Comprehensive Framework for Transitioning Monolithic to Microservices in MVC Context

Daniel Yeri Kristiyanto
Dept. of Informatics
Institute Teknologi Sepuluh Nopember
Surabaya, Indonesia
7025232003@student.its.ac.id

Riyanarto Sarno
Dept. of Informatics
Institute Teknologi Sepuluh Nopember
Surabaya, Indonesia
riyanarto@its.ac.id

Kelly Rossa Sungkono
Dept. of Informatics
Institute Teknologi Sepuluh Nopember
Surabaya, Indonesia
kelly@its.ac.id

Rini Handayani
Dept. of Informatics
Institute Teknologi Sepuluh Nopember
Surabaya, Indonesia
7025232001@student.its.ac.id

Anjani Shepard
Dept. of Digital Business
Ilearning Incorporation
Bucaramanga, Colombia
anjani.s8@ilearning.ee

2024 3rd International Conference on Creative Communication and Innovative Technology (ICCIT) | 979-8-3503-6749-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCIT62134.2024.10701144

***Abstract*—This systematic literature review explores the transition from monolithic architectures to Model-View-Controller (MVC)-based microservices within web frameworks, addressing a significant gap in existing research. By synthesizing findings from academic databases such as IEEE Xplore, Crossref, Google Scholar, OpenAlex, and Scopus, the study focuses on methodologies, metrics, and challenges inherent in this migration. Key metrics like coupling (afferent, efferent, and generic), Lines of Code (LOC), Class Hierarchy Metrics (CHM), and system performance (execution time, CPU usage) are analyzed. The introduction of the Monolith to Microservices Decomposition Framework (M2MDF) highlights essential phases of decomposition, addressing the integration of static, dynamic, and evolutionary data. Results indicate that despite extensive research on microservices, few studies target MVC-based migration, identifying a significant research gap. The practical implications of these findings are substantial, providing a structured approach to aid practitioners in the migration process, thereby enhancing scalability, maintainability, and alignment with business objectives. Future research directions include enhancing decomposition techniques, developing robust tool support, and establishing standardized evaluation metrics to aid practitioners in effectively transitioning to microservice architectures, thereby improving scalability and system modularity.**

***Keywords*—Monolith to Microservice Migration, MVC, SLR**

I. Introduction

In recent years, the rapid development of technology has significantly impacted sectors such as education, healthcare, and commerce [1], [2]. The International Telecommunication Union (ITU) reported that the number of internet users worldwide reached 4.9 billion in 2021, representing 63% of the global population. This surge in internet usage underscores the growing importance of digital literacy and access to technology. The COVID-19 pandemic further accelerated the adoption of digital tools, leading to a dramatic shift towards online learning and remote work [3], [4]. Approximately 85% of businesses reported accelerating the implementation of technologies for employee collaboration and interaction, highlighting the urgency of addressing the digital divide and ensuring digital literacy for all [5], [6], [7].

Monolithic software architecture, **despite** its fundamental role, often struggles with scalability, flexibility, and maintainability. Transitioning to MVC-based microservices architecture offers a viable solution, facilitated by the Monolith to Microservices Decomposition Framework (M2MDF). This framework, encompassing six phases of data gathering, monolith analysis, microservices identification, optimization, evaluation, and deployment addresses both technical and business challenges. The transition **enhances** software flexibility and scalability but also necessitates changes in project management [8], team collaboration, and product development strategies [9], [10], [11].

A. Decision support for migration from monolithic to MVC-based microservices

Fig. 1 shows the transition from Monolithic to Microservices architectures from 2014 to 2024. Monolithic architecture adoption dropped from 90% in 2014 to 50% in 2019, with a projected decline below 20% by 2024. Microservices adoption increased from 10% in 2014 to 50% in 2019 and is expected to reach 80% by 2024 [1], [12]. However, MVC Microservices adoption has remained consistently low, below 10% throughout this period.

Decomposing monolith-based systems into microservices poses significant challenges, often involving domain analysis, source code examination, execution traces, and version information. While important work exists, literature typically focuses on specific scenarios, domains, or programming languages [13], [5], [14], [15].

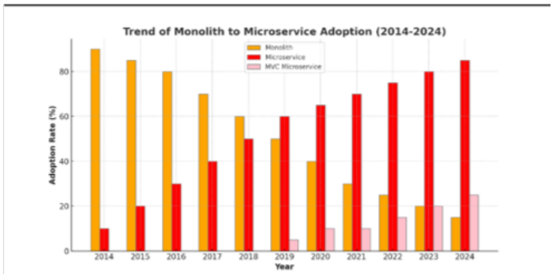

Fig. 1. Monolith Migration Trends to MVC-Based Microservices

This research introduces a framework to classify key methods for decomposition projects [1], [16]. Using a Systematic Literature Review (SLR) and snowballing, we examined 35 articles on decomposing monolith applications into microservices [17]. From 1530 initial search results, the selection criteria ensured relevant, high-quality studies were included: (1) publication date within the last five years, (2) peer-reviewed journals or conferences, (3) focus on microservices migration and related challenges, and (4) availability of full-text articles.

The narrowing process involved several stages [18]. First, titles and abstracts of 1530 papers were reviewed to exclude those unrelated to microservices or published before the specified timeframe, reducing the number to 350 [19]. Next, a detailed review of these papers focused on methodology, findings, and relevance to the research question, reducing the count to 80 [20], [9]. Finally, a full-text review evaluated the depth of analysis and contribution to the field, leading to the final selection of 33 papers [21].

B. *Existing Literature Reviews Monolith to MVC Microservice Architecture*

Recent secondary research on decomposing monolithic applications into microservices explores migration rationales, challenges, and support strategies, focusing on DevOps, development tools, techniques, and granularity [8], [22]. Key challenges include tool selection, team reorganization, migration decisions, microservices identification, and database consistency. Drivers for adopting microservices are optimized scalability, independent deployment, easier maintenance, team autonomy, loosely coupled services, technology flexibility, enhanced infrastructure, agility, reuse, and reduced development time in table I. The modernization process involves initiation, planning, execution, and monitoring. Techniques like Model-Driven (MD), Static Analysis (SA), and Dynamic Analysis (DA) are evaluated for object-oriented web applications. **Gaps** in linking monolithic decomposition with the Model-View-Controller (MVC) concept suggest MVC can aid modularization by defining service boundaries.

TABLE I
RESULTS OF EVIDENCE FINDINGS BASED ON THE REPOSITORY DATABASE

Database	**Research Question(s)**	**Main Topics**	**Paper**
IEEE Xplore, Google Scholar	What are the common challenges faced by organizations when transitioning from monolithic architectures to web MVC-based microservices?	Migration: Problems & Challenges	[1], [16], [14]
IEEE Xplore, Google Scholar	What are the key phases and elements involved in the decomposition process from MVC monolithic architectures to MVC microservices, as identified by existing literature?	Migration: Problems & Challenges	[11], [23], [24]
IEEE Xplore, Crossref	What are the existing approaches, tools, and methods utilized for decomposing MVC monolithic architectures into microservices?	Microservice Architecture	[25], [13]
IEEE Xplore, Google Scholar, OpenAlex	How do practitioners evaluate the success of monolith to microservices migration projects, and what metrics and datasets are commonly used for evaluation?	Migration: Problems & Challenges	[20]
IEEE Xplore, Crossref, Google Scholar	What are the limitations and gaps in current methodologies for decomposing monolithic architectures into microservices?	Migration: Problems & Challenges	[9]
IEEE Xplore, Crossref, Google Scholar	How do practitioners incorporate static, dynamic, and evolutionary data into the decomposition process, and what challenges arise from this integration?	Migration: Microservice Extraction	[4], [18]
IEEE Xplore, Crossref, Google Scholar	What are the current capabilities of the research community in terms of tool support for monolith decomposition into microservices, and what are the areas requiring further development or standardization?	Migration: Problems & Challenges	[17].

II. RESEARCH METHOD

The research employs a Systematic Literature Review (SLR) to examine the shift from monolithic architectures to web MVC-based microservices. Five reputable academic databases were systematically reviewed. Papers were selected based on their focus on approaches, frameworks, tools, or methodologies for migrating to web MVC microservices, excluding purely theoretical works. The analysis identified key insights, themes, challenges, and methodologies, leading to the development of the Monolith to Microservices Decomposition Framework (M2MDF) in Table II. The search terms used included "monolithic microservices," "monolithic modernization," "microservices migration," "microservices challenges," and ”MVC monolithic microservices,” covering publications onwards in IEEE Xplore, Crossref, OpenAlex, Scopus, and Google Scholar.

TABLE II
REPOSITORIES DATABASE LOGIC

Repositories Database	**Logic**
IEEE Xplore	[”Search Key=*monolith, AND migration* OR [”Search Key=*microservices AND migration* OR [”Search Key=*MVC monolith* OR [”Search Key=*MVC Microservices* OR [”Microservice challenges”]
Crossref	[”Search Key=*monolith, AND migration* OR [”Search Key=*microservices AND migration* OR [”Search Key=*MVC monolith AND challenge transformation* OR [”Search Key=*MVC Microservices AND move* OR [”advantages” AND ”disadvantages”]
OpenAlex	”MVC monolith transformation” OR ”monolith MVC migration” OR ”MVC Microservice transformation” OR ”Microservice transformation” OR ”Microservice challenges”
Scopus/Science Direct	[”Search Key=*monolith, AND migration* OR [”Search Key=*microservices AND migration* OR [”Search Key=*MVC monolith AND challenge transformation* OR [”Search Key=*MVC, Microservices AND move*
Google Scholar	”MVC monolith transformation” OR ”monolith MVC migration” OR ”MVC Microservice transformation” OR ”Microservice transformation” OR ”advantages” AND ”disadvantages”
PubMed	Title ... AND...OR Title...

A. Execution Method

Database repositories like IEEE Xplore, Crossref, OpenAlex, Scopus, and Google Scholar support research on transitioning from monolithic to microservices architectures, especially within the MVC framework. These platforms offer search terms like ”monolith AND migration” and ”microservices AND migration,” focusing on MVC transformation challenges and benefits. This helps researchers explore migration nuances, emphasizing the importance of precise research questions for an effective Systematic Literature Review (SLR). Despite progress in understanding microservices migration, gaps remain in Fig 2.

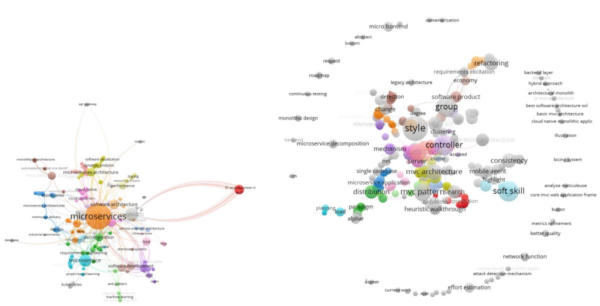

Fig. 2. Bibliometrics IEEE Xplore VS Google Sholar

In this context, we have identified four such research questions:

- RQ1: What are the key stages involved in decomposing monolithic applications into microservices, and what are the main components of each stage?
- RQ2: What current techniques, tools, and methods are utilized in the process of breaking down monolithic applications into microservices?
- RQ3: What metrics, datasets, and benchmarks are employed to assess and validate the decomposition of monolithic applications into microservices?
- RQ4: What are the identified gaps in the existing research literature on monolith-to-microservices decomposition?

This **research** consolidates innovations and introduces a framework to classify key methods for decomposition projects. Using a Systematic Literature Review (SLR) and snowballing, we identified and organized research on decomposing monolith applications into microservices. **From 1530 search** results, 35 research articles were ultimately examined. Data was gathered from IEEE Xplore, Scopus, Crossref, OpenAlex, Google Scholar, and PubMed using the Publish or Perish tool. In IEEE Xplore, 307 papers addressed ”microservice” or ”monolith,” with only 10 relevant to MVC-based migration. Scopus yielded no relevant papers, highlighting a research gap. OpenAlex produced 23 related papers, with 5 deemed relevant. PubMed had no pertinent findings. Crossref returned 1000 papers, with 9 meeting the relevance criteria. Google Scholar identified 200 papers on the transition from monolith to microservices, with 7 relevant to MVC-based migration and 2 SLR.

B. Refinement

The exclusion of studies followed general criteria refined through a multi-step filtering process. The search queried various repositories with specific keywords, yielding 1530 results. Refinement Step 1 involved reviewing titles and reducing the pool to 968 studies. If titles were insufficient, studies advanced to Step 2, where abstracts and conclusions were reviewed, narrowing the selection to 126 studies for full-text review, initially handled by the first author. In Step 3, full-text reviews against inclusion criteria resulted in 92 studies. Further refinement by group-wise filtering led to 33 studies in Step 4, with team reviews focusing on methodology, experiment, and evaluation.

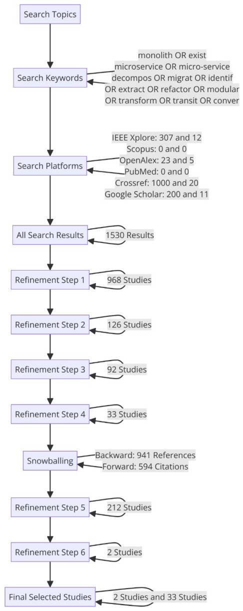

Fig. 3. Refinement Steps and Search

The literature review gathered additional references, yielding 941 backward and 594 forward citations. After removing duplicates and incomplete records, 212 studies were identified. Step 5 compared these with the initial 968, adding 2 more studies. The final selection included 2 primary studies and 33 additional studies, ensuring a thorough, high-quality set. Exclusion criteria included duplicate studies, books, patents, non-peer-reviewed studies, secondary studies, non-English studies, and pre-2014 publications in Fig. 3.

C. Analysis of the Data

The provided diagram illustrates a systematic process for extracting data from selected literature in Fig 4. The process begins with Step 1, where Ground Theory is applied to the selected studies (full text). In Step 2, obvious and inviolable elements are filtered out to identify key components. Step 3 involves aggregating these elements to form the initial M2MDF (Meta-Model Data Framework).

Following the aggregation, the process enters Step 4, an iterative refinement phase. This phase ensures the M2MDF aligns with the desired criteria. If the M2MDF fits the requirements, the data is extracted into the final literature dataset. If the M2MDF does not fit, further refinement is conducted (M2MDF Refine) until the framework meets the necessary standards. This iterative process emphasizes a meticulous approach, ensuring that only relevant and significant data elements are retained and aggregated. The refinement loop underscores the importance of continuous evaluation and adjustment to achieve an accurate and comprehensive data extraction. The flowchart highlights the importance of a structured methodology in literature data extraction, involving initial theoretical application, rigorous filtering and aggregation, and iterative refinement. This approach aims to produce a robust and reliable dataset for subsequent analysis, ensuring the extracted data is both comprehensive and precise.

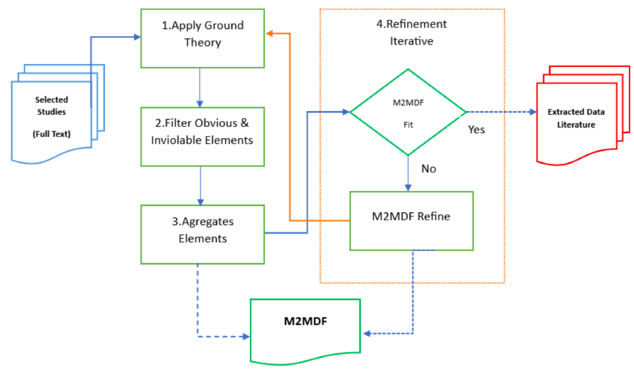

Fig. 4. Data Analysis Steps

RQ1: Apply Grounded Theory: The M2MDF analysis was primarily based on memoing, coding, constant comparison, and theoretical sorting of the selected literature, following the principles of grounded theory (GT). Due to the diverse, informal, and extensive nature of the initial memos and codes, listing them in the replication package would be impractical and of limited use to the reader. Furthermore, to preserve the true essence of the GT approach, readers replicating the method should not be guided or constrained by our codes. Instead, we provide a snippet of a selected paper with the associated memos and codes to illustrate our approach. It is important to note that given the nature of the study, a full GT process, including theoretical sampling and saturation, would not be appropriate when applied to a predefined dataset.

Filter Obvious and Inviolable Elements: Certain migration elements were clear and essential, even if they had not previously been assembled into a comprehensive framework. For instance, the collection of monolith data, analysis of that data, and identification of the resultant microservices are fundamental steps in the migration process. Further subclassification of these migration phases was required. Some of these subclasses were already explicitly identified in previous studies, such as categories for data collection and analysis.

Aggregates Elements: The research focused on integrating existing, explicitly identified, and established categories into an end-to-end decomposition framework. This process combined elements of GT with a light form of quantitative content analysis in a mixed-method approach. In studies where information was not explicitly presented, we synthesized the appropriate phases and subcategories based on GT, employing an analytical lens to organize phases and approaches. For studies with explicitly identified phases and subcategories, we performed a light content analysis to note and quantify these elements.

Refinement Iterative After each full-text review, the information was analyzed, compared, and categorized. New categories were assigned when no existing category was suitable.

The framework was revised incrementally to incorporate new categories and refine existing ones.

To answer RQ2, RQ3, and RQ4, a data-extraction template was developed based on the emerging M2MDF and the specific research questions. This template was applied to individual sets of papers, and the results were discussed in review meetings with the wider group. While the initial data analysis was performed by the first author, the outputs were reviewed by all authors. Full-text versions of all studies were shared, allowing for accuracy assessment by all authors. During the data synthesis stage, the collected data was compared, organized, and re-organized to address the research questions. This process was revisited each time a paper was reviewed. After multiple revisions during the literature review meetings, the final analysis results were achieved.

III. Result and Discussion

This study analyzes the transition from monolithic architecture to microservices-based Model-View-Controller (MVC) architecture using a systematic literature review (SLR) methodology. The results show that the adoption of monolithic architecture significantly decreased from 90% in 2014 to 50% in 2019, and it is projected to decline further to below 20% by 2024. Conversely, the adoption of microservices increased from 10% in 2014 to 50% in 2019 and is expected to reach 80% by 2024, although the adoption of MVC-based microservices remains low, below 10% over the same period. The decomposition process involves identifying microservices candidates through domain analysis, source code examination, execution traces, and version information. However, existing literature largely focuses on specific scenarios, domains, or programming languages, neglecting a more general approach in the MVC context in **Table III.**

This **study** also covers various technical aspects such as Coupling (Generic), Afferent Coupling, Efferent Coupling, Structural Coupling, Instability, among others. For instance, studies P1 and P3 from IEEE Xplore examine Coupling (Generic) and Afferent Coupling, while study P2 from Crossref adds elements like Structural Coupling and Instability. Studies P6 from Google Scholar and P7 from Crossref **focus** more on Structural Coupling. This study also notes that research from various sources including IEEE Xplore, Crossref, Google Scholar, OpenAlex, and Scopus/PubMed highlight different aspects of migrating to MVC-based microservices.

These findings indicate that although the potential benefits of MVC-based microservices are significant in improving system flexibility and scalability, their implementation still faces various challenges. The proposed Monolith to Microservices Decomposition Framework (M2MDF) can assist practitioners in planning and executing migration more effectively, as it provides a structured approach for each phase of decomposition. Future research should focus on developing more comprehensive tools and standard evaluation metrics that can be widely adopted to facilitate this migration. By addressing existing gaps and leveraging the M2MDF framework, future research is expected to make significant contributions to more effective and efficient software development practices, providing deeper insights into the adoption and evolution of microservices in the MVC context.

IV. Conclusion

This research explores various aspects of the migration from monolithic architectures to MVC-based microservices within web frameworks. The study gathers data from various academic databases such as IEEE Xplore, Crossref, Google Scholar, OpenAlex, and Scopus to analyze the trends, methodologies, and challenges associated with this migration process. Several studies (P1, P2, P3, P9, P15, P18, P19, P20, P21, P30) focus on measuring coupling, including generic coupling, afferent coupling, and efferent coupling. For example, studies P1 and P3 from IEEE Xplore evaluate generic coupling and efferent coupling, while P2 from Crossref also highlights afferent coupling and generic coupling.

This research also **emphasizes** the importance of system modulation and optimization using various metrics such as LOC (Lines of Code), CHM (Class Hierarchy Metrics), CHD (Class Hierarchy Depth), and relational cohesion. For instance, study P5 from IEEE Xplore demonstrates evaluation using LOC and CHM metrics. Several studies also stress the significance of testing and visualization to assess the effectiveness of the migration process. Additionally, some studies (P6, P7, P12, P18, P22, P26) from Google Scholar and Crossref highlight system performance and efficiency aspects, including execution time, CPU usage, memory usage, and network overhead. In terms of availability and scalability, some studies (P6, P8, P10, P16, P17, P19, P23, P30) underscore the importance of cluster size and the number of microservices.

However, bibliometric analysis of various databases also reveals that while research on microservices is extensive, few studies specifically address migration to MVC-based microservices. This absence indicates a significant research gap and suggests a substantial opportunity for further exploration in this area. Overall, this research indicates that while the migration to MVC-based microservices architecture offers significant potential benefits, many challenges remain to be addressed. The Monolith to Microservices Decomposition Framework (M2MDF) is expected to help overcome some of these challenges.

Acknowledgment

This research was funded by Institut Teknologi Sepuluh Nopember (ITS) under Penelitian Dana Departemen Program, Penelitian Keilmuan, Penelitian Flagship, Penelitian Kolaborasi Pusat and under the Project Scheme of the Publication Writing and Intellectual Property Rights (IPR) Incentive (Penulisan Publikasi dan Hak Kekayaan Intelektual/PPHKI) Program 2024, and the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia under Beasiswa Pendidikan Indonesia (BPI), Pusat Layanan Pembiayaan Pendidikan (Puslapdik), Balai Pembiyaan Pendidikan Tinggi (BPPT), and Lembaga Pengelola Dana Pendidikan (LPDP). BPI Number: 202327091197.

TABLE III
DECOMPOSITION CANDIDATES EVALUATION METRICS

Studies	Coupling (Generic)	Afferent Coupling	Efferent Coupling	Structural Coupling	Instability	IFN	LOC	CHM	CHD	Relational Cohesion	Modularity	SMQ	Functionality	Execution Time	CPU Usage	Memory Usage	Network Overhead	Efficiency	Scalability	Availability	Cluster Size	No. Of Microservice	NED	Manual Comparison	None	MVC	Source
P1[1]								V	V																		IEEE Xplore
P2[2]	V						V										V							V			Crossref
P3[3]		V	V		V		V																				IEEE Xplore
P4[9]						V					V	V											V				Crossref
P5[14]						V		V	V			V															IEEE Xplore
P6[9]																		V	V	V							Google Scholar
P7[19]														V	V	V											Crossref
P8[18]						V						V											V				IEEE Xplore
P9[13]							V				V											V	V				IEEE Xplore
P10[3]	V																				V						Google Scholar
P11[17]																								V			Open Alex
P12[18]														V	V	V											Crossref
P13[12]																								V			IEEE Xplore
P14[11]							V																				Crossref
P15[15]	V						V																	V			IEEE Xplore
P16[1]																									V		Scopus, PubMed
P17[7]																									V		Scopus, PubMed
P18[16]			V				V							V	V	V	V										Crossref
P19[3]																									V		Scopus, PubMed
P20[1]	V											V															Crossreff
P21[16]		V	V		V				V																		IEEE Xplore
P22[11]																					V						OpenAlex
P23[8]																									V		Scopus, PubMed
P24[4]						V		V	V			V	V														IEEE Xplore
P25[4]				V																							IEEE Xplore
P26[10]								V	V																		Google Scholar
P27[5]																											Google Scholar
P28[13]														V			V										OpenAlex
P29[6]																									V		Google Scholar
P30[6]			V				V							V		V		V	V	V							IEEE Xplore
P31[19]																											Scopus
P32[14]	V																										Crossref
P33[5]																							V				OpenAlex
P34[16]																							V				Google Scholar
P35[18]																							V				OpenAlex

REFERENCES

[1] N. Ivanov and A. Tasheva, "A hot decomposition procedure: Operational monolith system to microservices," in *Int. Conf. Autom. Informatics, ICAI 2021 Proc.*, pp. 182–187, 2021.

[2] F. Freitas, A. Ferreira, and J. Cunha, "A methodology for refactoring orm-based monolithic web applications into microservices," *J. Comput. Lang.*, vol. 75, no. February 2022, p. 101205, 2023.

[3] P. A. Suraya, T. Ramadhan, N. Lutfiani, A. Khoirunisa, and U. Rahardja, "Blockchain, information and speculation calculations in indonesia: Recent work," in *2022 10th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–8, IEEE, 2022.

[4] W. Sejati, A. N. Pusoko, E. V. Aryadi, S. Andajani, D. P. A. Hidayat, E. Kurniyaningrum, and N. Z. O'Connor, "Flood routing and dam breach parameter calculation on sepaku semoi dam," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 6, no. 1, pp. 136–148, 2024.

[5] J. Flora, P. Goncalves, M. Teixeira, and N. Antunes, "A study on the aging and fault tolerance of microservices in kubernetes," *IEEE Access*, vol. 10, no. November, pp. 132786–132799, 2022.

[6] X. Hou *et al.*, "Alphar: Learning-powered resource management for irregular, dynamic microservice graph," in *Proc. - 2021 IEEE 35th Int. Parallel Distrib. Process. Symp. IPDPS 2021*, pp. 797–806, 2021.

[7] A. Villa, J. O. Ocharan-Hernandez, J. C. Perez-Arriaga, and X. Limon, "A systematic mapping study on technical debt in microservices," in *Proc. - 2022 10th Int. Conf. Softw. Eng. Res. Innov. CONISOFT 2022*, pp. 182–191, 2022.

[8] R. Budiraharjo, R. Sarno, D. Wijaya, H. Prasetyo, and I. Waspada, "Simulations to predict process model alignment with standard operating procedure.," *International Journal of Simulation Modelling (IJSIMM)*, vol. 23, no. 1, 2024.

[9] D. Aulia, R. Sarno, S. C. Hidayati, A. N. Rosyid, and M. Rivai, "Identification of chronic obstructive pulmonary disease using graph convolutional network in electronic nose," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 1, pp. 264–275, 2024.

[10] F. S. Putri, H. R. Ngemba, S. Hendra, and W. Wirdayanti, "Sistem layanan ujian psikotes sim menggunakan computer based test berbasis website: Sim psychological test service system using computer based test based on website," *Technomedia Journal*, vol. 9, no. 1, pp. 92–104, 2024.

[11] G. Buchgeher, R. Ramlerf, H. Stummer, and H. Kaufmann, "Adopting microservices for industrial control systems: A five step migration path," in *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, vol. 2021-September, 2021.

[12] M. R. Anwar and L. D. Sakti, "Integrating artificial intelligence and environmental science for sustainable urban planning," *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, vol. 5, no. 2, pp. 179–191,

2024.
[13] N. K. A. Dwijendra, I. Vaslavskaya, N. V. Skvortsova, T. P. Rakhlis, U. Rahardja, M. H. Ali, A. H. Iswanto, L. Thangavelu, and M. M. Kadhim, "Application of experimental design in optimizing fuel station queuing system," *Industrial Engineering & Management Systems*, vol. 21, no. 2, pp. 381–389, 2022.
[14] S. A. N. Stanley and S. Sebastian, "A systematic review on the identification and classification of patterns in microservices," in *2023 4th Int. Conf. Emerg. Technol. INCET 2023*, pp. 1–8, 2023.
[15] H. Unlu, T. Hacaloglu, N. K. Omural, N. Caliskanel, O. Leblebici, and O. Demirors, "An exploratory case study on effort estimation in microservices," in *Proc. - 2023 49th Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2023*, pp. 215–218, 2023.
[16] E. Volynsky, M. Mehmed, and S. Krusche, "Architect: A framework for the migration to microservices," in *Proc. - 2022 Int. Conf. Comput. Electron. Commun. Eng. iCCECE 2022*, pp. 71–76, 2022.
[17] Z. Maharani, A. Saputra, *et al.*, "Strategic management of public health risks: Correlation between water quality and aedes sp. in south jakarta," *APTISI Transactions on Management*, vol. 8, no. 1, pp. 66–70, 2024.
[18] H. Nurhaeni, A. Delhi, O. P. M. Daeli, S. A. Anjani, and N. A. Yusuf, "Optimizing electrical energy use through ai: An integrated approach for efficiency and sustainability," *International Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 106–113, 2024.
[19] V. Meilinda, S. A. Anjani, and M. Ridwan, "A platform based business revolution activates indonesia's digital economy," *Startupreneur Business Digital (SABDA Journal)*, vol. 2, no. 2, pp. 155–174, 2023.
[20] L. Meria, C. Yohana, U. Purwohedi, D. Apriliasari, *et al.*, "Reinforcing lecturer readiness to change by increasing psychological capital and engagement," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 5, no. 3, pp. 261–277, 2023.
[21] M. Saidi, A. Tissaoui, and S. Faiz, "A ddd approach towards automatic migration to microservices," in *Proc. 2023 IEEE Int. Conf. Adv. Syst. Emergent Technol.* $IC_ASET2023, pp. 1--6, 2023$.
[22] K. Sungkono, R. Sarno, B. Onggo, and A. Septiyanto, "Optimising waste management collaboration processes using hybrid modelling," *International Journal of Simulation Modelling*, vol. 23, no. 1, pp. 53–64, 2024.
[23] K. M. Koyya and B. Muthukumar, "A survey of saga frameworks for distributed transactions in event-driven microservices," in *Proc. 3rd Int. Conf. Smart Technol. Comput. Electr. Electron. ICSTCEE 2022*, pp. 1–6, 2022.
[24] J. Kazanavičius and D. Mažeika, "An approach to migrate from legacy monolithic application into microservice architecture," in *2023 IEEE Open Conf. Electr. Electron. Inf. Sci. eStream 2023 - Proc.*, pp. 1–6, 2023.
[25] K. A. A. Manurung, H. Siregar, I. Fahmi, and D. B. Hakim, "Value chain and esg performance as determinants of sustainable lending in commercial bank: A systematic literature review," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 6, no. 1, pp. 41–55, 2024.