



Implementation of Serverless Architecture and Real-Time Database on Laprak Tracker Application using ReactJS and Firebase

Haidar Al-mutawalli^{1*}, M. Raghif Hibrizi Daulay², Muhammad Imam Khairi³, Habiburrozzaq Lubis⁴, Bayu Azi Ramadan⁵, Mhd.Furqan⁶

^{1,2,3,4,5,6} Universitas Islam Negeri Sumatera Utara

haidar0701232073@uinsu.ac.id^{1*}, mraghifdaulay@gmail.com², m.imamkhairi@gmail.com³, habiburrozzaqlubis@gmail.com⁴, bayuzairmdn@gmail.com⁵, mfurqan@uinsu.ac.id⁶

Abstract

Management of laboratory practicum reports often faces challenges such as inefficient progress tracking and manual data synchronization between students and laboratory assistants. This research aims to develop a web-based application named "Laprak Tracker" to streamline this process. The system is built using a serverless architecture with Google Firebase as the backend and Cloud Firestore for real-time data synchronization. The frontend is developed using ReactJS with a Single Page Application (SPA) approach to ensure responsiveness and high performance. The methodology used is the Waterfall model, encompassing requirements analysis, system design, implementation, and testing. Testing results using the Black Box method show that all features, including authentication, module management, and real-time status updates (ACC tracking), function correctly. This application provides an efficient solution for students to monitor their practicum progress and ensures data integrity through a centralized cloud system.

Keywords: Cloud Firestore, Firebase, Laboratory Management, ReactJS, Serverless Architecture

1. Introduction

Laboratory activities are an essential component of the academic curriculum in higher education, requiring students to complete various practical modules. These activities involve complex documentation and approval processes between students and laboratory assistants to ensure learning objectives are met [1]. Information management systems in laboratories are becoming increasingly vital to ensure that all student tasks are recorded efficiently. A web-based management system can save time and facilitate better communication during the report submission process, especially in a technical environment [2].

In developing such academic tools, the user interface and user experience (UI/UX) design are critical factors that determine the success of the system. A clean and navigable interface allows users to focus on their primary tasks without technical distractions or confusion [3].

Modern web application development often utilizes library-based frameworks like ReactJS to create high-performance interfaces. This technology allows for the creation of responsive components and interactive elements that significantly enhance user interaction [4]. The trend in current cloud computing has moved toward serverless architecture to simplify application development and infrastructure management. This approach eliminates the need for physical server maintenance while maintaining high scalability for various information system needs [5].

Furthermore, real-time data synchronization is a key feature for monitoring applications, ensuring that updates are immediately visible to all connected users. Technologies like Google Firebase provide the necessary infrastructure to handle real-time database needs effectively without complex backend configurations [6].

Despite the availability of these technologies, many students still face challenges in tracking their personal practicum progress. Problems such as forgetting the current module status, losing track of "ACC" (approvals) from different assistants, and the lack of a centralized dashboard often lead to delays and disorganization in report completion[7]. Furthermore, manual data management systems are often prone to errors and delays in information processing, necessitating the development of more comprehensive web-based solutions [8].

This research introduces "Laprak Tracker," a web-based application designed to solve these personal tracking issues. By implementing ReactJS for the frontend and Firebase for the serverless backend, this application provides a real-time monitoring solution for students. The primary objective is to evaluate how these technologies can be integrated to create an effective and user-friendly lab progress tracker.

2. Research Method

The development of the "Laprak Tracker" application follows the Waterfall model as the primary framework for the System Development Life Cycle (SDLC). This model provides a structured and sequential approach, which is often used to automate manual processes and provide a clear visualization of system requirements [9]. This structured approach ensures that each phase from requirements gathering to testing is completed before moving to the next stage [10].

2.1. Requirements Analysis

In this initial phase, the needs of the users (students and laboratory assistants) were identified. The core functional requirements include a secure authentication system, a management dashboard for practicum modules, and a real-time tracking mechanism for "ACC" (approvals). This requirements extraction process is essential to ensure that the final application addresses the actual problems faced in laboratory management [1].

2.2. System Architecture Design

The application is designed using a serverless architecture to ensure high scalability and minimal infrastructure management. This approach allows the frontend to communicate directly with backend services via APIs provided by Google Firebase [5]. The system architecture consists of a ReactJS frontend that handles user interactions and Cloud Firestore as a NoSQL database that stores user data and practicum progress in a document-based structure [11].

2.3. Implementation

The implementation phase translates the design into a working system using modern web technologies. The frontend is built using Vite and ReactJS, utilizing a component-based approach to ensure code modularity and maintainability [12]. For the user interface, Material UI (MUI) is implemented to provide responsive and consistent design elements across various screen sizes [13]. The core logic for real-time synchronization is handled through the onSnapshot listener in Firebase, which updates the UI immediately when changes occur in the database [14].

2.4. Testing

The final phase involves validating the application's functionality. This research employs Black Box Testing to evaluate the system based on its specifications. Each feature, such as user registration, module CRUD operations, and progress bar calculations, is tested to ensure that the outputs match the expected results without focusing on the internal code structure [15].

3. Results and Discussion

3.1. System Architecture

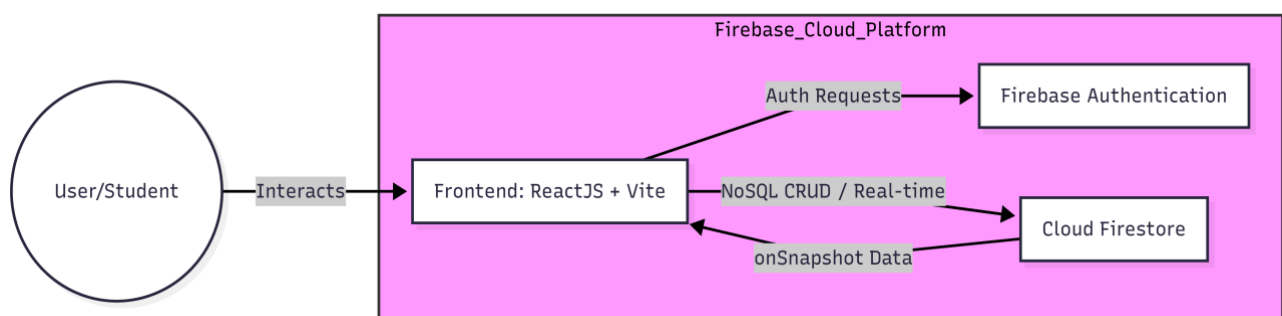


Fig. 1: System architecture diagram of the Laprak Tracker application

The "Laprak Tracker" application is built using a serverless architecture, which eliminates the need for manual server management and allows for automatic scaling. The frontend, developed with ReactJS, communicates directly with Google Firebase services. Firebase Authentication handles secure user access, while Cloud Firestore serves as the NoSQL database for storing practicum data.

3.2. System Logic and Workflow

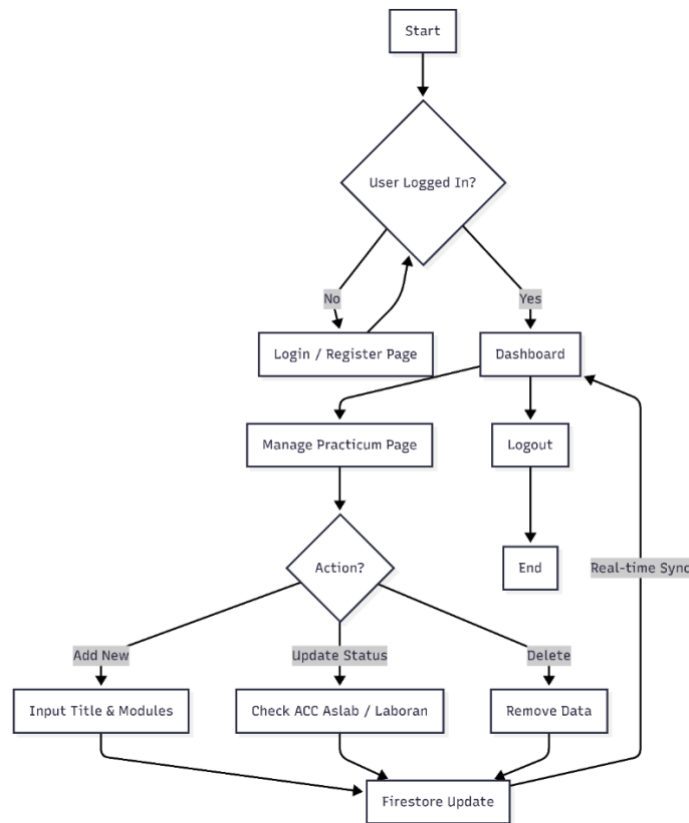


Fig. 2: Workflow flowchart illustrating the user journey from authentication to data management

The workflow of the application is designed to be intuitive for individual students. After a successful login, the system fetches the user's specific data from Firestore. The main logic resides in the real-time tracking of modules, where each update to the "ACC" status automatically triggers a progress calculation.

3.3. Database Schema Design

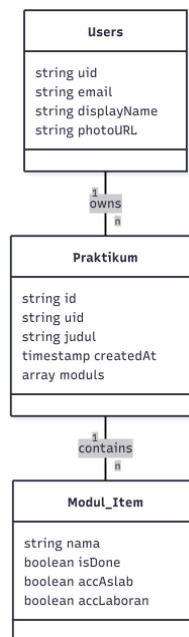


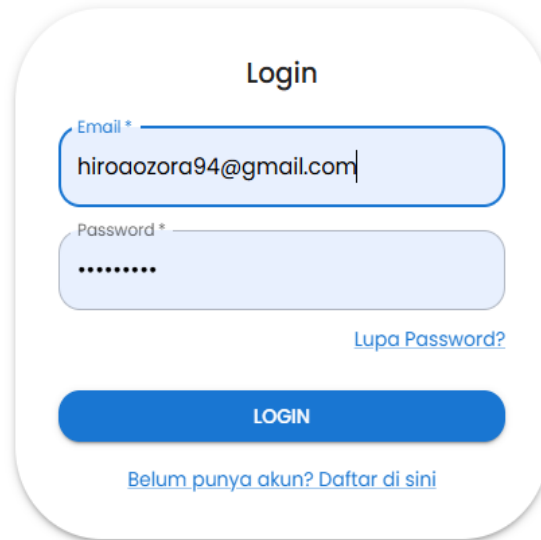
Fig. 3: NoSQL database schema and collection structure in Cloud Firestore

Since Cloud Firestore is a NoSQL database, the data is organized into collections and documents. The schema focuses on two main collections: users and praktikum. Each document in the praktikum collection contains an array of modules with their respective approval statuses.

3.4. Implementation and Interface

The implementation of "Laparak Tracker" results in a responsive web application that facilitates students in managing their laboratory tasks. The user interface is designed with a focus on simplicity and ease of use, utilizing Material UI components to maintain consistency.

1. Authentication Page



The login form is titled "Login" and is contained within a rounded rectangular box. It features two input fields: "Email *" with the value "hiroaozora94@gmail.com" and "Password *" with masked characters ".....". A blue "LOGIN" button is positioned below the password field. To the right of the password field is a link "Lupa Password?". Below the button is another link "Belum punya akun? Daftar di sini".

Fig. 4: User interface design for the login and registration authentication pages

Users must register and log in to access the dashboard. This ensures that the data stored in Cloud Firestore is personalized and secure for each student, following the security standards of Firebase Authentication.

2. Dashboard

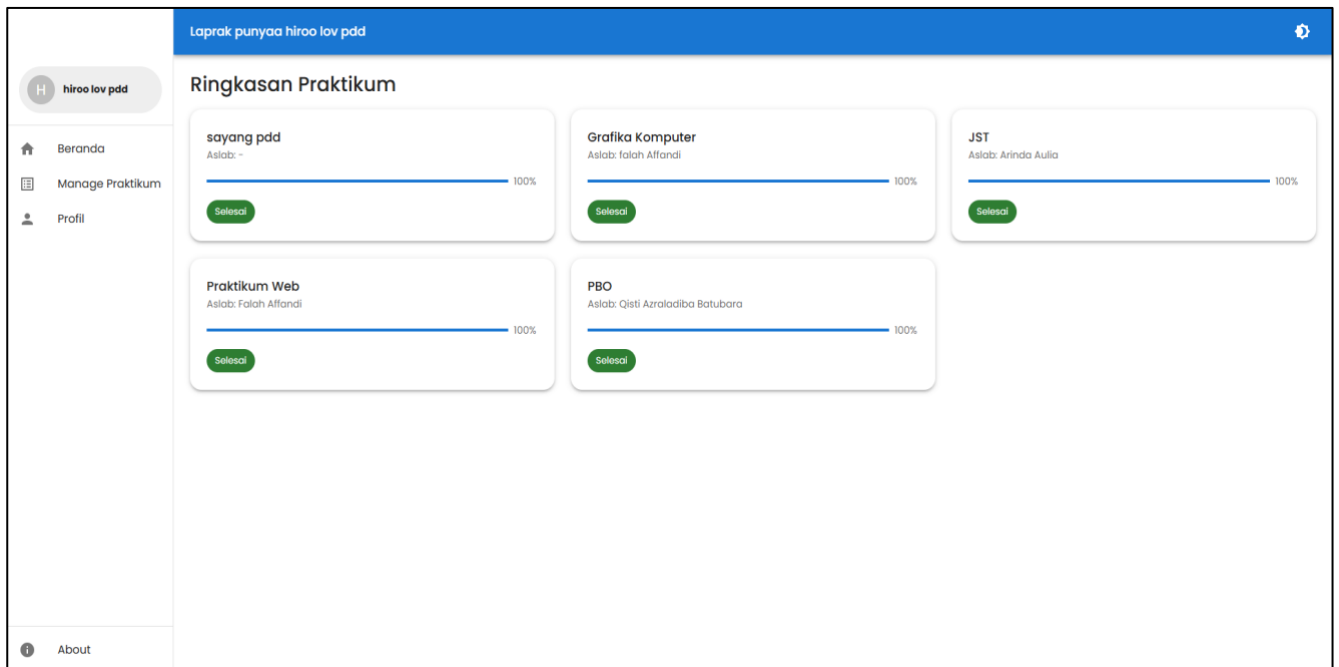


Fig. 5: Main dashboard interface displaying personal practicum progress and visual status

The dashboard displays a summary of the student progress. A dynamic progress bar calculates the percentage of completed modules based on the total number of tasks. This visual representation helps students quickly identify their current standing in a specific practicum.

3. Management Page

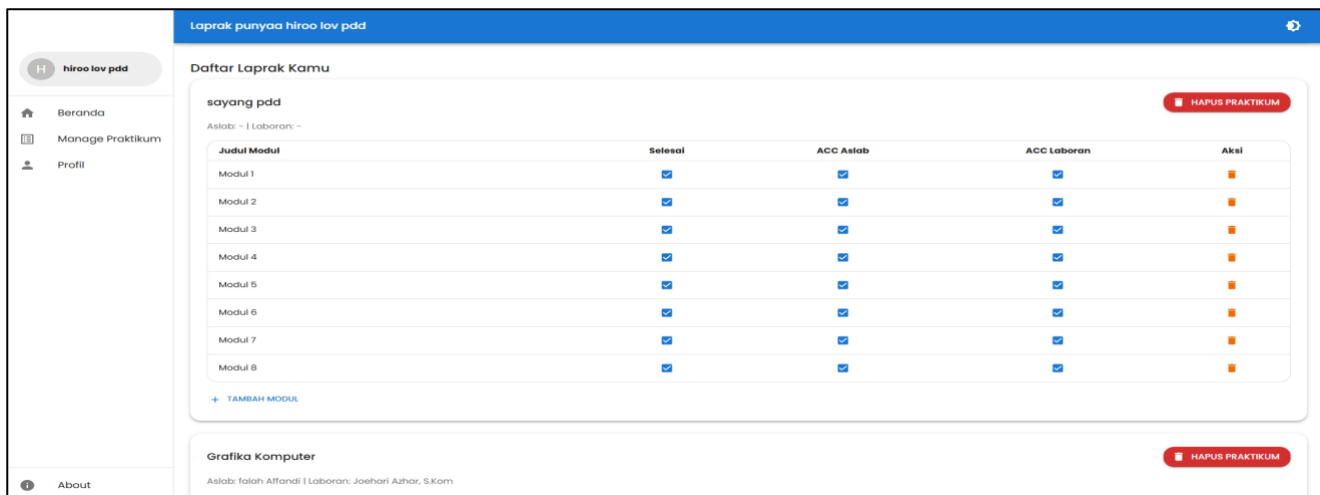


Fig. 6: Management interface for adding, editing, and tracking module approval status

This feature allows users to perform CRUD (Create, Read, Update, Delete) operations on practicum modules. Users can add new modules, edit titles, and track the "ACC" status from assistants and laboratory staff.

4. Profil Page

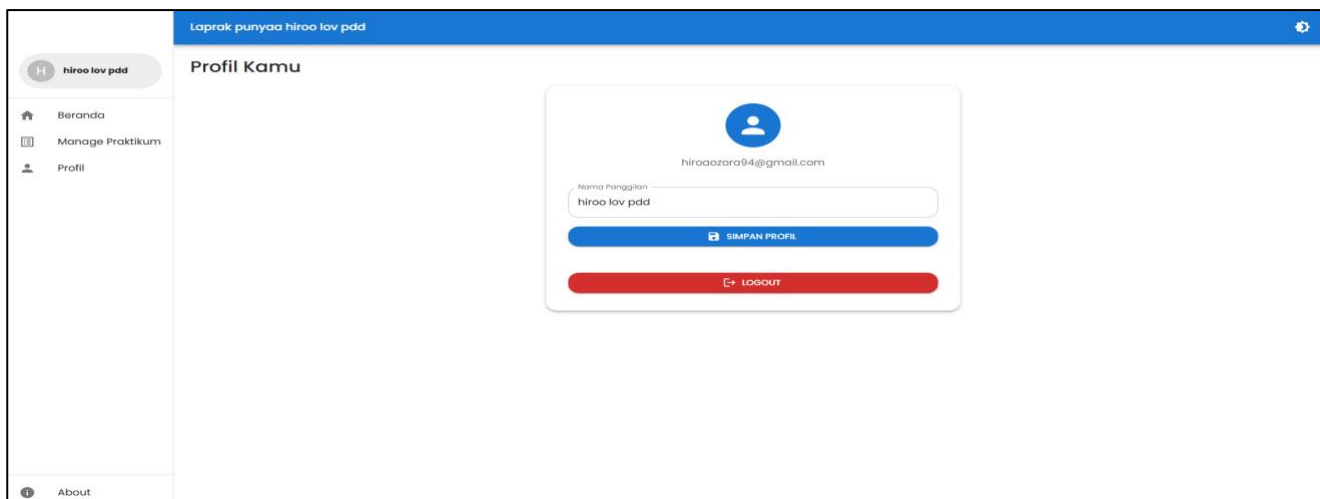


Fig. 7: User profile page for account management and interface personalization

The Profile Page (Figure 7) serves as the management center for user identity. It retrieves personal data such as display name, email, and profile picture directly from Firebase Authentication. This page allows users to personalize their accounts by updating their display names and photo URLs, which are then synchronized across the application via the AuthContext. Additionally, the profile page integrates a theme toggle for switching between light and dark modes, enhancing user comfort during extended use.

3.5 Real-time Data Synchronization Analysis

The core strength of the "Laptrak Tracker" application lies in its ability to synchronize data in real-time between the client and the cloud database. Unlike traditional web applications that require a page reload or manual polling to fetch updated data, this system utilizes a listener-based approach.

The implementation of the onSnapshot function allows the application to maintain a persistent connection with Cloud Firestore. When a user updates a module's status such as marking an "ACC" from an assistant, the change is instantly propagated to the database and pushed back to the user interface. This ensures that the progress bar and status indicators are always accurate and up-to-date. This analysis confirms that the integration of reactive frontend components with a real-time NoSQL backend significantly reduces data latency and provides a seamless interaction experience for students tracking their practicum milestones.

3.6 Functional Testing (Black Box Testing)

The system was tested using the Black Box method to ensure all functionalities work according to the requirements.

Table 1: Functional testing results using the Black Box testing method

No	Test Case	Test Scenario	Expected Result	Result	Status
1	User Registration	Registering a new account through the sign-up form	Account is successfully created and stored in Firebase Auth	Success	Valid
2	User Login	Logging in with valid email and password	User successfully gains access to the main dashboard	Success	Valid
3	Add Practicum	Adding a new practicum module to the system	New module appears in the list and is saved in Firestore	Success	Valid
4	Update Status (ACC)	Changing the progress status of a practicum	Progress bar and status icons update in real-time	Success	Valid
5	Delete Practicum	Removing a practicum module from the list	Module is removed from the UI and Firestore database	Success	Valid
6	Theme Toggle	Switching the application theme mode	Interface toggles between light and dark mode instantly	Success	Valid

4 Conclusion

The development of the "Laparak Tracker" application demonstrates that the integration of ReactJS and a serverless architecture using Google Firebase provides an effective solution for laboratory practicum monitoring. By utilizing Cloud Firestore's real-time synchronization, the application successfully eliminates data lag and communication gaps between students and laboratory assistants regarding progress and "ACC" status. The research shows that a component-based frontend approach combined with NoSQL cloud storage ensures a high-performance and scalable system without the need for traditional server management. Furthermore, functional testing through the Black Box method confirms that all core features, including secure authentication and real-time tracking, operate correctly. For future development, adding push notification features and a mobile-native version could further enhance user accessibility and engagement.

Acknowledgement

The authors would like to express their sincere gratitude to the Department of Computer Science, Faculty of Science and Technology, Universitas Islam Negeri Sumatera Utara (UINSU) for providing the laboratory facilities and academic support during this research. Special thanks are also extended to Mr. Mhd. Furqan for his valuable guidance, insights, and references that significantly improved the quality of this manuscript. Finally, we thank all the colleagues who participated in the testing and evaluation of the Laparak Tracker application.

References

- [1] M. R. Erzha, M. A. Halim, A. A. Santosa, M. I. Ravasia, and E. K. Pratama, "Designing a School Information System and Admission of New Students at Gunadarma Middle School Based on a Website Using the Prototype Method," *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 3, no. 3, 2024.
- [2] S. Syidada, "Analisis dan Perancangan Sistem Smart Campus Monitoring Praktikum menggunakan Gamifikasi," *JIFTI - Jurnal Ilmiah Teknologi Informasi dan Robotika*, vol. 3, no. 2, 2021.
- [3] D. Saputra, A. I. A. Sherill, and Q. M. Anindya, "Analisis UI dan UX Aplikasi Senayan Library Management System (SLIMS) dalam Pengelolaan Perpustakaan Digital," *THE LIGHT: Journal of Librarianship and Information Science*, vol. 4, no. 1, pp. 10–30, 2024.
- [4] K. Juan and S. Budi, "Pengembangan Menu Digital Menggunakan ReactJs," *Jurnal Strategi*, vol. 5, no. 1, 2023.
- [5] A. Leto, "Serverless Computing: Kajian Literatur untuk Memahami Arsitektur dan Implikasinya di Era Cloud Computing," *Integrative Perspectives of Social and Science Journal (IPSSJ)*, vol. 1, no. 1, 2024.
- [6] J. W. Jokanan, A. Widodo, N. Kholis, and L. Rakhmawati, "Rancang Bangun Alat Monitoring Daya Listrik Berbasis IoT Menggunakan Firebase Dan Aplikasi Android," *Jurnal Teknik Elektro*, vol. 11, no. 1, pp. 47–55, 2022.
- [7] A. F. Seanne, R. S. Putra, A. Wicaksono, and Farida, "Perancangan Aplikasi Pengelolaan Tugas Praktikum Laboratorium Rekayasa Perangkat Lunak Berbasis Web," in *Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi*, 2022.
- [8] J. Sari, Nurhidayati, I. W. S. Nst, and Mhd. Furqan, "Sistem Pengelolaan Pencatatan Barang di Gudang Bependasu Berbasis Web dengan Metode Waterfall," *Syntax: Journal of Software Engineering, Computer Science and Information Technology*, vol. 6, no. 1, pp. 627–635, 2025.
- [9] T. S. R. Dini, P. S. Alhirani, and Mhd. Furqan, "Perancangan Sistem Informasi Manajemen Aplikasi Pendaftaran Rekening Bank Baru Nasabah," *Neptunus: Jurnal Ilmu Komputer Dan Teknologi Informasi*, vol. 3, no. 2, pp. 1–12, 2025.
- [10] D. Oktaviani, F. S. Papiyaya, and P. F. Tanaem, "Perancangan Aplikasi E-Menu Restaurant dengan Menggunakan Cloud Computing dan Serverless Architecture Lambda," *EXPLORE: Jurnal Sistem Informasi dan Telematika*, vol. 12, no. 1, 2021.
- [11] M. J. Utama, S. N. Neyman, and K. Priandana, "Designing LMS Model Based on Serverless to Solve Performance Issue on High Concurrency Workload," *Jurnal Ilmu Komputer Agri-Informatika*, vol. 12, no. 2, pp. 142–154, 2023.
- [12] C. Chung and M. D. Lauro, "Perancangan Aplikasi Web Asisten Dosen Berbasis Design Thinking dengan Evaluasi SUS," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 10, no. 4, 2025.
- [13] J. Shafira, "Perancangan User Interface (UI) dan User Experience (UX) Website Manajemen Material Scaffolding Menggunakan Metode Pendekatan Design Thinking," Yogyakarta, 2022.
- [14] R. Andrianto and M. H. Munandar, "Aplikasi E-Commerce Penjualan Pakaian Berbasis Android Menggunakan Firebase Realtime Database," *Journal Computer Science and Information Technology (JCoInT)*, vol. 3, no. 1, pp. 20–29, 2022.
- [15] M. Nuraminudin, M. M. Dewi, S. Suharsono, A. Dahlan, and Lukman, "Implementasi ReactJS pada Pembuatan Sistem Informasi Digital Printing Berbasis Website," *Information System Journal (INFOS)*, vol. 6, no. 1, pp. 25–35, 2023.