Techno Nusa Mandiri: Journal of Computing and Information Technology
Vol. 22, No. 1 March 2025 | DOI: https://doi.org/10.33480/techno.v21i1.XXXX

SALES AND SAVINGS AND LOAN INFORMATION SYSTEM PROJECT AT PERKASA MITRA INSANI CONSUMER COOPERATIVE

Lucky Lukmanul Hakim1; Kelvan Derry Maulana2; Helmi Yunior3; Siti Masturoh4*

Information Systems Study Program1,2,3,4
Nusa Mandiri University, Jakarta, Indonesia[1,2,3,4]
https://www.nusamandiri.ac.id[1,2,3,4]
lucky.11250063@nusamandiri.ac.id[1]; kelvan.11250052@nusamandiri.ac.id2;
helmi.11250053@nusamandiri.ac.id3; siti.uro@nusamandiri.ac.id4*
(*) Corresponding Author



The creation is distributed under the Creative Commons Attribution-NonCommercial 4.0 International License.

Abstract—*Koperasi Konsumen Perkasa Mitra Insani faces operational challenges in managing retail transactions and savings and loans, which currently rely on a semi-manual system with fragmented spreadsheets. This conventional method results in slow financial cash reconciliation processes, a high risk of human error, the absence of a clear audit trail, and vulnerable financial data integrity due to the lack of a proportional Role-Based Access Control (RBAC) system. This research is designed to address these issues by developing a web-based Cooperative Information System that integrates Point of Sales (POS) services and savings and loans management circulation within a centralized architecture. This software development applies the System Development Life Cycle (SDLC) framework using the Waterfall model, which is systematically carried out through the stages of requirement analysis, system design, implementation, testing, and maintenance. The developed system also includes a transition from a monolithic structure to a decoupled architecture utilizing a REST API. Based on the results of Black-Box testing and User Acceptance Testing (UAT), the system is proven to effectively facilitate the automation of financial reporting, accurately enhance data security, and has achieved a high level of acceptance from both cooperative administrators and staff. The implementation of this information system contributes directly to strengthening transparency and time efficiency, as well as supporting the sustainability of the digital ecosystem transformation of Koperasi Konsumen Perkasa Mitra Insani in the era of global economic competition*

Keywords: *Information System, Cooperative, Sales, Savings and Loans, Web, REST API, RBAC, Waterfall.*

Intisari— *The Perkasa Mitra Insani Consumer Cooperative is faced with operational challenges in managing retail transactions and savings and loans that still rely on a semi-manual system with fragmented spreadsheets. This conventional method causes slow*
financial cash reconciliation process, high risk of human error, lack of a clear audit trail, and vulnerability to financial data integrity due to the lack of a proportionate Role-Based Access Control system. This research is designed to overcome these problems by developing a web-based Cooperative Information System that integrates sales services (Point of Sales) and savings and loan management circulation in one centralized architecture. The development of this software applies the Waterfall Model System Development Life Cycle (SDLC) method, which is carried out systematically through the stages of needs analysis, system design, implementation, system testing, and maintenance. The developed system also includes the transition from a monolithic structure to a decoupled architecture that utilizes REST API. Based on the results of Black-Box and User Acceptance Test (UAT) tests, the system has been proven to effectively facilitate the automation of financial reporting, improve data security accurately, and has obtained a high level of acceptance from cooperative management and staff. The implementation of this information system directly contributes to strengthening transparency, time efficiency, and supporting the sustainability of the transformation of the digital ecosystem of the Perkasa Mitra Insani Consumer Cooperative in the era of global economic competition.

Keywords: *Information Systems, Cooperatives, Sales, Savings and Loans, Web, REST API, RBAC, Waterfall.*

INTRODUCTION

In the context of increasingly digitized global economic developments, technological

Techno Nusa Mandiri: Journal of Computing and Information Technology
Vol. 22, No. 1 March 2025 | DOI https://doi.org/10.33480/techno.v21i1.XXXX

transformation is a crucial element for the operational continuity of an economic entity, including cooperatives. The Perkasa Mitra Insani Consumer Cooperative is still facing operational problems related to the recording of sales and savings and loans which are carried out semi-manually using fragmented spreadsheets(Ivena & Diaz, 2022).

This condition has an impact on the slow process of financial cash reconciliation, a high risk of human error, the absence of a transparent audit trail, and the vulnerability of data integrity due to the absence of role-based access control(Aswintama et al., 2025; Mujahidin et al., 2023). The cooperative's current local network system is not equipped with adequate access segmentation, so it has the potential to pose the threat of manipulation or loss of crucial transaction track record(Adi Saputra et al., 2023; Priadi & Idris, 2023).

Various previous studies have tried to develop information systems to digitize operational services, but there are still various gaps and limitations.

Research by Mujahidin et al. (2023) successfully implemented automated testing of REST APIs in cooperative applications, but its implementation has weaknesses because it is still focused exclusively on the Android mobile platform, making it less optimal for back-office operations. Furthermore, the research of Diaz and Sakaria (2022) designed a cash flow management system web-based, however, the system adopts a single accounting design model that is not fully relevant and is not able to accommodate integration with the retail cashier (Point of Sales) module. On the other hand, the study by (Annisa et al., 2024)and(Muharram & Hidayat, 2024)proving that the use of REST API with the Golang programming language has advantages in optimizing server performance and stability due to its concurrency capabilities, but the implementation of their research has not touched the complexity of the business flow of savings and loan cooperatives.

Based on the identification of advantages and disadvantages in previous research, there is a crucial research gap, namely the absence of an integrated system that is able to unite dual retail services (POS) and savings and loan circulation in one web-based ecosystem with a separate architecture (decoupled). Therefore, this study aims to design and build a sales and savings and loan information system at the Perkasa Mitra Insani Consumer Cooperative using REST API (Golang) and React JS communication. This system will accommodate the needs of POS integration, calculation of Residual Business Results (SHU), and savings and loan management in a central, transparent, effective, and efficient manner to answer the shortcomings of conventional monolithic systems.

MATERIALS AND METHODS

This research was carried out by taking a case study at the Perkasa Mitra Insani Consumer Cooperative, with the research subjects including the ranks of cooperative management, treasurers, and sales operational staff. The approach used is applied research that focuses on solving real problems in the field. Data collection is carried out through three main techniques:

1. Observation: Direct observation of the flow of retail transactions and the recording of cooperative financial cash.
2. Interviews: In-depth interviews with end-users to map out the specifics of system requirements and the allocation of access rights.
3. Literature Review: A comprehensive literature study related to cooperative regulation, retail inventory management, and software development architecture(Annisa et al., 2024; Aswintama et al., 2025).

The software development in this study uses the System Development Life Cycle (SDLC) method of the Waterfall model, which ensures that the system design runs sequentially, structured and effectively. The application of the Waterfall Method in this study is carried out in several stages, including:



Source : (Research Results, 2026)
Figure 1. Waterfall Method Flow

These stages include:

The software development methodology in this study uses the Waterfall sequential linear model adopted from the System Development Life Cycle (SDLC). The selection of the Waterfall model is based on the characteristics of the cooperative financial system which requires high discipline, complete documentation, and structured stages to avoid fatal errors in financial recording(Roger s

Pressman, 2021). This development process is divided into five phases that run in stages:

1. Requirements Analysis
   The initial stage focuses fully on identifying real operational obstacles faced by the Perkasa Mitra Insani Consumer Cooperative through field observation techniques and in-depth interviews with management and operational staff(Kenetth c.laudon & P.laudon, 2021). Based on the results of the analysis, it was found that the recording of separate retail sales and savings and loan circulation data caused delays in financial cash reconciliation and increased the chance of human error. The researcher identified two main aspects of needs, namely:
   a. Functional Needs: The system must be able to handle cashier transactions (Point of Sales), dynamically manage warehouse product inventory, record the circulation of deposit funds (principal, mandatory, voluntary), process member credit loan applications, calculate the distribution of Residual Business Results (SHU), and generate automatic consolidated financial statements(Law, 1992).
   b. Non-Functional Needs: Implementation of role-based access control (RBAC) systems to protect the confidentiality of financial data, use of JWT authentication tokens for data communication security, as well as rapid cashier interface response to serve high-volume transactions(Richards & Ford, 2021).
2. System Design
   This phase translates the requirements specifications into a software architecture blueprint that adopts a decoupled architecture model (Richards & Ford, 2021). Business process modeling and user interaction was carried out using Unified Modeling Language (UML). The researcher designed a Use Case Diagram to illustrate the system boundaries and access rights of the six main actors. Data flows and activity transitions were modeled through an Activity Diagram to illustrate the operational logic of each role. On the data storage side, the Entity Relationship Diagram (ERD) was constructed by establishing logical relationships between entities, including strong cardinality between member data, deposit accounts, inventory details, and cashier mutation records to ensure that there is no redundancy in financial data storage(PostgreSQL Global Development Group, 2022).
3. Implementasi (Implementation)
   The logical design and visual architecture are realized into functional program code lines by separating development into two parts:
   a) Backend (Server-Side): Developed using the Go programming language (Golang) to build a robust REST API. Golang was chosen for its excellence in handling large-scale data concurrency (goroutines) with efficient memory usage, it is ideal for processing queues of financial transactions simultaneously(Donovan & Kernighan, 2021).
   b) Frontend (Client-Side): Built using a React.js framework that implements the Virtual DOM feature to instantly update the cashier user interface without reloading the entire page, speeding up retail transaction service time(Sofi'ie & Qoiriah, 2023).
   c) Database: Using PostgreSQL as the primary relational database management system (RDBMS). PostgreSQL was chosen for its strict compliance with the ACID (Atomicity, Consistency, Isolation, Durability) standard to maintain the validity of financial transaction data (PostgreSQL Global Development Group, 2022). The security of interlayer communication is protected through a JWT token that is validated on each request.
4. Pengujian (Testing)
   System evaluation is carried out thoroughly to test technical and functional aspects of operations(Aliyah Aliyah et al., 2025; Roger s Pressman, 2021). Technical testing uses the Black-Box Testing method to validate the inputs and outputs of each menu function (such as login modules, stock filling, receipt printing, and reconciliation) to ensure that the functionality runs without errors. After the technical test is passed, a User Acceptance Test (UAT) is conducted involving end users (cashiers, treasurers, managers, and administrators) to simulate the real business scenario of the cooperative and measure the level of ease of operation of the system.
5. Maintenance
   The post-implementation phase includes system deployment on the cooperative's local server, product and member master data migration, and user training. The maintenance phase is carried out periodically to monitor server performance, mitigate network security risks, and implement bug fixes found during daily operations.

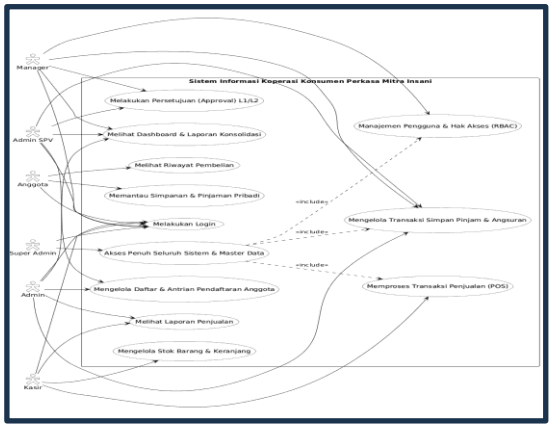
RESULTS AND DISCUSSION

Techno Nusa Mandiri: Journal of Computing and Information Technology
Vol. 22, No. 1 March 2025 | DOI https://doi.org/10.33480/techno.v21i1.XXXX

The process of designing and implementing the system has been successfully realized in the form of a web-based Cooperative Information System with a decoupled architecture. This system acts as a mini Enterprise Resource Planning (ERP) that summarizes the entire cooperative's financial operations.

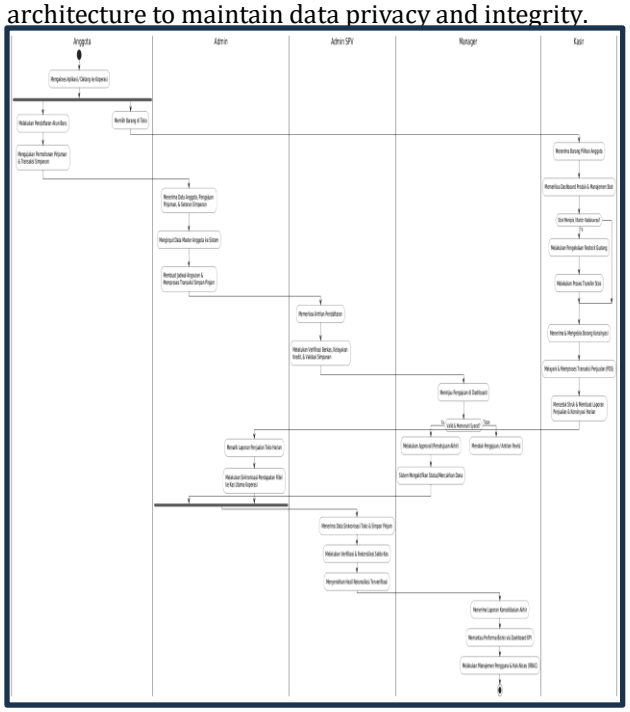
1. Planning of the PMI Cooperative Sales and Savings System

The logical architecture is represented through UML diagrams and data modeling. The Use Case Diagram divides access rights into several actors based on Role-Based Access Control (RBAC): Members (view deposit and loan history), Cashier (process POS sales transactions), Admin (manage inventory data and approve level 1 transactions), Super Admin (fully manage the system), and Manager (oversee financial statements). The Activity Diagram details the operational flow, for example in the sales process: when the cashier puts the product into the shopping cart, the system accumulates the total shopping directly, the cashier processes the cash/transfer payment, and then the system sends a response via the REST API to update the stock in real-time in the database table. In terms of databases, the Entity Relationship Diagram (ERD) is designed with solidly related tables such as Member entities that have one-to-many relationships with Deposits and Loans entities, as well as Product entities that are related to Transaksi_POS.



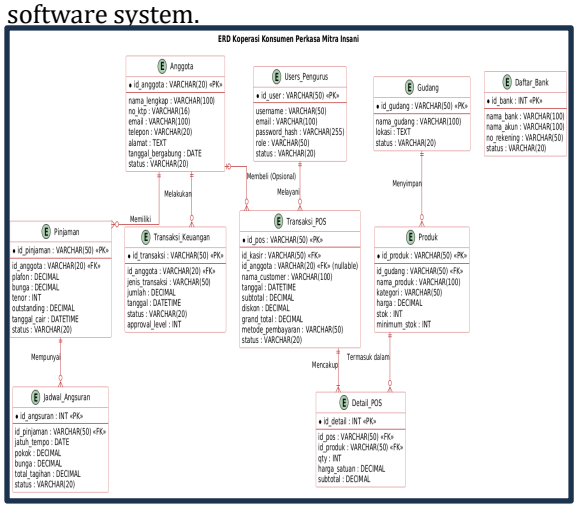
Source : (Research Results, 2026)
Figure 2.Usecase Diagram

Figure 2 divides access rights into several actors based on Role-Based Access Control (RBAC): Members (viewing deposit and loan history), Cashier (processing POS sales transactions), Admin (managing inventory data and approving level 1 transactions), Super Admin (fully managing the system), and Manager (overseeing financial statements). The design of these access restrictions aims to ensure the overall security of the system and avoid data manipulation. This is in line with the research(Richards & Ford, 2021) which emphasizes the importance of access segmentation in software architecture to maintain data privacy and integrity.



Source : (Research Results, 2026)
Gambar 3. ActivityDiagram

Figure 3 details the operational flow of the system, for example in the sales process: when the cashier inserts the product into the shopping cart, the system accumulates the total spend directly, the cashier processes the cash/transfer payment, and then the system sends a response via the REST API to update the stock in real-time in the database table. The design of this activity flow aims to improve transaction efficiency and minimize the potential for errors in the operational sales process. This supports a study (Pressman & Maxim, 2021) that shows that the clarity of activity modeling contributes to the operational reliability of the software system.

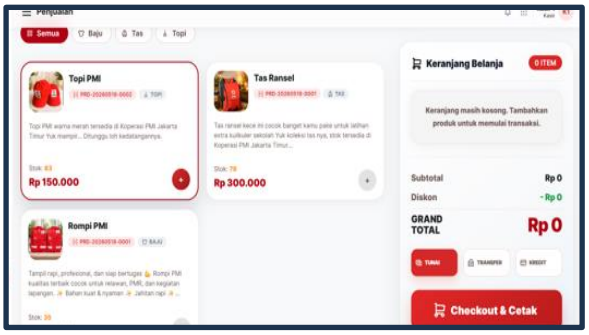


Source : (Research Results, 2026)
Gambar 4.Entity Relationship Diagram

Figure 4 illustrates the design of the database in the storage aspect with firmly related tables such as Member entities that have one-to-many relationships with Deposits and Loans entities, and Product entities that are related to Transaksi_POS. The design of this data relationship structure aims to minimize redundancy in financial data storage. This approach is in accordance with the relational database documentation standards (PostgreSQL Global Development Group, 2022) in order to maintain the validity of financial transactions through the fulfillment of the ACID principle.

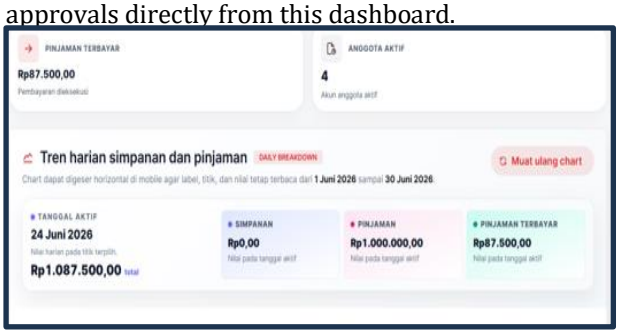
2. System and Interface Implementation (User Interface)

The system implementation combines the resilience of Golang's backend and the responsive interface of React JS. The User Interface (UI) display is designed to facilitate the speed and accuracy of transactions: Point of Sales (POS) interface: Provides an interactive screen for cashiers to enter products, view stock residue, automatically calculate the amount of change, and print digital receipts.



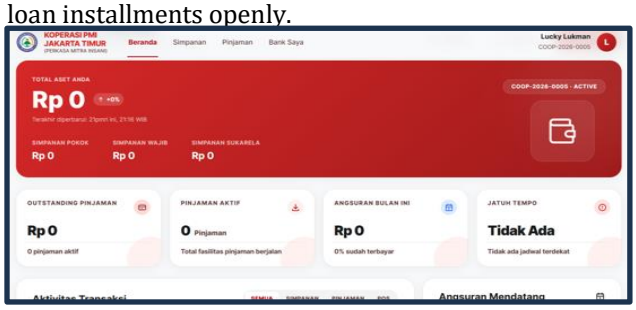
Source : (Research Results, 2026)
Figure 5.Point of Sales (POS) interface

Savings and Loan Dashboard: Displays a visual summary (graph) of total principal, mandatory, and voluntary deposits, as well as outstanding loan metrics. Admins and managers can perform tiered approvals directly from this dashboard.



Source : (Research Results, 2026)
Figure 6.Savings and Loan Dashboard

Member Portal: Provides a personal space for each member to monitor their membership status, deposit history, calculation of the value of their Residual Business Revenue (SHU), and remaining loan installments openly.



Source : (Research Results, 2026)
Figure 7. Member Portal

3. System Testing

The operational validation of the system was carried out through two comprehensive approach methods. First, Black-Box Testing, which was carried out on core scenarios such as authentication (login with incorrect credentials), transaction processing without stock availability, and inventory reduction validation. The results of the Black-Box test showed that the system successfully rejected invalid data and all transaction logic was executed 100% without computational errors. Second, the User Acceptance Test (UAT) test was carried out by directly involving the cooperative's management, treasurer, and cashier. The results of the UAT proved a very satisfactory level of user acceptance; users provided feedback that the system interface was easy to use and drastically accelerated the process of creating a recapitulation of monthly cash reporting.

Id. at 1	Modules/ Features	Testing Scenarios	Expected Results	Status
BB-01	Authentication & RBAC	Login with incorrect/unregistered credentials	The system denies access and displays an authentication error message	[✓ São Paulo [ ] File
BB-02	Authentication & RBAC	Login with the 'Cashier' role	The system redirects users only to the POS page	[✓ São Paulo [ ] File

Techno Nusa Mandiri: Journal of Computing and Information Technology
Vol. 22, No. 1 March 2025 | DOI https://doi.org/10.33480/techno.v21i1.XXXX

			and closes access to the save and borrow module	
BB-03	POS Module	Process sales transactions with sufficient stock	Successful transactions, receipts are printed, and the stock of goods is automatically reduced in the database	[✓] Pass [ ] Fail
BB-04	POS Module	Process sales transactions with out of stock	The system declines the transaction and gives a 'Insufficient Stock' notification	[✓] Pass [ ] Fail
BB-05	Save and Borrow	The Treasurer makes loan applications for Members	The data enters the 'Pending' status and awaits approval from the Administrator	[✓] Pass [ ] Fail
BB-06	Save and Borrow	Record loan installment payments	Members' loan balances are proportionally reduced and cooperative cash increases	[✓] Pass [ ] Fail
BB-07	Member Dashboard	Members check their savings and loan status	Perform an accurate and real-time recapitulation of members' financial data	[✓] Pass [ ] Fail
BB-08	Integritas Data	Reconciliation of retail cash profit with savings and loan mutations	The system can combine data automatically without inconsistencies (ACID compliance)	[✓] Pass [ ] Fail

CONCLUSION

Based on all the stages from preliminary to testing, it can be concluded that the problem of slow cash reconciliation and high human error due to the semi-manual recording system at the Perkasa Mitra Insani Consumer Cooperative has been successfully resolved. Through the application of the SDLC Waterfall methodology, this study has successfully designed and implemented an integrated Cooperative Information System with decoupled architecture (Golang REST API, React JS, and PostgreSQL). The results and discussion show that this system is able to unify the retail sales ecosystem (Point of Sales) and savings and loan circulation management in one centralized database, which is protected by Role-Based Access Control (RBAC)-based authentication. The results of the feasibility test through Black-Box Testing confirmed the absence of operational errors in the application functionality, and the User Acceptance Test (UAT) assessment validated the maximum satisfaction of the end users. This system has contributed to providing a clear audit trail, accelerating bookkeeping transparency, and closing the research gap of a single accounting system that was not integrated in previous studies.

REFERENCE

Adi Saputra, L., Muhammad Akbar, F., Cahyaningtias, F., Puspa Ningrum, M., & Fauzi, A. (2023). Security Threats to Corporate Management Information Systems. *Nusantara Cyber Education Journal*, *1*(2), 58–66. https://doi.org/10.38035/jpsn.v1i2.48

Aliyah Aliyah, Nahrun Hartono, & Asrul Azhari Muin. (2025). The Use of User Acceptance Testing (UAT) in Testing Financial Management Information Systems and Inventory of Goods. *Switch: Journal of Information Science and Technology*, *3*(1), 84–100. https://doi.org/10.62951/switch.v3i1.330

Annisa, R., Agung Ananda, R., & Sulistiono, W. E. (2024). Implementation of Golang Clean Architecture in Backend Point of Sales Website Design. *Journal of Informatics and Applied Electrical Engineering*, *12*(2), 2830–7062. http://dx.doi.org/10.23960/jitet.v12i2.4668

Aswintama, P., Haryanto, E., & Setyawan, R. A. (2025). Implementation of role-based access control, multi tenancy and audit logging in a single sign-on system. *Jurnal Mandiri IT*, *14*(1), 119–128. www.ejournal.isha.or.id/index.php/Mandiri

Donovan, A., & Kernighan, B. (2021). *The Go Programming Language.*

Ivena, C., & Diaz, A. (2022). *A web-based cash flow management system for GKT Antiochia Malang Church to facilitate reporting*. *10*(2), 63–72.

Kenetth c.laudon, & P.laudon, J. (2021). *Management Information Systems: Managing the Digital Firm (17th ed.). Pearson.*

Muharram, Y., & Hidayat, T. (2024). DEVELOPMENT OF E-COMMERCE BACK-END APPLICATIONS USING GOLANG REST API FOR SERVER PERFORMANCE OPTIMIZATION. *Journal of Informatics - Computing*, *11*, 7–13.

Mujahidin, S., Reinaldy Hermawan, M., & Cahyo Utomo, C. (2023). Implementation of Automated Test Case Generation in REST API on Android-Based Koperasi Application. *Journal of Information Systems and Informatics*, *5*(1), 123–133. https://doi.org/10.51519/journalisi.v5i1.431

PostgreSQL Global Development Group. (2022). PostgreSQL 16.11 Documentation The PostgreSQL Global Development Group. *The PostgreSQL Global Development Grou*, 1–3026. https://www.postgresql.org/docs/

Priadi, S., & Idris, M. (2023). Implementation of Rest in Building Web Services Using Golang (Case Study: Satudikti Application News Module). *INSTEK Journal (Informatics Science and Technology)*, *8*(2), 349–358.

Richards, M., & Ford, N. (2021). *Mark Richards & Neal Ford Software Architecture An Engineering Approach*. https://dokumen-pub.translate.goog/fundamentals-of-software-architecture-an-engineering-approach-1nbsped-1492043451-9781492043454.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc

Roger s Pressman. (2021). *Software Engineering: A Practitioner's Approach (9th ed.)*. https://doi.org/10.1145/336512.336521

Sofi'ie, F. A. F., & Qoiriah, A. (2023). Comparative Analysis of Javascript React and Vue Front-End Frameworks in Website Development. *Journal of Informatics and Computer Science (JINACS)*, *5*(02), 157–164. https://doi.org/10.26740/jinacs.v5n02.p157-164

Law, N. 25. (1992). doi:10.17977/um066.v3.i11.2023.4. Law, No. 25. 1992. "Law of the Republic of Indonesia No. 25 of 1992 concerning Indonesian Cooperatives." Regulation of the Republic of Indonesia (25): 1–57. https://www.peraturan.bpk.go.id. *Regulation of the Republic of Indonesia*, (25), 1–57. https://www.peraturan.bpk.go.id

Lampiran B. Bukti Submit Publish Artikel Ilmiah



UNIVERSITAS
NUSA MANDIRI