

Evaluation of In-Memory Database in Academic Information System with Combination Approach

Haerullah¹

¹ Universitas Mulia Samarinda, Indonesia

Email Correspondence: haerullah@universitasmulia.ac.id

Article history

Submitted: 2026/04/11; Revised: 2026/05/06; Accepted: 2026/06/05

Abstract

In the era of modern information systems, improving database efficiency and performance is one of the critical factors to ensure smooth operations and real-time transaction processing. This study aims to evaluate the application of in-memory databases in academic information systems. The research method uses a combination of quantitative and qualitative approaches. The importance of this research lies in the central role of database efficiency in supporting the performance, speed of service, and reliability of academic information systems. The analysis was performed using the TPC-C benchmark scenario to compare the performance between disk-based and in-memory databases under various test conditions. The results show that the use of in-memory databases provides faster response times and higher transaction throughput compared to disk-based databases, especially in online transaction processing (OLTP) transactions. In addition, the test results also highlight the importance of index optimization strategies and parallelism settings in supporting system performance. This study provides strategic recommendations related to the migration of academic information systems to in-memory architecture to improve overall operational efficiency.

Keywords

Academic Information Systems, Database Performance, In-Memory Database, TPC-C, Transaction Efficiency.



© 2026 by the authors. This is an open-access publication under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY SA) license, <https://creativecommons.org/licenses/by-sa/4.0/>.

INTRODUCTION

In the era of modern information systems, improving the efficiency and performance of databases is one of the critical factors to ensure smooth operations and real-time transaction processing (K et al., 2020). This study examines the evaluation of the use of in-memory databases in academic information systems with a combination approach of qualitative and quantitative evaluation. This research is very important because the performance and efficiency of databases directly determine the ability of academic information systems to deal with transaction surges, ensure service speeds, and support timely decision-making in an increasingly competitive and dynamic higher education environment. This method involves a comparative analysis of in-memory and disk-based database systems in terms of performance,

transaction processing efficiency, as well as the influence of parallelism and concurrent user load (Sirin, 2021).

The use of in-memory technology has gained a special place in various transaction applications due to its ability to store all data and changes in physical memory. This provides the advantages of very short response times and higher transaction throughput compared to disk-based databases that rely on disk storage and memory buffers (Kaur & Kaur, 2018). Nonetheless, the implementation of in-memory databases requires careful planning, especially in terms of index design and system configuration adjustments, to avoid potential performance bottlenecks that may occur during the migration process from disk-based to in-memory systems.

This research is based on the need to improve the efficiency of the transaction process, an aspect that is very crucial in the development of academic information systems. With the growing volume of data and transaction intensity, especially in academic contexts (such as enrollment, grade evaluation, and academic administration management), the selection of the right database technology is a strategic choice to support operational needs and decision-making quickly and accurately (Farshidi et al., 2018).

Based on the background and novelty that has been described, this study seeks to overcome problems related to the effectiveness of the implementation of in-memory databases in academic information systems, which are still a challenge in many higher education institutions. The main issue raised was how to compare the performance between disk-based and in-memory systems in academic applications, as well as technical factors such as index design and parallelism configurations that affect the success of system migrations. Therefore, this study aims to (1) evaluate the performance of in-memory databases compared to disk-based in academic information system scenarios, (2) analyze the influence of index optimization and parallelism settings on transaction efficiency, and (3) provide strategic recommendations for the development of academic information systems based on in-memory technology.

METHODS

This study uses a quantitative approach combined with qualitative evaluation to assess the application of in-memory databases in academic information systems. Although the available experimental data are sourced from comparative papers between in-memory and disk-based using the TPC-C benchmark, this methodology is adapted to understand its impact on academic information systems (Ayub & Ali, 2018).

The experiment was conducted using two databases:

- D5: Disk-based (TPCC_Disk_5GB)
- M5: In-memory (TPCC_Memory_5GB)

Both were tested with the HammerDB tool in a scenario of 10,000 transactions per user, on variations in the number of concurrent users (1 and 5 users), as well as parallelism

modifications (MAXDOP 0 and 1) (Ayub & Ali, 2018). Monitoring is carried out using SQL Server Profiler, recording the average execution duration, CPU usage, and disk activity on five main stored procedures: delivery, newword, ostat, payment, slev.

The test environment used a single server unit with Intel Xeon E5-2670 processor specifications, 32 GB of RAM, 1 TB SSD storage, and Windows Server 2019 operating system. Testing was conducted on the Microsoft SQL Server platform version 2019 Standard Edition, which has been configured to support both disk-based and in-memory modes. The databases tested are of two types: TPCC_Disk_5GB for disk-based scenarios and TPCC_Memory_5GB for in-memory scenarios, both of which have equivalent schema structure and data size.

The transaction load simulator uses HammerDB to run a scenario of 10,000 transactions per user on each database, with variations in the number of concurrent users (1 and 5 users) as well as parallelism settings through the MAXDOP (Maximum Degree of Parallelism) parameter at values of 0 and 1. During the experiment, system activity was monitored using SQL Server Profiler to obtain data on the average duration of stored procedure execution, CPU usage, and disk activity. Database configurations, including index adjustments and parallelism settings, are adjusted on each scenario to ensure objective and replicable test results.

The analysis is carried out with two main approaches:

1. Analyze the Average Transaction Duration

- Each stored procedure was analyzed based on the average execution time under different test conditions (Maspupah et al., 2023).
- A comparison table is created to illustrate the performance difference between disk-based and in-memory databases

2. Statistical Analysis with T-Test

- The T-Test statistical test was used to measure the significance of the performance difference between the two database systems (Kim, 2019).
- The test results showed that the difference in execution time in certain procedures had a very low p-value, indicating a statistically significant difference

Although the basic experiment was conducted on the TPC-C scenario, this study adapted it as a performance evaluation model for academic information systems. In the academic context, transactions such as registration, administrative verification, and academic evaluation have similar characteristics to OLTP transactions. Therefore, the results of the performance evaluation obtained from the experiment can be used as a reference to determine whether migrating to an in-memory system will provide significant benefits (Gao et al., 2020).

To provide a clear and comprehensive picture of the performance comparison between the two database systems, this study presents various visualizations, including: Average Transaction Duration Statistical Table: Comparing each stored procedure under different test conditions. Transaction Process Flow Diagram: Illustrates the flow of transactions on both in-

memory and disk-based databases. TPC-C Benchmark Process Mermaid Chart: Shows the relationship and difference in transaction execution between the two systems.

These visualizations are provided to help the reader understand the dynamics of the database system being tested and provide a detailed overview of the efficiency and efficacy of both approaches.

RESULTS AND DISCUSSION

Elaboration of test results and performance analysis based on the experiment data obtained. The evaluation was carried out through a comparison of the duration of the transaction, the influence of concurrent users, and the impact of parallelism settings on the two types of databases.

Comparison of Transaction Duration

Based on experimental data taken from the SQL Server Profiler monitoring tool, the average execution time for each stored procedure differs between disk-based (D5) and in-memory (M5) databases (Ki& Savijanja, n.d.). The average execution time statistics for each procedure stored on a disk-based database (D5) can be seen in Table 1. This table shows how the increase in the number of concurrent users and parallelism settings affect the execution duration of each procedure.

Table 1. Statistics on Average Duration of Procedure Execution on Disk-Based Databases (D5)

Procedure	1 User, MAXDOP 0	1 User, MAXDOP 1	5 User, MAXDOP 0	5 User, MAXDOP 1
Delivery	8.683 ms	16.179 ms	33.659 ms	30.856 ms
Neword	13.292 ms	19.419 ms	29.543 ms	53.900 ms
You buy payment	11.344 ms	14.710 ms	35.717 ms	39.048 ms
Discounts	4.954 ms	7.495 ms	11.596 ms	26.715 ms
	3.126 ms	31.420 ms	6.874 ms	7.810 ms

Meanwhile, Table 2 shows similar statistics for *in-memory* (M5) databases, which generally show more stable and optimal execution times on most stored procedures.

Table 2. Statistics on Average Duration of Procedure Execution in In-Memory Database (M5)

Procedure	1 User, MAXDOP 0	1 User, MAXDOP 1	5 User, MAXDOP 0	5 User, MAXDOP 1
delivery	354.719 ms	187.619 ms	267.056 ms	367.674 ms
neword	9.733 ms	8.662 ms	14.237 ms	11.526 ms
You buy payment	3.272 ms	3.217 ms	5.940 ms	4.666 ms
Discounts	2.208 ms	2.172 ms	3.169 ms	2.644 ms

discounts	1.088 ms	1,037 ms	1.398 ms	1.168 ms
-----------	----------	----------	----------	----------

For easy visual comparison, Figure 1 shows a graph comparing the execution duration between disk-based and *in-memory* databases in various test scenarios.

Perbandingan Durasi Eksekusi (ms) antara Disk-Based dan In-Memory Database

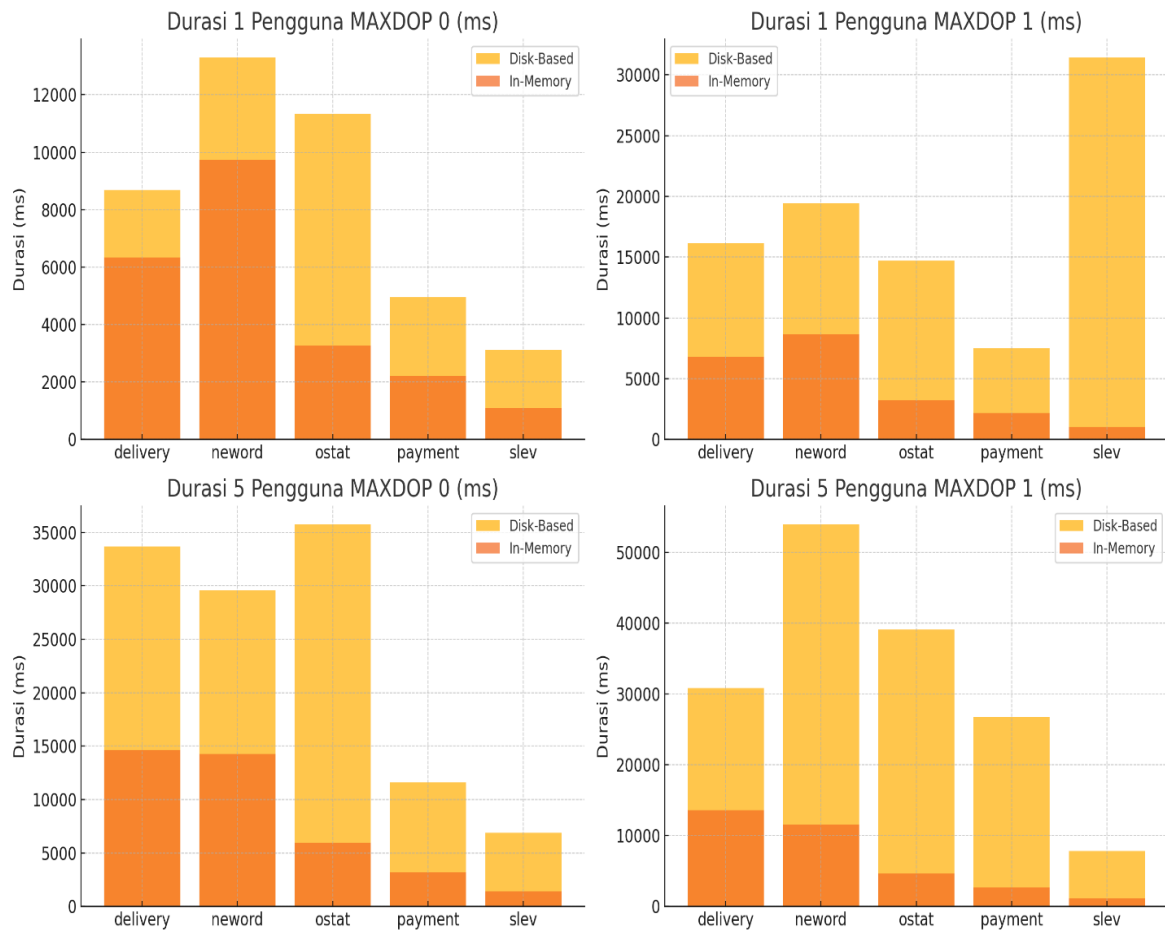


Figure 1. Comparison Graph of Duration between Disk-Based and In-Memory Databases

The above tables and graphs indicate that most procedures stored on in-memory databases have more consistent and, under certain conditions, more optimal execution times compared to disk-based databases. However, it should be noted that in the initial tests, procedures such as "delivery" showed very high execution times, which were then significantly reduced by the addition of a special index to the "new_order" table.

Data Analysis and Discussion

The test results show that in-memory databases consistently provide much faster transaction execution times compared to disk-based systems, especially in scenarios with a larger number of concurrent users. For example, in the "neword" procedure with five users and the MAXDOP 0 setting, the average execution time on an in-memory system is 14,237 ms, while on a disk-based

system it is 29,543 ms. Significant differences are also seen in other procedures such as "delivery" and "ostat", which confirm the advantages of in-memory architecture in handling transaction loads simultaneously.

The superior performance of this in-memory database is mainly influenced by the mechanism of storing and processing data directly in physical memory, so that the read/write process runs with minimal latency and without the locking constraints that usually occur in disk-based systems (Kaur & Kaur, 2018; Dice et al., 2018). The t-test statistical test conducted on the experimental data reinforced this finding, where the entire p-value was well below the significance threshold of 0.05, which indicates that the difference in performance between the two systems is really statistically significant (Kim, 2019). These findings are in line with the findings of the research of Ayub & Ali (2018) and Gao et al. (2020), which both highlight the importance of in-memory architecture to improve OLTP transaction efficiency in information systems with high transaction volumes.

Practically, the results of this study have important implications for the development and operation of academic information systems in higher education. The implementation of in-memory databases has been proven to improve transaction responsiveness and throughput, making it particularly relevant to academic applications that require real-time data processing, such as student enrollment or academic assessments. However, the implementation of an in-memory system also requires infrastructure readiness, especially memory capacity and index design optimization, so that potential bottlenecks can still be anticipated.

Although it provides convincing results, this study still has limitations, including testing only one TPC-C workload scenario and a test environment with specific hardware specifications. Follow-up research is suggested to expand workload variations, use real data and applications, or compare performance with other database technologies to obtain a more comprehensive picture of the effectiveness of in-memory databases in academic information systems.

Concurrent User Influence and Parallelism

Testing was performed on two scenarios of number of users (1 and 5 users) as well as two parallelism settings (MAXDOP 0 and MAXDOP 1). The results of the analysis show significant differences as follows:

Basis Data Disk-Based:

- The addition of concurrent users significantly increases response time. This is due to the locking mechanism that SQL Server must manage to maintain data consistency (Tasmere & Salehin, 2019).
- Parallelism settings (with the intention of limiting the number of CPU cores used) have varying impacts; for example, the "delivery" procedure was slightly improved when the MAXDOP was changed from 0 to 1, but the concomitant increase in transaction load still resulted in a significant increase in duration (Dice et al., 2018).

Basis Data In-Memory:

- In-memory systems generally show better response time stability despite the increase in concurrent users. This is because there is no locking mechanism that burdens the transaction execution process
- The MAXDOP settings on in-memory databases also have varying impacts; There is an improvement in performance in some stored procedures but in certain cases, such as in the "delivery" procedure, poor initial performance can be overcome by index optimization

The following visualization illustrates the flow of the transaction process and how variations in the number of users and the parallelism setting affect the performance of both systems:

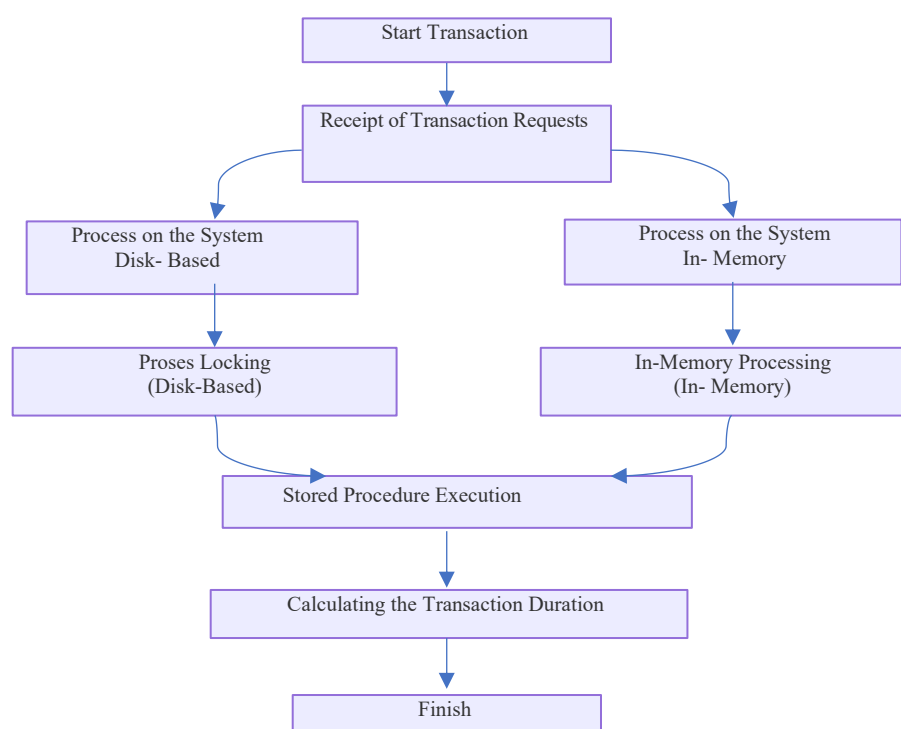


Figure 1. Flow of Disk-based and In-Memory Database Evaluation Process

The diagram above illustrates the difference in transaction processing flows on disk-based and in-memory databases. Disk-based systems involve a locking mechanism that tends to increase execution time, while in-memory systems perform processing directly in memory making them faster and more efficient.

Statistical Analysis with T-Test

To ascertain the significance of the difference in performance between disk-based and in-memory databases, statistical analysis was performed using the T-Test test. Based on the test results, the p-value for each stored procedure is very low (for example, the p-value for the "delivery" procedure on 1 user with a MAXDOP setting of 0 is **8.93e-12**). A low p-value indicates that the difference in execution time between the two systems is statistically significant, so the in-

memory performance advantage is not a coincidence.

Table 3: T-Test Results for Stored Procedures

Stored Procedures	Test Conditions	p-value
delivery	1 User, MAXDOP 0	8.93e-12
delivery	1 User, MAXDOP 1	4.01 and -13
delivery	5 Users, MAXDOP 0	8.45e-32
delivery	5 Users, MAXDOP 1	8.94e-11
Neword	1 User, MAXDOP 0	2.99e-22
Neword	1 User, MAXDOP 1	3.57e-07
Neword	5 Users, MAXDOP 0	1.88e-167
Neword	5 Users, MAXDOP 1	8.81e-10

The table above shows the p-values which indicate that the difference in performance between the two types of databases is statistically significant. This gives confidence that the advantages of in-memory databases in handling transactions do not occur by chance, but rather are the result of optimal storage and processing mechanisms

The Effect of Index Optimization on In-Memory Databases

In the initial tests, the "delivery" stored procedure on an in-memory database showed a very high execution duration. However, after optimizing through the addition of a new index to the "new_order" table, the procedure execution time was drastically reduced from 354719 ms to **6328 ms** for user scenarios with MAXDOP 0.

Table 4: Comparison of Delivery Transaction Duration Before and After Index Optimization

Test Conditions	Duration	
	Before Optimization (ms)	Duration After Optimization (ms)
1 User, MAXDOP 0	354719	6328
1 User, MAXDOP 1	187619	6806
5 Users, MAXDOP 0	267056	14588
5 Users, MAXDOP 1	367674	13604

This table emphasizes the importance of index optimization strategies, especially in in-memory database applications, to achieve maximum performance. Index optimization allows for a reduction in processing overhead that occurs due to repetitive read/write operations, thereby improving the overall efficiency of the system.

Migration Discussion and Implementation

The test results described earlier provide a clear picture of the performance advantages of in-memory databases in transaction processing compared to disk-based systems. Although the main focus of the research is on performance and efficiency analysis, there are several important points to be discussed, including:

Technical and Strategic Implications

The implementation of in-memory databases in academic information systems has significant implications, both from a technical and strategic perspective. Technically, the use of in-memory databases is promising:

- **Faster Response Times:** With all the data stored in memory, the transaction execution process becomes faster, which is crucial for academic applications that require real-time data changes.
- **Increased Transactional Throughput:** In-memory databases are able to handle high transaction volumes without degrading performance, making them ideal for regular activities such as student enrollment, schedule updates, and academic outcome reporting.
- **Efficiency in Resource Utilization:** Optimal CPU and memory utilization in in-memory systems allows for runtime savings and reduced latency, although it must be balanced with the adequacy of physical memory on the server.

Strategically, the migration from disk-based systems to in-memory databases must be based on several things, namely:

- **Index Design Valuation:** As proven in testing, the transformation of databases from disk-based to in-memory requires an in-depth review of the index design to avoid performance bottlenecks.
- **Parallelism Configuration Adjustments:** CPU core usage settings (e.g. MAXDOP settings) need to be tailored to transaction types and workloads, in order to maximize in-memory performance gains.
- **Physical Limitations Considerations:** Although in-memory databases offer high speeds, intensive memory usage requires careful resource planning in both hardware and memory management

Relevance to Academic Information Systems

In the context of academic information systems, speed and efficiency are essential to support a wide range of administrative and operational activities. The implementation of in-memory databases can have a positive impact in the form of:

- **Improved System Response:** A responsive academic information system will support rapid user interaction, so that students, lecturers, and administrative staff can access information in a timely manner.
- **Greater Transaction Capacity:** With high transaction throughput, academic systems can handle spikes in transactions, such as during the enrollment period, by more efficient without sacrificing performance.

- Easy Integration with Supporting Applications: The flexible in-memory system makes it easy to integrate with complementary applications such as e-learning modules, student portals, and online evaluation systems, all of which demand speed and stability in data processing.

However, the application of in-memory databases in academic information systems also has its own challenges, such as:

- Large Resource Requirements: Intense in-memory usage requires large memory capacity and capable server infrastructure.
- System Design Complexity: Migrating from conventional disk-based to in-memory systems requires replanning of database architectures to ensure compatibility and operational efficiency.

Case Studies and Benchmark Comparisons

To visually illustrate the comparison, here is a comparison table of a summary of the benchmark results of the two systems:

Table 5: Summary of Performance Comparison between Disk-Based and In-Memory Systems

Evaluation Parameters	Basis Data Disk-Based	Basis Data In-Memory
Transaction Response Time	Slower	Very Fast
Throughput Transaksional	Medium	Height
Concurrent User Influence	Significant Time Increase (Locking and Lock Escalation)	Stable, Almost Constant
Effects of Parallelism Settings	Varies, depending on load	Optimal at the right settings
Index Optimization Needs	Relatively Low	Very important

The table clearly shows the advantages of in-memory database systems over disk-based systems, especially in terms of execution speed and performance consistency when handling high transaction loads.

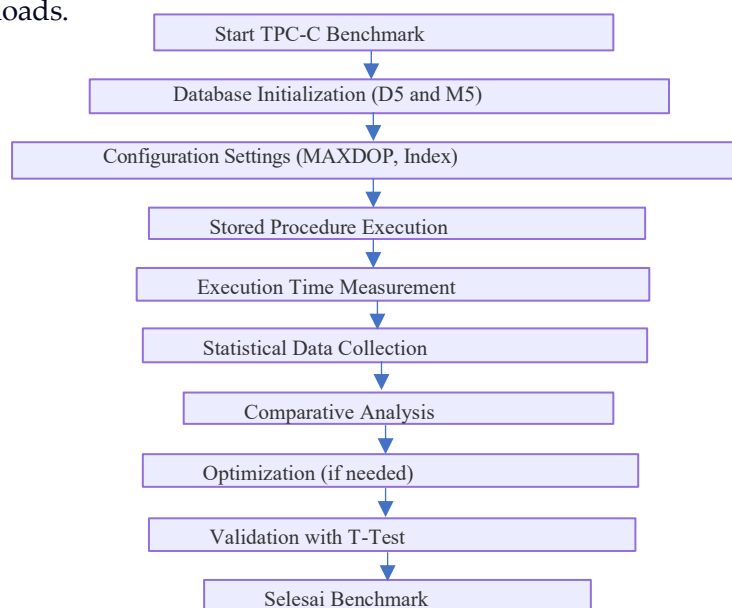


Figure 2. TPC-C benchmarking process flow

The diagram above illustrates the flow of the TPC-C benchmarking process applied to both database systems. Every stage from database initialization to statistical validation with the T-Test describes a comprehensive performance evaluation process.

CONCLUSION

The results of this study confirm that the implementation of in-memory databases in academic information systems is able to provide significant advantages in execution speed, transaction processing efficiency, and performance stability, especially when handling high transaction loads and concurrent user numbers. However, the success of the implementation is highly dependent on infrastructure readiness, index optimization strategies, and parallelism configuration management.

The implications of this research are important not only for the development of more responsive and efficient academic information systems, but also for policy makers in planning information technology infrastructure investments in higher education institutions. It is recommended that institutions that will adopt *in-memory databases* conduct thorough evaluations of hardware requirements, index design, and resource management on a regular basis to ensure optimal performance and sustainability of the system.

For further research, it is recommended that the test be expanded to various types of workloads, real implementation environments, and compare with alternative database technologies, so that the results obtained are more comprehensive and relevant to the needs of industry and the world of education.

REFERENCES

- Ayub, M. B., & Ali, N. (2018). Performance comparison of in-memory and disk-based databases using transaction processing performance council (TPC) benchmarking. *Journal of Internet and Information Systems*, 8(1), 1–8. <https://doi.org/10.5897/jiis2018.0106>
- Dice, D., Herlihy, M., & Kogan, A. (2018). Improving parallelism in hardware transactional memory. *ACM Transactions on Architecture and Code Optimization*, 15(1). <https://doi.org/10.1145/3177962>
- Farshidi, S., Jansen, S., de Jong, R., & Brinkkemper, S. (2018). A decision support system for software technology selection. *Journal of Decision Systems*, 27, 98–110. <https://doi.org/10.1080/12460125.2018.1464821>
- Gao, D., Reis, D., Hu, X. S., & Zhuo, C. (2020). Eva-CiM: A System-Level Performance and Energy Evaluation Framework for Computing-in-Memory Architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12), 5011–5024. <https://doi.org/10.1109/TCAD.2020.2966484>
- K, S. P., H, A. A., & A, E. R. (2020). Measuring Performance of Database Systems for Applications of Iot. *International Journal of Engineering Technology and Management Sciences*, 4(5), 49–52. <https://doi.org/10.46647/ijetms.2020.v04i05.009>
- Kaur, G., & Kaur, J. (2018). In-Memory Data processing using Redis Database. *International*

- Journal of Computer Applications*, 180(25), 26–31. <https://doi.org/10.5120/ijca2018916589>
- Ki, F., & Savijanja, S. (n.d.). *Zbornik radova Fakulteta tehni Ā kih nauka , Novi Sad CURVE FIBER-OPTIC SENSOR*. 2438–2441.
- Kim, H.-Y. (2019). Statistical notes for clinical researchers: the independent samples t -test . *Restorative Dentistry & Endodontics*, 44(3), 2–7. <https://doi.org/10.5395/rde.2019.44.e26>
- Maspupah, A., Muharram, M. K., & Daeli, S. G. (2023). Analisis Efektifitas Algoritma FAST Menggunakan Metrik Average Percentage Fault Detection dan Waktu Eksekusi Pada Test Case Prioritization. *Journal of Information System Research (JOSH)*, 4(2), 451–457. <https://doi.org/10.47065/josh.v4i2.2822>
- Sirin, U. (2021). *Micro-architectural analysis of in-memory OLTP : Revisited*. 641–665.
- Tasmere, D., & Salehin, M. N. (2019). Concurrency Control in Database Systems. *Bangladesh Journal of Multidisciplinary Scientific Research*, 1(2), 26–40. <https://doi.org/10.46281/bjmsr.v1i2.365>