Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Design of a Multi-Tenant SaaS-Based Centralized Financial System Using a Silent Accounting Approach

Rizki Cahya Putri[1]*, Azhari Shouni Barkah[2], Aulia Suryaning Tyas[3], Intan Nur Sifa[4], Purnia Setiawati[5], Mayza Nurul Khasanatun Nisa[6], Sri Rahayu[7], Lina Nur Afifah[8]

[1,3,4,5,6,7,8]*Informatic, Faculty of Computer Science, Amikom Purwokerto University*
[2]*information Technology, Faculty of Computer Science, Amikom Purwokerto University*
rizzkicahyaputri127@gmail.com[1]*, azhari@amikompurwokerto.ac.id[2], suryaningg.tyas@gmail.com[3], intannrsfa@gmail.com[4], setiawatipurnia@gmail.com[5], mayzanurul55@gmail.com[6], srirahayu23sa11a117@gmail.com [7], linanurafifah14@gmail.com[8]

Abstract

Village financial management faces various fundamental challenges, including transaction recording that is still manual, a lack of integration between financial systems and village operational activities, and the absence of a platform capable of serving multiple villages within a single efficient infrastructure. These conditions result in financial reporting processes that are inefficient, error-prone, and difficult to account for. This study aims to design a centralized financial system based on a multi-tenant Software as a Service (SaaS) architecture using the Silent Accounting approach, defined as an automated transaction recording mechanism triggered by operational module activities without manual intervention. This study employs a qualitative descriptive method with a literature review approach. The design yields three main artifacts a system flowchart illustrating the workflow from user authentication, role assignment, and transaction validation through to automatic journal entry and posting to general ledger an Entity Relationship Diagram (ERD) modeling the database structure consisting of seven entities and a Data Flow Diagram (DFD) breaking down the system into five main processes A multi-tenant architecture with a 'tenant_id' column ensures data isolation between villages while allowing a single platform to serve multiple village simultaneously. The Silent Accounting mechanism ensures that all village financial activities are recorded consistently, accurately and in real time. The design is expected to serve as the foundation for the development and scalable village financial management platform.

Keywords: *Double-Entry Bookkeeping, Multi-Tenant, Silent Accounting, Software as a Service, Village Financial System*

1. Introduction

Transparent and accountable village financial management is the cornerstone of credible and effective village governance. With the enactment of Law No. 6 of 2014 on Villages, village governments have been granted the authority to manage Village Funds, which continue to increase annually, thereby requiring a more professional financial management system [1]. given the significant amount of funds allocated, village governments are required to present financial reports that are precise, accurate and accountable.

However, village financial management still faces various challenges, such as manual transaction recoding, a lack of structure, and village officials limited understanding of basic accounting principles [2]. These conditions make the financial reporting process inefficient and prone to errors. Furthermore, existing financial systems are generally not integrated with village operational activities, resulting in financial recording being conducted separately and not in real-time. This leads to delays in information presentation and suboptimal decision-making. The use of information technology through digital financial systems can be a solution to these problems. The implementation of an accounting information system in villages allows the entire process from planning to reporting to be carried out using a single software application, thereby making the management of village financial data easier and more accountable. However, currently available systems are generally still standalone and cannot automatically integrate all transaction source from various village operational modules into a single centralized platform. Many systems are still used separately and are incompatible with one another, making the financial data consolidation process slow and prone to inconsistencies. Furthermore, there is currently no approach that combines a multi-tenant architecture with an automatic recording mechanism, which would allow a single platform to be used by multiple village simultaneously while maintaining the isolation of each village's data.

Based on these issues, this study focuses on the design of a centralized financial system based on multi-tenant Software as a Service (SaaS) using the Silent Accounting approach, which is defined as a mechanism for automatically recording transactions based on the activities of operational modules without manual intervention from operators. The multi-tenant SaaS architecture allows a single platform to be used by many village simultaneously with data isolated between tenants via the 'tenant_id' column [3][4]. Thereby providing efficiency in terms of cost, maintenance, and system scalability. The design results are presented in the form of a system flowchart and an Entity Relationship Diagram (ERD) as representations of the proposed system's logical architecture.

2. Literature Review

2.1. Village Financial Management

According to Ministry of Home Affairs Regulation No. 20 of 2018, village financial management is a series of activities that includes planning, implementation, administration, reporting, and accountability regarding village finances. Village financial management is an integrated cycle where each stage is interconnected with the others it must be carried out by every village to ensure that governance, community empowerment, and village development proceed as planned [4]. transparency and accountability are two key principles that musk be upheld, as both significantly influence the effectiveness of village fund management [5].

2.2. Accounting Information System

An Accounting Information System is an organizational component that collects, classifies, processes, analyzes, and communicates financial information relevant to decision-making by both internal and external stakeholders of the organization. The primary function of an accounting information system is to process financial and non-financial transactions that directly impact the organization's financial processes. The implementation of an accounting information system enable the automation of various routine accounting tasks, such as recording transactions and preparing financial statements, thereby significantly reducing the risk of human error and improving operational efficiency [6].

2.3. Software as a Service and Multi-Tenant

Software as a Service is a cloud computing service model in which software is provided and accessed via the internet without requiring installation on the user's device [7]. The SaaS model offers cost efficiency and ease of maintenance because the service provider is responsible for all system management. A multi-tenant architecture is a software development approach in which a single application instance serves multiple users (tenants) simultaneously while maintaining data isolation between tenants [3]. Data isolation is implemented through a shared database mechanism with a unique identifier column 'tenant_id' that ensures each query accesses only the data belonging to the relevant tenant [8]. This approach allows a single platform to be used by many villages simultaneously at a lower operational cost compared to systems implemented separately.

2.4. Silent Accounting

In this study, Silent Accounting is defined as an automated mechanism for recording financial transactions triggered by the activities of the system's operational modules without manual intervention from operators. This concept adopts the principle of automation in Accounting Information System, whereby the system can automatically record financial transactions as they occur, reducing the need for time-consuming manual data entry [6]. Whit this approach, the process from journal entry to posting to the general ledger takes place behind the scenes, following the flow of transactions occurring in each operational module, so that operators do not need to perform separate accounting entries

2.5. Double-Entry Bookkeeping

Double-entry bookkeeping is an accounting recording system in which every transaction must be recorded in both the debit and credit columns. This method was adopted and developed due to its ability to avoid negative numbers. however, this system only provides information about the past in the form of equity and the present in the form of assets, whereas users of financial statements also need information about the future. Another weakness of this system is that it still relies on historical data predict the company's future condition, which considered insufficient to satisfy stakeholders [9].

2.6. Entity-Relationship Diagram

An Entity Relationship Diagram (ERD) is a data modeling method that uses diagrams to display information about entities and their attributes and to illustrate that relationships between entities. The main components of an ERD consist of entities, which represent real-world objects in the system; attributes, which are the properties of each entity; primary keys, which uniquely identify each row of data; and relationship which describe the connections between entities. Relationship in an ERD are divided into three types, one-to-one, one-to-many, and many-to-many [10]. with a well-design ERD, an information system is expected to improve data management efficiency and minimize errors in manual processes.

2.7. Data Flow Diagram

A Data Flow Diagram (DFD) is a system modeling technique that visually depicts data flow, helping developers and stakeholders understand how data is processed, stored, and flows within an information system prior to technical implementation. A DFD has four main components, processes, data flows, data stores and external entities. DFDs are organized hierarchically, starting from level 0, which depicts the system as a whole, down to level 1, which breaks the system down into major processes along with their relationship to data stores and related entities [11].

3. Research Methodology

3.1. Type of Research

This study is a qualitative descriptive study using a literature review approach. Descriptive research is used to describe, analyze, and design a system based on conceptual review without directly implementing the system. The literature review approach involves collecting,

examining and synthesizing various relevant scientific references, such as journal, articles and regulatory documents, as a foundation for designing a multi-tenant SaaS-based centralized financial system using the Silent Accounting approach.

3.2. Data Collection Methods

Data collection for this study was conducted through a literature review, specifically by examining various sources related to the research topic. The sources used include national an international scientific journals on accounting information system, SaaS architecture, multi-tenancy, village financial management, as well as applicable regulations such as Law No. 6 of 2014 on Villages and Ministry of Home Affairs Regulation No. 20 of 2019 on Village Financial Management. The collected data was used as the basis for determining system requirements, designing process flows, and developing data models, which were visualized in the form of flowchart and Entity Relationship Diagram (ERD), and Data Flow Diagram (DFD).

3.3. System Design Methodology

3.3.1. Requirements Analysis

The requirement analysis phase involves identifying the system's functional requirements based on the results of a literature review. The identified requirements include an automatic transaction recording mechanism using the Silent Accounting approach, cross-module financial data management, and data isolation between tenants using the 'tenant_id' column in the shared database architecture. This phase also identifies the actors involved in the system-namely, the financial administrator and the accounting administrator along with their respective access rights in accordance with designed workflow.

3.3.2. System Design

The system design phase produced two main design artifacts. First, a flowchart illustrating the system's workflow from the login process, role assignment, transaction receipt and validation, journal creation, through posting to the general ledger and report generation. Second, an Entity Relationship Diagram (ERD) that models the system's data structure, encompassing six main entities, 'tabel_transaksi', 'tabel_jurnal', 'tabel_jurnal_detail', 'tabel_buku_besar', 'tabel_akun' and 'tabel_anggaran', along with their relationship and attributes. Third, a Data Flow Diagram (DFD) consisting of a Context Diagram level 0 that illustrates the interaction between the system and external entities, and a DFD Level 1 that decomposes the system into five main processes, user authentication, account and budget management, transaction management, journal recording and posting and reporting.

4. Result and Discussion

4.1. Overview

The system designed in this study is a centralized financial module that automatically and integrally manages all village financial flows. This system is built on a multi-tenant SaaS (Software as a Service) architecture with a shared database approach, where each village registered as a tenant uses the same database infrastructure but their data is isolated via the 'tenant_id' column in each table. The system's primary function is to automatically receive transaction flows from all operational modules through the Silent Accounting mechanism a financial journal recording process that runs in the background without manual intervention from operators. Every incoming transaction is validated, recorded as a journal entry using the double-entry bookkeeping principle, posted to the general ledger, and ultimately generates real-time village financial reports. The system involves two user roles: the financial administrator, who is responsible for validating incoming transactions from all operational modules and the accounting administrator, who is responsible for the journal entry process, posting to the general ledger and generating financial reports.

4.2. System Architecture

This centralized financial system is designed using a Software as a Service (SaaS) multi-tenant architecture with a shared database approach. In this architecture, all tenants in the context of this study referring to villages using system share a single database infrastructure. Data isolation between tenants is achieved through the 'tenant_id' column present in every table ensuring that each village can only access its own data even through the data is physically stored in the same database. This approach was chosen because it is more efficient in terms of infrastructure management compared to a database per tenant architecture, while still guaranteeing the security and privacy of each village's financial data.

From a data flow perspective, this system implements the Silent Accounting mechanism as the foundation of financial integration. Every transaction occurring in the operational modules is automatically sent to the centralized financial system as a transaction event. The system then processes the event through reception, validation, journal entry and posting to the general ledger without requiring manual intervention from the operator. This flow ensures that all village financial activities are recorded consistently, accurately and real-time.

Overall, the system architecture consist of three main layers. First, the operational module layer responsible for generating and transmitting transaction data. Second, the financial processing layer encompassing transaction validation, journal entry and general ledger posting. Third, the reporting layer that automatically generates village financial reports based on the processed data.

4.3. Flowchart System

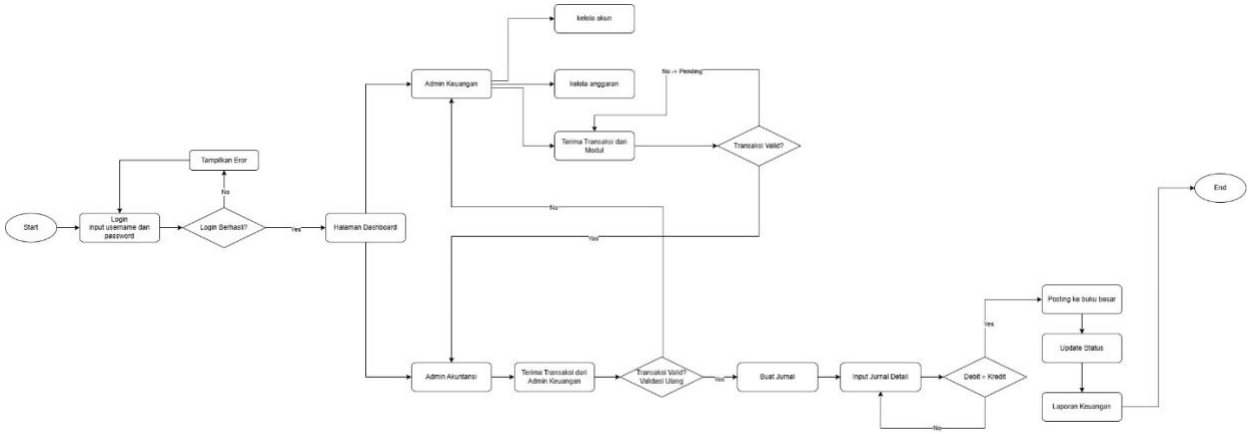

Fig. 1: Flowchart System

Figure 1 illustrates the workflow of the centralized financial system designed in this study. The system workflow begins with the authentication process, where users enter their username and password. If the login fails, the system display an error message and redirects the user back to the login page. If the login is successful, the user is redirect to the dashboard page, which serves as the system's navigation hub. From the dashboard page, the system detects the user's role and directs them to the appropriate path, namely Financial Admin or Accounting Admin.

In the Financial Admin section, there are three main menus available. First. The Manage Account menu, used to manage financial account master data (chart of accounts). Second, the Manage Budget menu, used to enter and manage annual budget data. Third, the Receive Transactions menu, used to receive incoming transactions from operational modules. The system then checks whether the transaction is valid or not. If invalid the transaction is saved with a pending status and the process is returned to the Financial Admin for review. If valid, the transaction is forwarded to the Accounting Admin workflow.

In the Accounting Admin workflow, the system receives transactions forwarded from the Financial Admin and performs a revalidation. If the transaction is invalid, the process is returned to the Financial Admin. If valid, the system automatically creates a journal entry via the Silent Accounting mechanism. The Accounting Admin then inputs the journal details, during which the system checks the balance between debit and credit amounts. If the balance is not met, the journal input process is repeated until balanced. Once balanced, the system posts to the general ledger, updates the transaction status, and automatically generates financial reports. The process concludes once the reports are successfully generated.

4.4. Entity Relationship Diagram

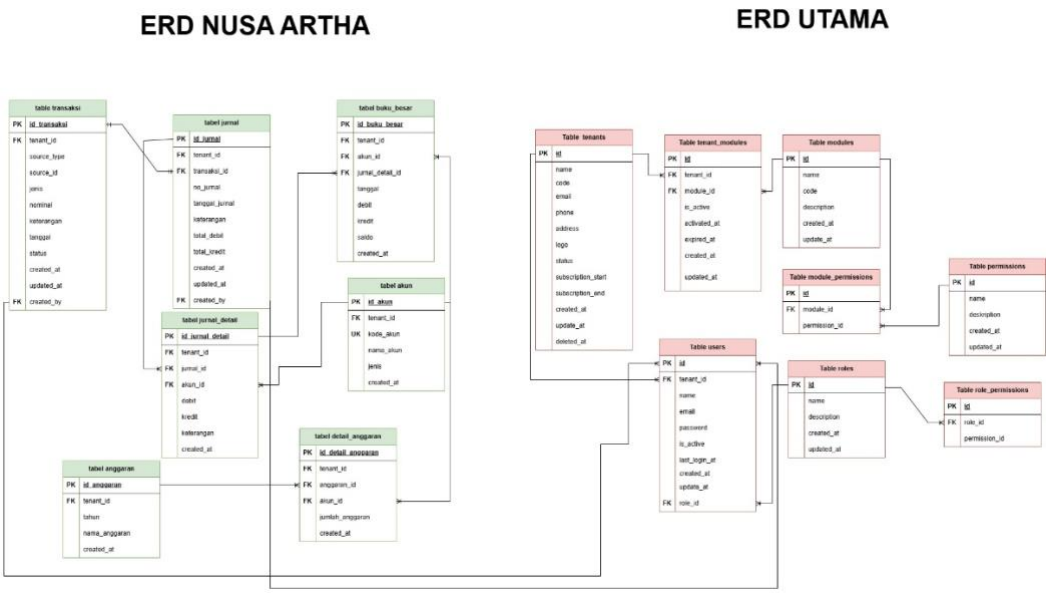

Fig. 2: Entity Relationship Diagram

The Entity Relationship Diagram in Figure 2 illustrates the database structure of the centralized financial system, which consist of main entities such as 'tabel_transaksi', 'tabel_jurnal', 'tabel_jurnal_detail', 'tabel_buku_besar', and 'tabel_akun', as well as supporting entities 'tabel_anggaran' and 'tabel_detail_anggaran'. The relationship between these entities reflect the connection between transaction data. Journal recording processes and general ledger management. transaction data from operational modules is documented in 'tabel_transaksi', while the details of each journal entry are stored in 'tabel_jurnal_detail' based on accounts defined in 'tabel_akun'. The system implements a multi-tenant approach through the tenant_id column present in every entity, facilitating isolated data management for each village using

the platform. Additionally, the 'tabel_transaksi' and 'tabel_jurnal' entities contain a created_by column as a foreign key referencing the 'table_users' in the main System ERD, indicating which user perfomed each transaction validation and journal entry.

4.5. Data Flow Diagram

4.5.1. DFD Level 0 (Context Diagram)

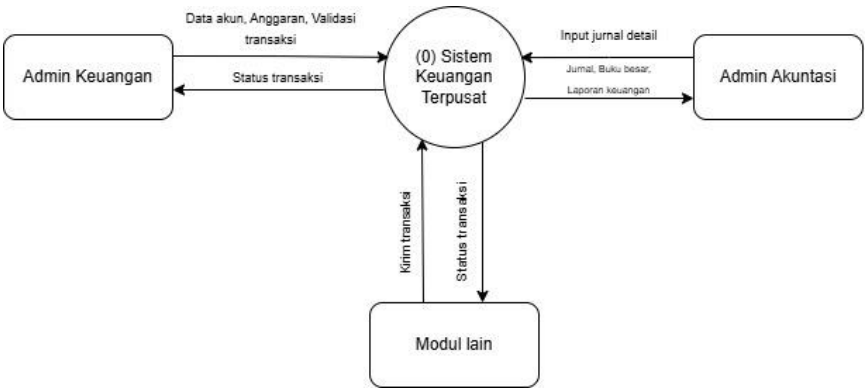

Fig. 3: Context Diagram

Figure 3 show a Lever 0 DFD that provides an overview of the centralized financial system. At this level, the system is depicted as a single process that interacts with three external entities. The Finance Administrator sends account data, budget, and transaction validation to the system and receives transaction status as feedback. The Accounting Administrator sends journal entry details as input and receives journal entries, general ledger data, and financial reports as output. Meanwhile, Other Modules act as sources of operational transactions that send transaction data to the system and receive confirmation of processing status.

4.5.2. DFD Level 1

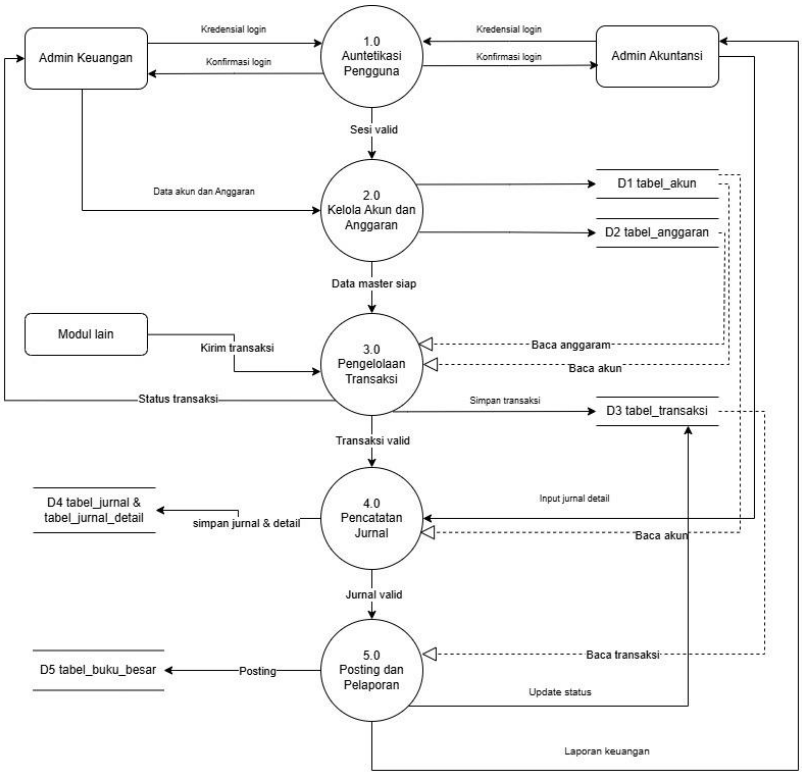

Fig. 4: Data Flow Diagram

Figure 4 shows a Level 1 DFD that breaks down the centralized financial system into five main processes. The first process is user authentication, which verifies the login credentials of both the Finance Administrator and the Accounting Administrator. The second process is account and budget management, which receives master data from the Finance Administrator and stores it in the 'tabel_akun' and 'tabel_anggaran'. The third process is transaction management, which receives data from Other Modules, stores it in the 'tabel_transaksi', and sends the transaction status to the Finance Admin. The fourth process is journal entry recording, which receives detailed journal inputs from the Accounting Admin and stores the results in the 'tabel_jurnal' and tabel_jurnal_detail' via the Silent

Accounting mechanism. The fifth process is posting and reporting, which posts to the 'tabel_buku_besar' and generates financial reports as the system's final output.

5. Conclusion

This study presents the design of centralized financial system based on a multi-tenant SaaS architecture using the Silent Accounting approach for village financial management. The design produces three main artifacts: a system flowchart that illustrates the complete workflow, from user authentication and dashboard navigation to transaction validation by the Finance Administrator, and finally automatic journal entry and posting to the general ledger by the Accounting Administrator an Entity-Relationship Diagram (ERD) that models the database structure consisting of seven interrelated entities, and a Data Flow Diagram comprising Context Diagram (Level 0) and Level 1 DFDs that break down the system into five main processes.

The Silent Accounting mechanism enables transactions from operational modules to be recorded automatically without manual intervention, thereby reducing the risk of errors and improving recording efficiency. The multi architecture, which uses a shared database and the 'tenant_id' column as a data isolation mechanism, allows a single platform to serve multiple villages simultaneously at a lower operational cost while maintaining the security and privacy od each village's financial data. The proposed system design is expected to serve as the foundation for the development of an integrated, transparent, accountable and scalable village financial management platform. Further research is recommended to proceed to the system implementation and testing phase to evaluate the design's performance and effectiveness in a real-word environment.

References

[1] P. Aplikasi, S. Informasi, K. Desa, S. Dan, S. Firmawati, and Y. E. Putri, "PENGARUH APLIKASI SISTEM INFORMASI KEUANGAN DESA (SISKEUDES) DAN KOMPETENSI APARATUR DESA TERHADAP AKUNTABILITAS PENGELOLAAN KEUANGAN DESA DI KABUPATEN SUMBAWA BARAT [The Influence Of The Village Financial System Application (SISKEUDES) And Village Appara," pp. 852–856, 2025.

[2] C. Betty *et al.*, "Penguatan Tata Kelola Keuangan Desa melalui Pelatihan Penyusunan Laporan Keuangan," vol. 6, no. 6, pp. 715–721, 2025, doi: 10.31004/abdidas.v6i6.1218.

[3] S. Pushpan, "Multi-Tenant Architecture : A Comprehensive Framework for Building Scalable SaaS Applications," pp. 1117–1126, 2024.

[4] D. Nuriman and R. Komalasari, "Rancang Bangun Aplikasi Web Berbasis Multi-Tenant untuk Layanan Konsultasi Tugas," vol. 8, no. 4, pp. 1854–1865, 2025.

[5] Baihaqi;, D. Asmawanti-S;, and M. R. Putradana, "THE VILLAGE FINANCIAL MANAGEMENT THROUGH ACCOUNTABILITY, TRANSPARENCY AND COMMUNITY PARTICIPATION," vol. 6, no. Februari, pp. 15–36, 2023.

[6] D. H. Anriva, "TANTANGAN DAN SOLUSI PENERAPAN SISTEM INFORMASI AKUNTANSI DI INDONESIA : SEBUAH ANALISIS TEMATIK," vol. 13, no. 2, pp. 97–109, 2024.

[7] W. Setiawan, N. Fajriyah, T. Duha, U. M. Tangerang, and U. Nias, "ANALISA LAYANAN CLOUD COMPUTING DI ERA DIGITAL," vol. 1, no. 1, pp. 32–39, 2022.

[8] M. I. Athallah, A. T. Ayunda, S. Informasi, and U. Pradita, "Model Arsitektur Multi-Tenant Berbasis Role-Based Access Control pada Sistem Manajemen Waralaba".

[9] M. F. Mahaini, F. Regina, A. Danessa, and A. Kusumastuti, "Treping System (New Trebit for Bookkeeping System): Implementasi Triple Entry Accounting dalam Sistem Entitas Pendidikan," 2022.

[10] M. R. Akbar, A. Zurfadly, and M. Apriani, "PERANCANGAN DATABASE ELITE HOTEL TEMBILAHAN MENGGUNAKAN ERD (ENTITY RELATIONSHIP DIAGRAM)," vol. 3, no. 2, pp. 105–117, 2025.

[11] A. Nova, K. Pane, E. Sri, E. Rumapea, F. Azzahra, and I. Anugrah, "TUGAS AKHIR BERBASIS WEBSITE PADA PROGRAM STUDI TEKNIK INDUSTRI ITEBA MENGGUNAKAN PENDEKATAN DATA FLOW DIAGRAM (DFD)," vol. 8, no. 2, pp. 259–265, 2025.