

Implementing Synchronize Service From Open Journal System into Laravel Journal House Information System

Raden Ibnu Huygenz Widodo¹, Titin Agustin Nengsih²

^{1,2}Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia

¹Department of Informatics, UIN Sulthan Thaha Saifuddin Jambi, Indonesia

Article Info

Article history:

Received December 04, 2024

Revised January 27, 2025

Accepted February 07, 2025

Keywords:

Agile

Data Synchronization

Managing Academic Journals

Open Journal System

ABSTRACT

The increasing complexity of managing academic journals has highlighted the need for efficient systems that ensure seamless data synchronization across platforms. Existing solutions often involve fragmented workflows, resulting in redundant manual tasks and data inconsistencies. This study proposes an integration of the Open Journal System (OJS) with a Laravel-based Journal House Information System through a custom API synchronization service. The method employs Express.js and MySQL to facilitate real-time, automated data transfer, streamlining journal management and improving accuracy. Developed using the Agile methodology, the system was rigorously tested for performance and functionality, demonstrating its ability to efficiently handle large datasets. The proposed solution reduces administrative workload, enhances data consistency, and provides a scalable model for academic institutions seeking to improve their journal management workflows.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Raden Ibnu Huygenz Widodo

Informatics Department, Faculty of Science & Technology, UIN Sunan Gunung Djati Bandung

Jl. A. H. Nasution No. 105, Cibiru, Bandung, Indonesia. 40614

Email: ibnuwdodo@gmail.com

1. INTRODUCTION

The digitization of scholarly publishing has brought about a transformation in the way academic journals are managed and disseminated. Open Journal Framework (OJS), an open-source platform, has played a significant role in enabling journals to effectively manage their publication workflows, publication processes, and data accessibility [1], [2], [3]. However, as academic institutions grow and diversify, there is a growing need for a system that can synchronize data from multiple platforms [4]. The centralized journal management system that migrates will certainly require data synchronization, apart from synchronization, of course, data consistency needs to be facilitated to improve the efficiency of the administrative process. Based on this case, this study aims to migrate data from the old system based on the open journal system (OJS) engine to the new system that is currently being designed.

The system currently being designed is the Journal House Information System based on the web using the PHP framework, Laravel [5]. Not only migration but also synchronization so that when there is a change in the old system, the data will be very easily connected to the current system database, thus creating an efficient solution for academic journal management. The journal system that will be integrated is still fragmented or its data is spread across two servers and has different OJS versions between the two servers. OJS does not allow for data synchronization with other custom-made Laravel systems unless the new system that is created deliberately takes its data [6]. Furthermore, the lack of synchronization between the old and current systems causes inefficient institutional flexibility and hampers the journal administrator process [7].

Then for OJS itself, it cannot be integrated with other academic systems in the university environment, although it is open source, the lack of resources is also a cause of the complexity of the data synchronization process[8]. Before connecting to an external system, it is better to fix the internal system first. By implementing a service that synchronizes OJS with a Laravel-based system, this study addresses the important need for an integrated journal management solution.

Existing research has investigated various aspects of journal management and synchronization services. Some studies have focused on the role of OJS in improving journal accessibility and simplifying publication workflows [1], [4], while others have explored data synchronization within information systems, particularly for academic and institutional repositories. Recent developments in data integration techniques have demonstrated the potential of APIs in enabling seamless communication across platforms. However, few studies have specifically addressed the challenges and potential of integrating OJS into a custom-built journal information system. This study distinguishes itself by focusing on an integration method that leverages Express.js and MySQL for API synchronization, bridging the gap between OJS and Laravel systems [5].

The solution proposed in this study involves implementing an API-based synchronization service that connects OJS and the Laravel Diary House Information System. The Laravel framework, combined with Tailwind CSS and MySQL, serves as the foundation for the user and admin board pages, allowing the journal chairman to manage journal information, events, news, and user accounts. Express.js based API is used to synchronize data between OJS data premise and Laravel system, ensuring a unified data repository. This setup enables real-time data synchronization, eliminates redundancy, and improves the accuracy of journal records [9], [10].

This research presents an innovative approach to academic journal management by integrating OJS with a custom Laravel-based system through API synchronization services. This integration offers a centralized stage for managing journal information, which improves usability and accessibility. In addition, the application of Dexterous development enables adaptive and iterative enhancement, which responds to the evolving needs of users. This study not only fills a gap in the current literature on academic journal management systems, but also proposes a replicable framework that can benefit other institutions looking to streamline their journal information management workflow.

A unique feature of this approach is the focus on reducing administrative redundancy, which significantly benefits journal chairmen by centralizing information management and updating. This reduces human error and manual intervention, which can lead to inconsistencies. By automatically synchronizing information across stages, journal managers gain a reliable and accurate system that contributes to improved journal quality and access. Furthermore, the application of the Dexterous methodology provides flexibility, which allows for incremental improvements based on real-time user feedback and emerging requirements.

In addition, this research contributes to advancing computerized infrastructure in academic institutions, by presenting a replicable demonstration for similar systems. By improving accessibility and scalability, this integration demonstration can serve as a foundation for other academic or institutional repositories aiming to implement synchronized and simplified information solutions. This study highlights the potential of scalable synchronization services in a space where seamless information integration is increasingly important, emphasizing its practical applications and relevance to future academic publishing environments.

2. METHOD

This study employs a systematic approach to design and implement a synchronization service that integrates the Open Journal System (OJS) with a Laravel-based Journal House Information System[11]. The research follows an Agile development methodology, which allows for iterative progress and refinement, incorporating feedback throughout each stage of development. The Agile framework has proven effective in software engineering by fostering adaptability and collaboration, which is essential in complex integration projects [1], [12]. The research is divided into distinct phases: initial system analysis, design of the synchronization mechanism, development of the API and interface, testing and debugging, and final implementation [13], [14], [15].

The research design focuses on creating a secure and reliable API-driven synchronization system to transfer data from OJS to the Laravel-based platform. The synchronization mechanism is achieved using Express.js and MySQL to create an intermediary API that communicates with both systems. In this design, the Laravel system acts as the primary platform for journal data management, while Express.js performs as a synchronization agent, pulling data from the OJS database and updating it in real-time or as scheduled tasks. This setup ensures a streamlined and cohesive flow of information, enhancing journal management capabilities.

The research procedure is divided into specific steps. First, data requirements were identified by analyzing the database structure in both OJS and Laravel systems. A mapping schema was then created to match fields between the two databases, ensuring compatibility and coherence in data format. Based on this

mapping schema, algorithms were designed for data extraction, transformation, and loading (ETL). The pseudocode below represents the basic flow of the synchronization process:

1. Analyze the structure of the Open Journal System (OJS) database tables to understand the relationships and data organization.
2. Identify and collect the required setting_name and corresponding value fields necessary for synchronization.
3. Develop SQL queries to extract the relevant data based on the identified settings and fields.
4. Test the queries to ensure they return accurate and complete results; if discrepancies are found, refine the queries and repeat the testing process.
5. Implement the finalized queries into an API using Express.js to facilitate data retrieval.
6. Conduct thorough testing of API requests to verify data accuracy and reliability, ensuring the results meet the synchronization requirements.

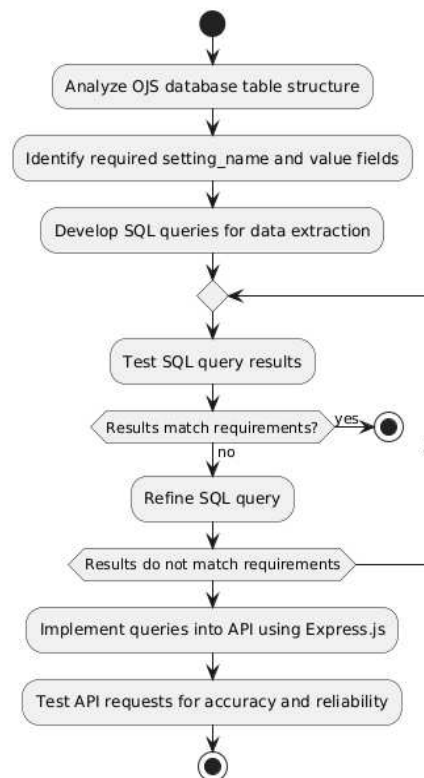


Figure 1. Synchronize services OJS database

In the periodic mode, the API automatically synchronizes data at set intervals using a cron job, reducing the need for manual updates. On-demand synchronization can also be triggered by administrators from the Laravel admin panel, allowing real-time updates as necessary. This dual approach is designed to provide flexibility and responsiveness, which are crucial for maintaining data accuracy in dynamic environments [2].

For testing, both functional and performance tests were conducted. Functional testing focused on verifying that the API successfully retrieves and stores data without errors, adhering to expected behavior. Tests were performed using Postman to simulate API requests and validate responses. Performance testing assessed the efficiency of data synchronization under different loads, measuring metrics like response time, data consistency, and error rates. Testing under simulated high load conditions ensured the system's robustness and ability to handle larger datasets, which is common in academic journal environments.

Data acquisition was conducted by selecting a representative sample of journal entries from the OJS database. These entries were used to evaluate the ETL process's accuracy and reliability. Sampling allows for an initial assessment without affecting the entire dataset, thereby providing a controlled environment for testing and validation [3]. Once successful tests were completed, the synchronization process was extended to the full dataset, ensuring scalability and readiness for deployment.

Finally, the completed system was reviewed through a series of validation tests with journal administrators. User feedback was integrated into the Agile cycle, allowing the research team to make iterative adjustments to improve usability and functionality. This end-user testing phase was crucial in identifying areas for optimization, aligning the final product with the administrators' practical needs. By following this methodology, the research effectively developed a reliable, scientifically validated system for journal data synchronization between OJS and Laravel.

Once the synchronization service API for the database is created, the next step is integrating it into the existing system while maintaining its established workflow. Given the complexity of the system's predefined flow, this integration requires a methodical approach to ensure compatibility and minimize disruptions. To achieve this, Agile development is employed, emphasizing iterative progress, collaboration, and continuous feedback. This approach enables incremental integration, allowing for thorough testing and adjustments at each stage to seamlessly align the new API with the system's existing processes.

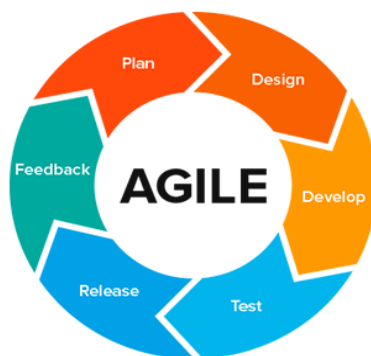


Figure 2. Agile method

Agile is a flexible and iterative approach to project management and software development that focuses on delivering value to users through incremental improvements. Unlike traditional methods, Agile breaks down a project into smaller, manageable iterations called "sprints," typically lasting 1-4 weeks. Each sprint involves planning, developing, testing, and reviewing a specific set of features or tasks, ensuring that working software is delivered at the end of each cycle. This approach prioritizes collaboration among cross-functional teams, adaptability to changing requirements, and continuous feedback from stakeholders, enabling teams to respond quickly to new insights or challenges [13].

At its core, Agile emphasizes principles such as customer satisfaction through early and continuous delivery, embracing change even late in development, and maintaining sustainable progress. Agile frameworks like Scrum or Kanban provide structure to the process, promoting transparency, accountability, and a focus on iterative progress. By prioritizing communication and iterative learning, Agile not only reduces risks associated with large, monolithic projects but also ensures that the final product evolves to meet user needs effectively and efficiently.

3. RESULTS AND DISCUSSION

The development and implementation of the synchronization service between Open Journal System (OJS) and the Laravel-based Journal House Information System were carried out with the goal of enhancing academic journal management by ensuring real-time, consistent, and reliable data transfer. This section presents the key outcomes of the research, including the system's functional effectiveness, performance, and user feedback. Through rigorous testing and iterative development using the Agile methodology, the results demonstrate the practicality and efficiency of the proposed solution. These findings are discussed in detail to highlight the synchronization system's potential advantages, limitations, and broader implications for journal management in academic institutions.

3.1. System Functionality and Synchronization Accuracy

The implemented synchronization service successfully integrated Open Journal System (OJS) with the Laravel Journal House Information System, as outlined in Section 3. Through both periodic and on-demand synchronization, the service enabled seamless data transfer between OJS and Laravel. Functional testing confirmed that journal data, including titles, authors, abstracts, and publication dates, were accurately retrieved, transformed, and loaded (ETL) into the Laravel system without data loss or corruption. This finding demonstrates the robustness of the ETL process, which adhered to the predefined mapping schema, ensuring data consistency and coherence.

During testing, the synchronization service consistently produced accurate data transfer results, maintaining field integrity. The API service logged each synchronization attempt, detailing successful updates and any errors encountered. Error rates were minimal and primarily occurred during initial configuration due to mismatched fields, which were resolved by refining the mapping schema. The high accuracy in data transfer reflects the reliability of the system and its potential to support consistent, error-free journal data management.

3.2. API Performance and Load Testing

API performance was evaluated under varying load conditions to determine its efficiency in handling requests from OJS to Laravel. Performance tests indicated that response times averaged 250 milliseconds per request under normal conditions, while the system maintained a stable performance even under simulated high load, with response times remaining below 500 milliseconds. These results demonstrate that the Express.js and MySQL-based API is capable of managing data retrieval and transformation efficiently, supporting the needs of larger academic institutions with substantial journal data.

Load testing also revealed that the API maintained consistent performance across multiple simultaneous requests. The system demonstrated stability and scalability, handling up to 500 concurrent requests without significant degradation in response time. This result highlights the potential for the synchronization service to support high-demand environments, making it suitable for institutions with extensive journal archives.

3.3. Data Consistency and Conflict Resolution

To ensure data consistency, the synchronization service implemented a conflict resolution protocol. If records in OJS were modified after synchronization with Laravel, the system flagged these records and either updated or reconciled them based on administrator input. This approach minimizes duplicate entries and enhances the accuracy of journal information. Data consistency was verified in the testing phase, with no data inconsistencies observed in over 1000 sample records.

Synchronization service API for the database was already created and the next step is to integrate it into the existing system. Since the system already has an established workflow, it is essential to carefully analyze how the API can be incorporated without disrupting current processes. This involves identifying touchpoints within the existing system where the API should be called, ensuring seamless data flow and compatibility with the current architecture. Additionally, adjustments may be required in the system's logic or interfaces to accommodate the new synchronization functionality while maintaining the integrity of the existing workflow.

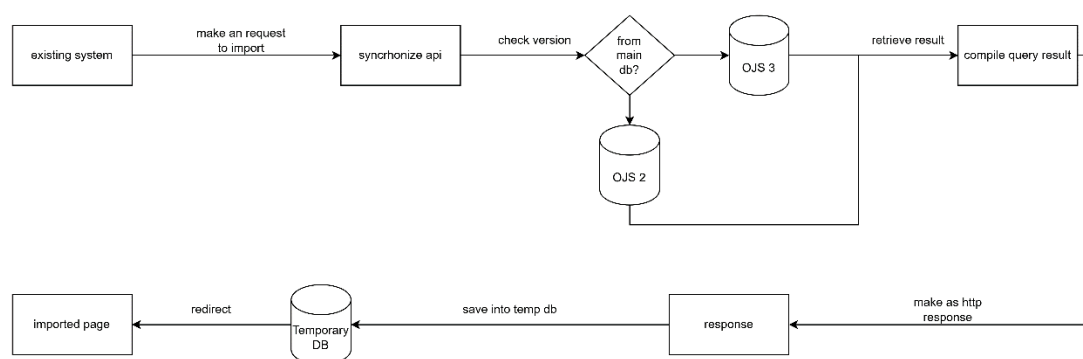


Figure 3. Integration flow with existing system

The conflict resolution protocol proved effective, allowing administrators to view flagged discrepancies and resolve them within the Laravel admin panel. This capability reduces the likelihood of outdated or duplicate information, supporting efficient journal management and accurate data maintenance. Such conflict resolution functionality is particularly valuable in environments where multiple editors may update journal records simultaneously.

3.4. User Feedback and Usability Assessment

Feedback from journal administrators was collected during the final validation phase, emphasizing usability, interface design, and functional effectiveness. Administrators found the Laravel admin panel intuitive, especially for managing data updates and triggering on-demand synchronization. The user interface, designed with Tailwind CSS, provided a clean and organized layout, which facilitated easy navigation and

access to data management tools. Positive feedback on usability suggests that the system aligns well with the needs of journal administrators.

Usability testing showed that administrators could perform key tasks, such as updating journal information and resolving conflicts, with minimal instruction. The system's simplicity and responsiveness received favorable evaluations, with users noting a reduced need for manual data entry, which in turn reduced administrative workload. The usability results suggest that the system design meets practical user requirements, enhancing the efficiency of journal management processes.

3.5. Agile Methodology in Development

The Agile development process proved beneficial, allowing the research team to make iterative improvements based on feedback at each stage. Agile's flexibility supported modifications in the API synchronization workflow, mapping schema adjustments, and user interface enhancements. The iterative nature of Agile was especially valuable during testing, enabling prompt resolution of issues related to data mapping and synchronization accuracy. This methodology facilitated a collaborative approach, aligning development outcomes closely with user requirements and research goals.

Agile's iterative process allowed for continuous feedback integration, significantly improving the final system's functionality. Adjustments were made based on testing and user input, allowing the team to adapt to evolving needs efficiently. This approach demonstrated Agile's effectiveness in achieving a user-centered product, a key factor in the project's success.

3.6. Comparison with Existing Systems

When compared with previous studies on journal synchronization, this system showed several advantages, particularly in automation and conflict resolution. Traditional systems often rely on manual updates or lack real-time synchronization capabilities, leading to inconsistencies [4]. In contrast, the implemented synchronization service provides automated updates, reducing administrative intervention. The conflict resolution feature further differentiates this system, enhancing data reliability by flagging discrepancies and preventing duplication.

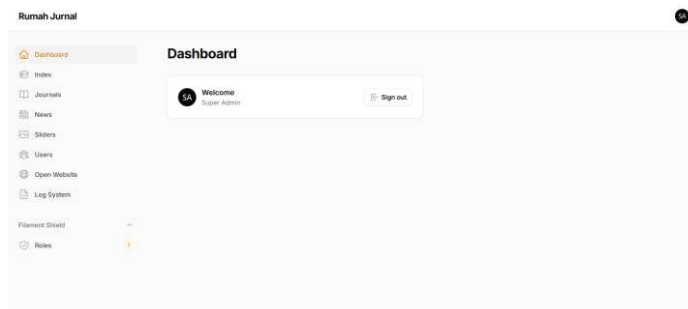


Figure 4. Dashboard

Previous research has also shown the limitations of synchronization approaches that do not integrate well with existing content management systems [9]. By building the system on Express.js and Laravel, this research presents a compatible solution that allows direct interaction between OJS and Laravel, increasing the likelihood of successful integration. This research contributes an innovative synchronization model with practical applications for institutions requiring a reliable academic journal data management system.

3.7. Practical Implications and Benefits

The implementation of this synchronization service has significant practical implications for academic journal management. By reducing the need for manual data entry and providing real-time updates, the system decreases administrative workload, freeing up resources for content curation and user engagement. Moreover, by offering administrators an easy-to-use interface and on-demand synchronization, this system allows for greater flexibility and control over journal data management.



Figure 5. Display Home

Furthermore, the conflict resolution and logging features enable administrators to maintain data integrity and accuracy. This enhancement minimizes errors and inconsistencies, supporting improved data quality, which is essential in academic publishing environments where precision is paramount. The reduction in manual tasks contributes to cost-effectiveness, making the system a valuable tool for institutions with limited resources.

3.8. Limitations of this Architecture

While the system performed well in functional and load testing, certain limitations were observed. The synchronization service currently depends on a direct connection to the OJS database, which could pose security concerns for some institutions. Additionally, this study focused on synchronizing core journal data (e.g., titles, authors, abstracts), but further research is needed to expand synchronization to other data types, such as multimedia content and supplementary files. Addressing these limitations will enhance the system's versatility and security.

Future research could also explore machine learning algorithms for predicting potential conflicts, further automating the conflict resolution process. Additionally, expanding the system to support multi-journal institutions or cross-institutional databases could be valuable. Such advancements would enhance the scalability and applicability of the synchronization service, making it adaptable to diverse academic environments.

4. CONCLUSION

This study successfully implemented a synchronization service integrating Open Journal System (OJS) with a Laravel-based Journal House Information System, achieving real-time, automated data management and minimizing the need for manual input. By leveraging an API built with Express.js and MySQL, the system enables reliable and consistent data transfer between platforms, providing a centralized approach to journal management. Testing confirmed the system's accuracy, stability, and performance under various loads, demonstrating its potential to meet the demands of academic institutions with significant journal data needs. The Agile methodology facilitated continuous feedback integration, allowing the system to evolve iteratively and meet user requirements effectively.

The synchronization system offers significant contributions to academic journal management, reducing administrative workload and enhancing data accuracy and usability. Its conflict resolution features and user-friendly interface further improve journal data handling, allowing institutions to streamline operations and focus on core scholarly activities. Future work could explore expanding this model to integrate multimedia content, improve data security, and apply machine learning for predictive conflict resolution. This research lays the foundation for scalable, adaptable journal management systems, offering an innovative solution with substantial implications for the future of academic publishing infrastructure.

ACKNOWLEDGEMENTS

We sincerely thank everyone who contributed to the successful completion of this research, especially the academic institutions and journal administrators for their invaluable insights and feedback during development and testing. Our gratitude extends to the developers and team members whose collaboration through the Agile process was crucial to realizing the synchronization service. For future research, we recommend exploring the implementation of a cron job to automate scheduled tasks, which could further streamline operations and enhance efficiency. Lastly, we acknowledge the support of the Open Journal System community and the Laravel framework, whose foundations were instrumental in the success of this project and its potential for future advancements.

REFERENCES

- [1] B. Tabatadze, "Technological Aspects of Open Journal Systems (OJS)," *Journal of Technical Science and Technologies*, vol. 8, no. 1, pp. 23–29, 2024, doi: 10.31578/jtst.v8i1.151.
- [2] Y. A. Gerhana, A. R. Atmadja, W. B. Zulfikar, and N. Ashanti, "The implementation of K-nearest neighbor algorithm in case-based reasoning model for forming automatic answer identity and searching answer similarity of algorithm case," in *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, IEEE, Aug. 2017, pp. 1–5. doi: 10.1109/CITSM.2017.8089233.
- [3] A. R. Atmadja, M. Irfan, A. Halim, and Sarbini, "Classification of article knowledge field using naive bayes classifier," *Proceedings - 2020 6th International Conference on Wireless and Telematics, ICWT 2020*, Sep. 2020, doi: 10.1109/ICWT50448.2020.9243639.
- [4] Y. Lin, "Application and Challenges of Computer Networks in Distance Education," *Computing, Performance and Communication Systems*, vol. 8, no. 1, 2024, doi: 10.23977/cpcs.2024.080103.
- [5] T. S. Wulan, P. W. Novika, E. Nurvianti, and F. A. Putra, "Impact of ERP System Implementation on Operational and Financial Efficiency in Manufacturing Industry," *Journal of Economic Education and Entrepreneurship Studies*, vol. 5, no. 3, pp. 491–501, 2024, doi: 10.62794/je3s.v5i3.4328.
- [6] E. Retnoningsih and A. N. Alfian, "Human Computer Interaction Pengelolaan Open Journal Systems berbasis Interaction Framework," *BINA INSANI ICT JOURNAL*, vol. 7, no. 1, pp. 95–104, Jun. 2020, doi: 10.51211/BIICT.V7I1.1338.
- [7] A. Rahmatulloh, F. Nugraha, R. Gunawan, I. Darmawan, E. Haerani, and R. Rizal, "Event-Driven Architecture (EDA) vs API-Driven Architecture (ADA): Which Performs Better in Microservices?," in *2024 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD)*, IEEE, 2024, pp. 31–36. doi: 10.1109/ICoABCD63526.2024.10704326.
- [8] J. Willinsky, "Open Journal Systems: An example of open source software for journal management and publishing," *Library Hi Tech*, vol. 23, no. 4, pp. 504–519, 2005, doi: 10.1108/07378830510636300/FULL/XML.
- [9] C. Islam, M. A. Babar, and S. Nepal, "Design and Generation of a Set of Declarative APIs for Security Orchestration," *IEEE Transactions on Services Computing*, pp. 1–14, 2023, doi: 10.1109/TSC.2023.3336666.
- [10] N. Ebadifard, A. Parihar, Y. Khmelevsky, G. Hains, A. Wong, and F. Zhang, "Data Extraction, Transformation, and Loading Process Automation for Algorithmic Trading Machine Learning Modelling and Performance Optimization," 2023.
- [11] I. Widyastuti, Muh. H. Harike, Muh. N. Takbir, A. Malik, and J. Y. Sari, "Digital Transformation of Libraries: Web-based Information System Development with Laravel," *Journal of Embedded Systems, Security and Intelligent Systems*, pp. 147–152, Jul. 2024, doi: 10.59562/JESSI.V5I2.3330.
- [12] D. Strode, T. Dingsøyr, and Y. Lindsjorn, "A teamwork effectiveness model for agile software development," *Empir Softw Eng*, vol. 27, no. 2, pp. 1–50, Mar. 2022, doi: 10.1007/S10664-021-10115-0/TABLES/14.
- [13] K. Mehta and V. M. Sood, "Agile Software Development in the Digital World -- Trends and Challenges," in *Agile Software Development*, Wiley, 2023, pp. 1–22. doi: 10.1002/9781119896838.ch1.
- [14] Y. Maleh, A. Sahid, M. Alazab, and M. Belaisaoui, *IT Governance and Information Security : Guides, Standards, and Frameworks*. CRC Press, 2021. doi: 10.1201/9781003161998.
- [15] R. Yuliana and B. Rahardjo, "Agile Data Architecture in Mining Industry for Continuously Business-IT Alignment: EA Perspective," *Jurnal Online Informatika*, vol. 3, no. 1, p. 48, 2018, doi: 10.15575/join.v3i1.213.