The Design of a Digital Transformation in University Facility Maintenance

Sef Sofa Maulanaja
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
maulanaja@student.telkomuniversity.ac.id

Meta Kallista
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
metakallista@telkomuniversity.ac.id

Ashri Dinimaharawati
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ashridinimaharawati@telkomuniversity.ac.id

***Abstract* — The maintenance of academic facilities at the TULT Building of Telkom University has traditionally been carried out manually using physical media, leading to several drawbacks, including susceptibility to data loss, delayed reporting, and a lack of integrated monitoring. To address these problems, a high-performance digital monitoring system was developed to digitize the processes of recording, reporting, and scheduling maintenance activities. This paper details the comparative analysis and empirical validation of the technology stack selected for this system. The system was built using Vue.js for the frontend, Express.js for the backend, and PostgreSQL for database management. The selection process was based on a quantitative design matrix scoring methodology, evaluating alternatives for each architectural layer. The system's performance and usability were then rigorously validated through a multi-stage testing protocol, including Alpha, Beta, and Stress Testing. Alpha testing results showed that all features functioned properly with a 100% success rate. Beta testing, involving administrators, technicians, and general users, yielded an overall usability score of 4.51 out of 5, indicating excellent user acceptance. Furthermore, stress testing with up to 300 concurrent virtual users demonstrated exceptional stability, with a 0% error rate and an average response time of 224.11 ms. These results validate the chosen technology stack as a robust and efficient foundation for a high-performance facility management application, ensuring a more structured, efficient, and well-documented maintenance process.**

***Keywords* — monitoring system, digital transformation, Vue.js, Express.js, QR code**

I. INTRODUCTION

The operational backbone of any modern university relies heavily on the consistent availability and functionality of its academic facilities. Prior to this project, the maintenance process at the TULT Building of Telkom University was a legacy, paper-based operation. The system depended entirely on physical media such as "checklist sheets" and "control cards" to track and monitor maintenance activities. This manual approach, while simple in concept, created significant operational friction and was no longer adequate for a dynamic academic environment.

The reliance on manual protocols was the primary driver for this digital transformation initiative. The existing system was plagued by inherent weaknesses, including a high susceptibility to data loss, human error in record-keeping, and a lack of centralized, real-time monitoring. This resulted in significant delays between the occurrence of a facility failure, its reporting, and the eventual dispatch of a technician. The consequences of these inefficiencies were far-reaching, creating a negative impact on the university's core mission. Economically, the system promoted a costly reactive maintenance model. Operationally, it diminished the productivity of the maintenance team. Most critically, from a service perspective, malfunctioning facilities—such as broken air conditioners or faulty projectors—directly disrupted lectures, leading to diminished satisfaction among students and faculty and a tangible decline in the quality of the academic experience.

This paper details the design and impact of the digital solution developed to address these challenges. The project's goal was to execute a complete digital transformation of the facility maintenance workflow. We present the architecture of the new platform, the evidence-based methodology used to select its technology stack, and a comprehensive analysis of its impact. This impact is measured through a multi-stage testing protocol, providing empirical validation of the system's functionality, user acceptance, and performance, thereby demonstrating a successful transition from an inefficient manual process to an intelligent, data-driven platform.

II. THEORITICAL BACKGROUND

The development of the facility management system required a technology stack capable of delivering high performance, real-time responsiveness, and robust data integrity. The core technologies selected were Vue.js, Express.js, and PostgreSQL.

A. Vue.Js

A modern JavaScript framework focused on constructing user interfaces, it uses a component-based design that makes it easier to develop dynamic, sophisticated single-page applications. Its key advantages include a gentle learning curve, comprehensive documentation, and high performance due to its lightweight virtual DOM implementation. This makes it well-suited for creating responsive and efficient interfaces for website.

B. Express.Js

A lightweight, adaptable Node.js framework that delivers a comprehensive range of features for building both web and mobile apps. As a backend framework, Express.js excels in building fast and scalable backend applications. Its event-driven, non-blocking I/O model is particularly

effective handling a high volume of concurrent connections, a critical requirement for real-time monitoring systems that process numerous simultaneous data transactions.

C. PostgreSQL

A powerful, open-source object-relational database system (ORDBMS) with a strong reputation for reliability, feature robustness, and performance. It offers full compliance with ACID (Atomicity, Consistency, Isolation, Durability) principles, ensuring high data integrity for transactional systems. PostgreSQL also supports advanced data types, including native JSONB support, and sophisticated indexing, making it a versatile choice for applications that require both structured relational data and the flexibility to handle complex queries.

III. METODOLOGY

The development of the system followed a structured engineering methodology, encompassing technology selection, system architecture design, and the implementation of core algorithms. This approach ensured that each component of the solution was deliberately chosen and designed to meet the specific operational needs of the facility maintenance workflow.

A. Technology Stack Selection Process

An evidence-based approach was used to select the optimal technology for each layer of the application. A design matrix scoring method was employed to quantitatively evaluate alternatives based on predefined parameters critical to system performance and reliability.

- Frontend**:** Vue.js was compared against React.js. Vue.js was selected due to its superior performance in execution speed, greater memory efficiency, and faster startup time, which are crucial for a responsive user experience on both web and mobile platforms.
- Backend**:** Express.js was evaluated against Laravel and Django. Express.js was chosen for its lightweight architecture and the high performance of its non-blocking I/O model, making it ideal for handling the high volume of concurrent requests expected in a real-time monitoring system.
- Database Management System**:** PostgreSQL was compared with MySQL and MongoDB. PostgreSQL was selected for its superior performance in complex query scenarios, robust ACID compliance ensuring data integrity, and advanced features like JSONB support, which provided the best combination of reliability and flexibility for the system's structured data needs.

B. System Design and Architecture

The system was designed with a clear, role-based architecture to streamline interactions and digitize the entire maintenance lifecycle.

- Use Case Design: The system is built around three primary actors: Admin, Technician, and User. The Use Case Diagram clearly defines their interactions: Users report issues, Admins manage and approve tasks, and Technicians execute the maintenance work. This model creates a closed-loop digital workflow with clear responsibilities for each role.
- Operational Flowchart: Detailed flowcharts were designed for each user role to map out the step-by-step operational procedures. The user's flow involves scanning a QR code to report an issue. The admin's flow includes receiving, approving, and scheduling the task. The technician's flow covers accepting the task, performing the repair, and submitting proof of work. This ensures a logical and traceable process from problem identification to resolution.
- Database Schema (ERD): An Entity Relational Diagram (ERD) was designed to model the data structure in PostgreSQL. Key entities such as users, roles, assets, rooms, tasks, and tickets were defined with clear relationships. This well-structured schema ensures data integrity and efficiently supports the system's complex operational workflows.

IV. RESULT AND DISCUSSION

The quantitative analysis yielded clear winners for each architectural layer, justifying the final technology stack.

- Frontend**:** Vue.js was selected, scoring a total of 15 points compared to React.js's 11. It demonstrated superior performance in execution speed, greater memory efficiency, and a faster startup time, which were deemed critical for a responsive user experience.
- Backend**:** Express.js was chosen with a score of 14, outperforming Laravel (10) and Django (13). Its event-driven, non-blocking I/O architecture provided superior reliability and scalability for handling the high volume of concurrent requests expected in a real-time monitoring system.
- Database**:** PostgreSQL was the definitive choice, scoring 19 points against MySQL's 17 and MongoDB's 12. In direct performance tests, PostgreSQL excelled in handling complex queries and demonstrated superior data integrity as a mature relational database, proving that a NoSQL solution was not inherently faster for this project's structured data needs.

A. System Validation Results

The empirical testing phase provided strong validation for the positive impact of the digital transformation.

- Alpha Testing: The system achieved a *100% success rate* across all 30 test scenarios for the Admin, Technician, and User roles. All core functionalities operated as designed without critical errors, and initial performance metrics, such as page load times (under 1.5 seconds), met the non-functional requirements.
- Beta Testing: User feedback was overwhelmingly positive. The system achieved an overall average usability score of *4.51 out of 5.00,* falling into the "Very Good" category. All individual aspects, including ease of use (4.47), interface design (4.60), and efficiency (4.53), received high marks. Furthermore, all UAT scenarios were successfully completed by end-users, confirming the system's practical utility and alignment with real-world workflows.

- Stress Testing**:** The new platform demonstrated exceptional stability and performance under extreme load. When subjected to *300 concurrent virtual users*, the backend handled all 33,725 requests with a *0% error rate*. The average response time remained low at *224.11 ms,* and the system sustained a throughput of 124.62 requests per second. These results empirically validate that the Express.js backend and PostgreSQL database are capable of handling loads far exceeding normal operational use, confirming the system's robustness and scalability.

  The discussion of these results highlights a clear causal link: the rigorous, data-driven selection of a high-performance technology stack (Vue.js, Express.js, PostgreSQL) directly led to the development of a system that was not only functionally complete but also empirically validated as stable, usable, and highly performant.

V. CONCLUTION

This project successfully executed a digital transformation of the facility maintenance process at Telkom University, replacing an inefficient, error-prone manual system with an intelligent, high-performance digital platform. The design, built upon a carefully selected technology stack of Vue.js, Express.js, and PostgreSQL, has proven to be effective. The impact of this transformation was validated through comprehensive testing, which demonstrated 100% functional success, excellent user acceptance with a usability score of 4.51 out of 5, and exceptional stability under stress. While acknowledging current limitations, such as the need for more notification channels and advanced search capabilities, the project has established a powerful and scalable foundation. Future work will focus on expanding functionality to include inventory management and offline capabilities, further enhancing the system's value as a comprehensive operational management platform.

REFERENCES

[1] Priyatna, N. M. (2024). Transformasi Digital: Efisiensi dan Inovasi dalam Manajemen Operasional. *Economic Reviews Journal*, *3*(3).

[2] Axza, F., Sofi'ie, F., & Qoiriah, A. (2023). Analisis Perbandingan Framework Front-End Javascript React dan Vue Pada Pengembangan Website. *Journal of Informatics and Computer Science*, *05*.

[3] Amarulloh, A. (2023). ANALISIS PERBANDINGAN PERFORMA WEB SERVICE REST MENGGUNAKAN FRAMEWORK LARAVEL, DJANGO, DAN Node JS PADA APLIKASI BERBASIS WEBSITE. *JURNAL TEKNIK INFORMATIKA STMIK ANTAR BANGSA*.

[4] Yunanto Putra, Y., Purwaningrum, O., & Hadi Winata, R. (2022). PERBANDINGAN PERFORMA RESPON WAKTU KUERI MYSQL, POSTGRESQL, DAN MONGODB COMPARISON OF QUESTION TIME RESPONSE PERFORMANCE MYSQL, POSTGRESQL, AND MONGODB.

[5] Mone, F., & Simarmata, J. E. (2021). APLIKASI ALGORITMA GENETIKA DALAM PENJADWALAN MATA KULIAH. *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, *15*(4), 615–628.

[6] Ihsani, I., Pramuntadi, A., Gutama, D. H., & Wijaya, D. P. (2022). IMPLEMENTASI ALGORITMA GENETIKA DALAM PENENTUAN RUTE OPTIMAL UNTUK KURIR KANTOR POS BERBASIS WEB (STUDI KASUS: KANTOR POS WATES). *Indonesian Journal of Business Intelligence (IJUBI)*, *5*(2), 76.

[7] Florensia, N. P., et al. (2025). ANALISIS PENGUJIAN SISTEM PAKAR PENYAKIT SELADA MENGGUNAKAN METODE BLACK BOX & WHITE BOX TESTING. *Jurnal Informatika dan Teknik Elektro Terapan*, *13*(2).

[8] Nuantra, V. A., et al. (2022). Faktor Usability Testing Terhadap Penggunaan Presensi Di Web SIA UTY. *Jurnal Teknologi dan Manajemen Industri Terapan (JTMIT)*, *1*(3), 173–182.

[9] Yakub, H., et al. (2024). Sistem Informasi E-Commerce Berbasis Website Dengan Metode Pengujian User Acceptance Testing. *JSITIK: Jurnal Sistem Informasi dan Teknologi Informasi Komputer*, *2*(2), 113–127.

[10] Suprapto, A., & Sasongko, D. (2021). EVALUASI PERFORMA WEBSITE BERDASARKAN PENGUJIAN BEBAN DAN STRESS MENGGUNAKAN LOADIMPACT (STUDI KASUS WEBSITE IAIN SALATIGA).