. Users authenticate using email and password, which triggers JWT and OAuth-based authorization flows.



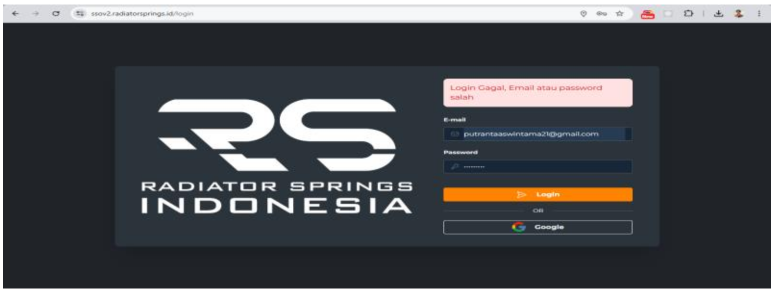
Anda menolak akses lokasi. Harap aktifkan lokasi di pengaturan browser atau tekan tombol di bawah.

E-mail: putrantaaswintama21@gmail.com

Password: ••••••

Figure 6. Access denied for unauthorized location

Figure 6 captures the system response when a user attempts to access resources from an unauthorized physical or network location. The system enforces access policies by rejecting the request and displaying an appropriate error message, enhancing security by location-aware controls.



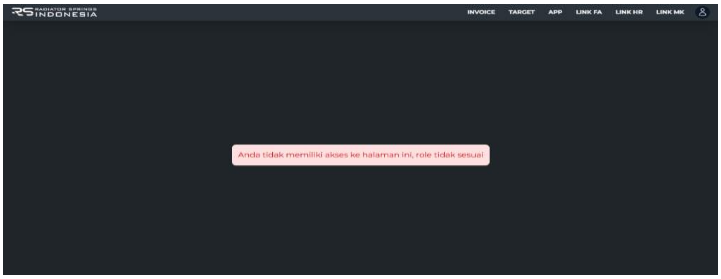
Login Gagal, Email atau password salah

E-mail: putrantaaswintama21@gmail.com

Password: ••••••••

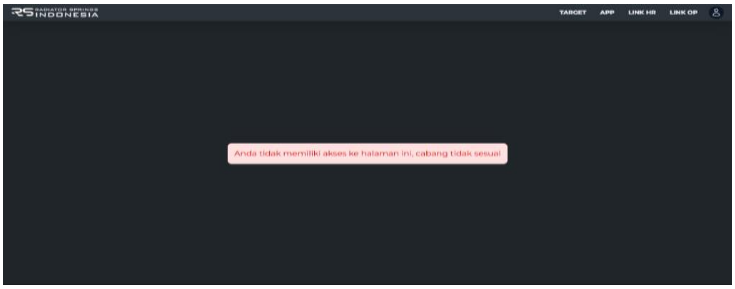
Figure 7. Login failure: incorrect email or password

Figure 7 display the validation feedback provided when a user enters incorrect credentials. Immediate error messaging improves user experience and mitigates brute-force attack risks by preventing ambiguous responses.



Anda tidak memiliki akses ke halaman ini, role tidak sesuai

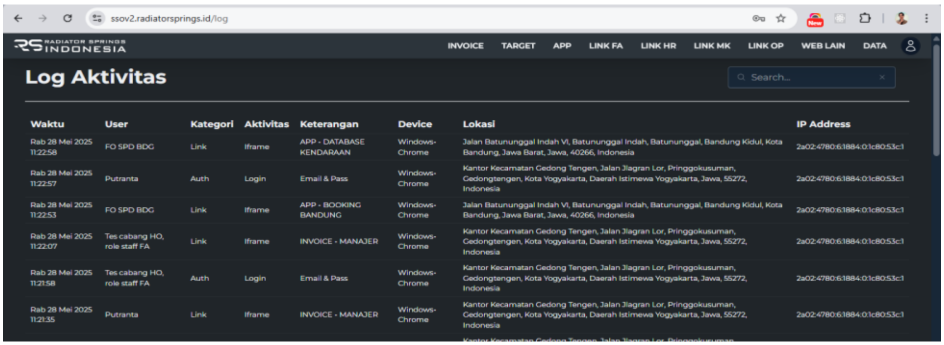
Figure 8. Error page unauthorized access due to RBAC restrictions



Anda tidak memiliki akses ke halaman ini, cabang tidak sesuai

Figure 9. Error page unauthorized access due tenant restrictions

Figure 8 and 9 show the error page shown when a user tries to access resources they are not authorized to, either due to role restrictions or tenant isolation. This prevents privilege escalation and cross-tenant data breaches.



Log Aktivitas

Waktu	User	Kategori	Aktivitas	Keterangan	Device	Lokasi	IP Address
Rab 28 Mei 2025 11:22:58	FO SPD BDG	Link	Iframe	APP - DATABASE KENDARAAN	Windows-Chrome	Jalan Batununggal Indah VI, Batununggal Indah, Batununggal, Bandung Kidul, Kota Bandung, Jawa Barat, Jawa, 40266, Indonesia	2a02:4780:6:1884:0:1c80:53c:1
Rab 28 Mei 2025 11:22:57	Putranta	Auth	Login	Email & Pass	Windows-Chrome	Kantor Kecamatan Gedong Tengen, Jalan Jlagran Lor, Pringgokusuman, Gedongtengen, Kota Yogyakarta, Daerah Istimewa Yogyakarta, Jawa, 55272, Indonesia	2a02:4780:6:1884:0:1c80:53c:1
Rab 28 Mei 2025 11:22:53	FO SPD BDG	Link	Iframe	APP - BOOKING BANDUNG	Windows-Chrome	Jalan Batununggal Indah VI, Batununggal Indah, Batununggal, Bandung Kidul, Kota Bandung, Jawa Barat, Jawa, 40266, Indonesia	2a02:4780:6:1884:0:1c80:53c:1
Rab 28 Mei 2025 11:22:07	Tes cabang HO, role staff FA	Link	Iframe	INVOICE - MANAJER	Windows-Chrome	Kantor Kecamatan Gedong Tengen, Jalan Jlagran Lor, Pringgokusuman, Gedongtengen, Kota Yogyakarta, Daerah Istimewa Yogyakarta, Jawa, 55272, Indonesia	2a02:4780:6:1884:0:1c80:53c:1
Rab 28 Mei 2025 11:21:58	Tes cabang HO, role staff FA	Auth	Login	Email & Pass	Windows-Chrome	Kantor Kecamatan Gedong Tengen, Jalan Jlagran Lor, Pringgokusuman, Gedongtengen, Kota Yogyakarta, Daerah Istimewa Yogyakarta, Jawa, 55272, Indonesia	2a02:4780:6:1884:0:1c80:53c:1
Rab 28 Mei 2025 11:21:35	Putranta	Link	Iframe	INVOICE - MANAJER	Windows-Chrome	Kantor Kecamatan Gedong Tengen, Jalan Jlagran Lor, Pringgokusuman, Gedongtengen, Kota Yogyakarta, Daerah Istimewa Yogyakarta, Jawa, 55272, Indonesia	2a02:4780:6:1884:0:1c80:53c:1

Figure 10. Audit logging dashboard

The audit dashboard presents a detailed, searchable log of user activities, including login attempts, role changes, and resource access. This dashboard allows administrators to monitor system usage in real-time and supports compliance audits.

Quantitative Metrics

Quantitative Metrics refer to the measurable data used to evaluate the system's performance, security, and efficiency. These metrics include factors like login time, unauthorized access attempts, system uptime, and audit log entries, providing concrete evidence of how well the system performs in real-world conditions (Barbeau et al., 2021).

Table 1.Quantitative metrics of testing results

Metrics	Before Implementation	After Implementation	Change %
Average login time (seconds)	1.204	1.401	+16.36%
Unauthorized access attempts	8 per month	< 1 per month	-87.5%
Time to onboard new user(min)	7	2	-71.43%
Audit log entries per day	N/A	300+	N/A

The implementation of Role-Based Access Control (RBAC), multi-tenancy, and audit logging in the Single Sign-On (SSO) system led to significant improvements in security, operational efficiency, and system monitoring, as evidenced by the quantitative metrics. After the integration, unauthorized access attempts dropped by 87.5%, highlighting the effectiveness of RBAC in restricting unauthorized access. Although the average login time slightly increased by 16.36%, this can be attributed to the additional security checks enforced by the new system, which enhance data protection. Furthermore, the onboarding process for new users saw a remarkable 71.43% reduction in time, indicating that multi-tenancy features streamlined the user management process, making it faster and more efficient. Additionally, the introduction of audit logging generated over 300 entries per day, providing a comprehensive record of system activities and facilitating improved tracking of user interactions. These results demonstrate the positive impact of implementing RBAC, multi-tenancy, and audit logging, particularly in enhancing security, reducing operational overhead, and ensuring compliance through detailed activity logs.

Comparative Analysis

Comparative Analysis refers to the process of comparing the system's performance before and after the implementation of new features or improvements. It involves analyzing key metrics, behaviors, and outcomes to highlight the differences and improvements made by the changes (Fernandes et al., 2022). Post-implementation, the system demonstrated significant improvements in preventing privilege escalation and cross-tenant data access, as illustrated by the Access Denied errors in Figure 6. The additional security measures effectively blocked unauthorized attempts to escalate privileges or access data from other tenants. While login latency slightly increased due to the added authorization checks, it remained within acceptable limits, ensuring a smooth user experience without noticeable delays.

Moreover, user management efficiency improved considerably, evidenced by a 71.43% reduction in time to onboard new users. This reduction was primarily due to the automated role assignment and streamlined processes for managing user access across multiple tenants, which significantly decreased the administrative burden.

Security Testing Results

Penetration tests were conducted to evaluate the system’s resilience against common attack vectors targeting access control vulnerabilities (Altulaihan et al., 2023). The tests focused on two primary threat scenarios: a) Privilege Escalation Attempts: Simulated attacks where users attempted to gain higher access privileges beyond their assigned roles; b) Cross-Tenant Data Access Attempts: Attempts by users to access or manipulate data belonging to other tenants (branches).

Table 2. Security testing results

Test Scenario	Number of Attempts	Successful Breaches	Obesrvations
Privilege Escalation Attempts	20	0	All unauthorized privilege elevations blocked
Cross-Tenant Data Access Attempts	20	0	Tenant isolation effectively enforced
Authentication Bypass Attempts	10	0	No token forgery or session hijacking detected

The table 2 summarizes the outcomes of various security test scenarios performed on the system. Three types of tests were conducted: privilege escalation, cross-tenant data access, and authentication bypass. Each test was attempted multiple times (20, 20, and 10 attempts respectively), with zero successful breaches reported across all scenarios. The observations indicate that unauthorized privilege elevations were blocked, tenant isolation was properly enforced, and no instances of token forgery or session hijacking were detected. These results demonstrate the effectiveness of the system's security mechanisms.

4. CONCLUSION

The implementation of the Single Sign-On (SSO) system, incorporating Role-Based Access Control (RBAC), Multi-Tenancy, and Audit Logging, has proven to be highly effective in enhancing authentication security and operational efficiency. The integration of these components addresses critical challenges in enterprise environments, especially those with complex user roles and multiple organizational units. The system effectively prevented unauthorized access and privilege escalation with 87.5% fewer unauthorized access attempts observed compared to preimplementation levels. Security policies were successfully enforced through RBAC and tenant isolation strategies. The integration of audit logging led to the creation of over 300+ log entries per day, providing full traceability of user activities. These logs are invaluable for security monitoring, incident detection, and regulatory compliance, particularly for industries requiring stringent access controls like healthcare and finance.

The SSO system has greatly improved overall security posture and user management efficiency, evidenced by a 71.43% reduction in time to onboard new users. By ensuring that users only have access to what they need (RBAC), protecting against cross-tenant data breaches (multi-tenancy), and maintaining a comprehensive audit trail of user actions, this system is well-suited for organizations with multiple departments, divisions, or branches.

However, it is important to note that this system was specifically designed for internal use within PT Radiator Springs Indonesia. While it addresses significant security and operational challenges, its implementation is not optimized for adoption by public organizations or local governments with limited technical resources. For broader adoption in such environments, additional considerations would be necessary, such as simplified deployment, cloud-based solutions, and cost-effective scalability. Ensuring user-friendliness and compatibility with legacy systems would be key for organizations with fewer technical capabilities.

Given the positive results within the internal company context, it is clear that adopting an SSO system like this provides organizations with enhanced security, operational efficiency, and regulatory compliance, while scaling to meet future demands. However, for wider implementation in the public sector, further modifications would be required to cater to different organizational needs and resource constraints.

To improve the integration of SSO with external applications, further development can utilize third-party identity federation protocols such as OpenID Connect and SAML. One step that can be taken is implementing a centralized identity broker to manage various identity providers such as Google, Microsoft, or Okta. Additionally, it is important to perform role mapping between the internal system and external applications to ensure consistent access control.

ACKNOWLEDGEMENTS

We sincerely thank all individuals and institutions who contributed to the completion of this research. Special appreciation is extended to our colleagues for their valuable insights, encouragement, and consistent support throughout the study. We are also grateful to the participants for their time and input, as well as to the supporting organizations for providing the necessary resources and facilities. Each contribution has been instrumental in the successful execution of this work.

REFERENCES

Adewale, T. (2024). *Identity-Centric Security in Cloud Computing: Safeguarding Workloads with Robust Access Controls*. https://www.researchgate.net/publication/389546976

Akhtar, A., Bakhtawar, B., & Akhtar, S. (n.d.). EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS. *International Journal of Technology, Innovation and Management (IJTIM)*, *2*, 2022. https://doi.org/10.54489/ijtim.v2i1.77

AlMaqousi, A. (2024, August). Enhancing Security in Remote Laboratory Environments: A Layered Approach. *Proceedings of the 6th International Conference on Statistics: Theory and Applications*. https://doi.org/10.11159/icsta24.164

Altulaihan, E. A., Alismail, A., & Frikha, M. (2023). A Survey on Web Application Penetration Testing. In *Electronics (Switzerland)* (Vol. 12, Issue 5). MDPI. https://doi.org/10.3390/electronics12051229

Barbeau, M., Cuppens, F., Cuppens, N., Dagnas, R., & Garcia-Alfaro, J. (2021). Resilience Estimation of Cyber-Physical Systems via Quantitative Metrics. *IEEE Access*, *9*, 46462–46475. https://doi.org/10.1109/ACCESS.2021.3066108

Chatterjee, S. (2022). Integrating Identity and Access Management for Critical Infrastructure: Ensuring Compliance and Security in Utility Systems. *Research Gate*, *8*(2). https://doi.org/10.5281/zenodo.14540999

Dimitrijević, N., Zdravković, N., Bogdanović, M., & Mesterovic, A. (2024). *Advanced Security Mechanisms in the Spring Framework: JWT, OAuth, LDAP and Keycloak*. http://ceur-ws.org

Dona, R., & Ciuffo, B. (2022). Virtual Testing of Automated Driving Systems. A Survey on Validation Methods. *IEEE Access*, *10*, 24349–24367. https://doi.org/10.1109/ACCESS.2022.3153722

Fareed, M., & Yassin, A. A. (2022). Privacy-preserving multi-factor authentication and role-based access control scheme for the E-healthcare system. *Bulletin of Electrical Engineering and Informatics*, *11*(4), 2131–2141. https://doi.org/10.11591/eei.v11i4.3658

Farhadighalati, N., Estrada-Jimenez, L. A., Nikghadam-Hojjati, S., & Barata, J. (2025). A Systematic Review of Access Control Models: Background, Existing Research, and Challenges. In *IEEE Access* (Vols. 13, 2025, pp. 17777–17806). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2025.3533145

Fernandes, G. V. O., Costa, B. M. G. N., Trindade, H. F., Castilho, R. M., & Fernandes, J. C. H. (2022). Comparative analysis between extra-short implants (≤6 mm) and 6 mm-longer implants: a meta-analysis of randomized controlled trial. In *Australian Dental Journal* (Vol. 67, Issue 3, pp. 194–211). John Wiley and Sons Inc. https://doi.org/10.1111/adj.12900

Fugkeaw, S., Rattagool, S., Jiangthiranan, P., & Pholwiset, P. (2024). FPRESSO: Fast and Privacy-Preserving SSO Authentication with Dynamic Load Balancing for Multi-Cloud-based Web Applications. *IEEE Access*, *12*, 157888–157900. https://doi.org/10.1109/ACCESS.2024.3485996

Ghadge, N. (2024). Optimizing Identity Management: Key Strategies for Effective Governance and Administration. *International Journal of Security, Privacy and Trust Management*, *13*(3), 01–11. https://doi.org/10.5121/ijsptm.2024.13301

Hron, M., & Obwegeser, N. (2022). Why and how is Scrum being adapted in practice: A systematic review. *Journal of Systems and Software*, *183*. https://doi.org/10.1016/j.jss.2021.111110

MARTIN, A. Q. B., AUVARIQUE, N. T., FIDÈLE, T., & NKENLIFACK, M. J. (2022). Continuous Single-Sign-On (CSSO) method for authentication and authorization in microservices architectures. In *Research Square*. https://doi.org/10.21203/rs.3.rs-1579462/v1

Nyembe, F. H., van der Poll, J. A., & Lotriet, H. H. (2023). Formal Methods for an Agile Scrum Software Development Methodology. *Proceedings of the International Conference on Advanced Technologies*. https://doi.org/10.58190/icat.2023.35

Ody, E., Straube, B., He, Y., & Kircher, T. (2023). Perception of self-generated and externally-generated visual stimuli: Evidence from EEG and behavior. *Psychophysiology*, *60*(8). https://doi.org/10.1111/psyp.14295

Olabanji, D., Fitch, T., & Matthew, O. (2023). Multi-tenancy in Cloud-native Architecture: A Systematic Mapping Study. *WSEAS TRANSACTIONS ON COMPUTERS*, *22*, 25–43. https://doi.org/10.37394/23205.2023.22.4

Owen, A., Oye, E., & Owen, A. (2022). *User Access Control Strategies in Enterprise Content Management Systems*. https://www.researchgate.net/publication/390232582

Putu, N., Ananda, T., Gusti, I., Pramesti, A., Putri, D., & Kusuma, N. (2025). Analysis and Design of Web-Based Inventory Receipt and Management Information Systems at Heycaps.Co Stores Using the Prototype Method. In *Jurnal Sistem Informasi dan Ilmu Komputer Prima (JUSIKOMP)* (Vol. 8, Issue 2).

PYROH, M., TERESHCHUK, G., & TOROSHANKO, O. (2025). AUTHENTICATION PRINCIPLES AS SECURITY ASPECTS OF WEB DEVELOPMENT. *MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES*, *1*, 294–301. https://doi.org/10.31891/2219-9365-2025-81-36

Vasudevan, A. (2023). *Master's Programme in Security and Cloud Computing Formal Analysis and Verification of OAuth 2.0 in SSO Modelling and Verification using PSPSP in Isabelle/HOL, and OFMC*.

Yu, L., Ma, S., Zhang, Z., Tao, G., Zhang, X., Xu, D., Urias, V. E., Lin, H. W., Ciocarlie, G., Yegneswaran, V., & Gehani, A. (2021). ALchemist: Fusing Application and Audit Logs for Precise Attack Provenance without Instrumentation. *28th Annual Network and Distributed System Security Symposium, NDSS 2021*. https://doi.org/10.14722/ndss.2021.24445

Yussuff, A. I. O., Goke, A., Folorunsho, H. B., & Adedoyin, M. A. (2024). Development of Integrated Web-Based Continuous Assessment Management System. *UNIOSUN Journal of Engineering and Environmental Sciences*, *6*(2). https://doi.org/10.36108/ujees/4202.60.0221

Zhang, Y., & Pan, F. (2022). Design and Implementation of a New Intelligent Warehouse Management System Based on MySQL Database Technology. *Informatica (Slovenia)*, *46*(3), 355–364. https://doi.org/10.31449/inf.v46i3.3968