p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

Performance and Efficiency Testing Analysis of Database Systems in Academic Information Systems

Utami Kusuma Dewi*[1], Ryan Lingga Wicaksono[2], Mahendra Dwi Febri[3], Villy Satria [4]
[1,2,3,4]Telkom University, Indonesia
E-mail: *[1]utamikusdew@telkomuniversity.ac.id, [2]wicaksonoryan@telkomuniversity.ac.id , [3]mahendradp@telkomuniversity.ac.id, [4]villysatria@telkomuniversity.ac.id

Abstract

This study examines the performance and efficiency of database systems within academic information systems, acknowledging the increasing demand for responsiveness and reliability in managing complex academic data. As educational institutions increasingly rely on digital systems, performance testing becomes essential to ensure that these systems continue to support the learning environment effectively. Guided by the ISO/IEC 25010 standard, the research focuses on evaluating three key aspects of performance efficiency: time behavior, resource utilization, and capacity. Using JMeter, a range of user load scenarios were simulated, and the results were examined through Control Quality Charts and Nelson Rules to detect underlying issues affecting system performance. The findings reveal that 82.5% of queries demonstrated good time behavior, and 80% performed well in resource usage. However, half of the tests related to capacity highlighted the need for further improvements. Some queries experienced delays and consumed excessive CPU and memory resources, indicating areas where optimization is required. These insights highlight the importance of refining queries and managing resources more effectively to ensure a seamless user experience. Future research should consider automated optimization, machine learning-based performance prediction, and system scalability, especially in more dynamic and distributed academic environments.

Keywords—*database performance, academic information system, ISO/IEC 25010, JMeter, quality control chart.*

Submission: June 19, 2025 *Accepted: July 05, 2025* *Published: July 10, 2025*
DOI Article: https://doi.org/10.25134/ilkom.v19i2.438

1. INTRODUCTION

The advancement of information technology has significantly influenced nearly every aspect of daily life today. In the modern era, almost all fields of work rely on information and communication technology to support their operations, including data management in the fields of education and academia. [1]. The education and academic sectors are inherently complex systems, as they involve various components that can contribute to internal challenges. [2]. Academic systems play a crucial role in managing academic data and information, including schedules, grades, attendance records, and other details related to the teaching and learning process. [3]. At the core of these systems lies the database, a vital component responsible for storing and managing continuously evolving information [4]. As the number of users and the complexity of data continue to grow, database performance becomes a key factor in ensuring that system operations run smoothly, effectively, and responsively. [5]. Efficiency is a key aspect in nearly all software systems, including those used in information systems. This efficiency is measured based on factors such as response time, the number of users the system can handle, and its energy efficiency.[6]. Therefore, performance testing of databases is crucial for evaluating and identifying potential issues that may compromise overall system performance. [5].

Performance testing is a method used to assess how well an information system or software performs under specific load conditions.[7], [8] One of the standard models for evaluating software quality is ISO/IEC 25010 [9], which defines a range of software quality characteristics, including performance efficiency. This aspect includes sub-characteristics such as Time Behavior, Resource Utilization, and Capacity. The model provides a comprehensive framework for evaluating the performance quality of an information system or software application, measuring response time, resource utilization, and the system's maximum handling capacity. [9].

The Control Quality Chart and Nelson Rules serve as practical tools for identifying deviations or anomalies in system performance.

p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

The Control Quality Chart is used to monitor process variations, while the Nelson Rules help detect patterns that may indicate potential issues in system operations. [10]. Applying these tools within the context of database performance testing enables more accurate management and evaluation of system quality.

In this study, the quality of the database is measured within an academic information system that manages data related to education and teaching. The evaluation is conducted using a performance testing approach, focusing on the previously mentioned sub-characteristics. The results of the measurements are analyzed using the Control Quality Chart and Nelson's Rules to identify areas requiring improvement and to assess the overall quality of the system's performance. This approach aims to ensure that the database used in the academic information system can handle dynamic data growth and high-performance demands. [9], [10]

2. RESEARCH METHOD

The research method is divided into several main stages, as illustrated in Figure 1 below.

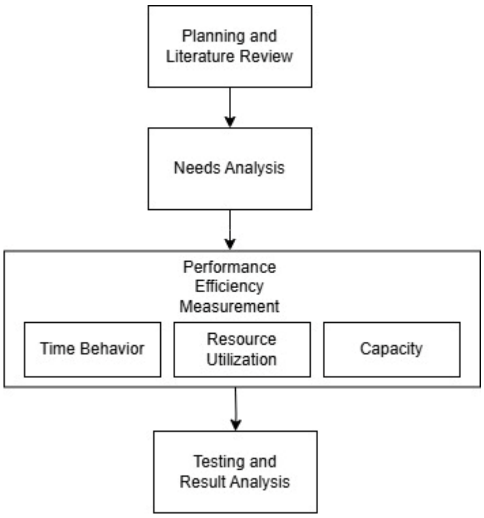


Figure 1. Research Method

2.1. Planning and Literature Review

The study begins with problem formulation, focusing on database performance testing, specifically in measuring response time, resource utilization, and capacity to ensure adequate system performance. The primary objective of this research is to evaluate the performance of the database system used in managing academic data. This stage also includes a literature review on ISO/IEC 25010, performance testing methodologies, and testing tools such as JMeter, Control Quality Chart, and Nelson Rules, which serve as the foundation for designing the performance testing to be conducted.

Table 1. Specification Needs

System Name	Specification
Processor	Intel Core i5
Memory	16 GB
Disk	500 GB
System Operation	Microsoft Windows 11
Database	Oracle 11 g
Programming Language	Java
IDE	Netbeans
Local server	Localhost
Browser	Google Chrome and Mozilla Firefox

In addition, the research planning also includes the need for both hardware and software to develop tools for conducting measurements, as outlined in Table 1.

2.2 Needs Analysis

This stage involves reviewing the specifications and components of the information system used to manage academic data.

a. Review the Requirement Specification Database
This involves reviewing the database's functional and non-functional requirements, including access speed, storage capacity, and data processing capabilities.
b. Identification of Database Component Specifications
This stage involves identifying the main components of the database system to be tested, including data structures, the queries in use, and the underlying storage and processing infrastructure.
c. Query Selection
This stage involves selecting the queries to be used in the performance tests, particularly those that involve user data and academic information.

In general, this stage is intended to ensure that the performance testing encompasses the critical components of the system that impact database performance.

2.3 Measurement

p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

The third stage involves developing and configuring testing tools to measure the performance efficiency of the database system. The focus of the measurement is on three main sub-characteristics: Time Behavior, Resource Utilization, and Capacity. The detailed matrix of these sub-characteristics is presented in Table 2.

Table 2. Performance Testing Sub-Characteristics

Sub-Characteristics	Matrix
Time Behavior	Response Time
	Turnaround Time
	Throughput
	Response Compliance
	Processing Margin Rate
Resource Utilization	Data Volume
	Memory Capacity
	Hard Disc Capacity
	I/O Device
	CPU Utilization
	Storage Period
Capacity	Max of I/O Device
	Max Utilization Rate of Transmission System

The tool used for testing is JMeter, which is configured to simulate various user load scenarios and measure response time, CPU utilization, and memory utilization.[11]. Additionally, the Control Quality Chart and Nelson Rules are prepared to analyze test results, monitor performance variations, and identify potential issues in the system's operational processes. [5], [9], [10].

2.3.1 JMeter Configurations

JMeter is configured by designing a test plan that includes various test cases to measure response time and resource utilization during the testing process. Each selected query is executed under multiple user load scenarios, ranging from light to heavy loads. The test results are then used to evaluate system performance and identify potential performance issues, such as high latency or inefficient resource usage. [5], [9], [10].

2.3.2 Query

Query testing is conducted by evaluating various types of database operations, such as select, insert, update, and delete, which are commonly used in managing data within academic information systems. The attributes measured during the testing include Execute Time, Memory Runtime, Memory Usage, and CPU Time. Each query is tested under different load levels to ensure that the system can handle a wide range of user requests efficiently.

2.4 Testing and Result Analysis

During testing, JMeter collects data related to response time, CPU utilization, memory usage, and I/O capacity to evaluate system performance. The test results are used to analyze how well the system handles load and utilizes resources efficiently, as well as to identify areas that require improvement. Subsequently, the collected data will be further analyzed using the Control Quality Chart and Nelson Rules.

a. Control Quality Chart
   In this study, the Control Quality Chart is used to analyze the results of database performance testing, specifically to monitor response time and resource utilization, including CPU and memory usage.

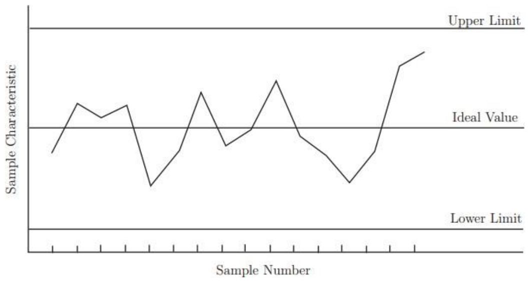


Figure 2. Control Quality Chart

The data from JMeter will be mapped into a chart, as illustrated in Figure 2, which is used to identify deviations from the established control limits, indicating potential issues in system performance. This tool helps ensure that the system remains within the desired quality control and provides insights for improvement when significant variability occurs..

Ideal Value

$$x = \frac{x_1 + x_2 + x..+ c_k}{k} \quad (1)$$

Upper Limit

$$UCL = x + \frac{3_s}{\sqrt{n}} \quad (2)$$

Lower Limit

$$LCL = x - \frac{3_s}{\sqrt{n}} \quad (3)$$

Keterangan

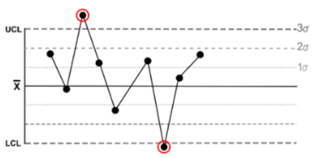
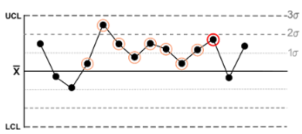
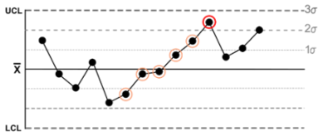
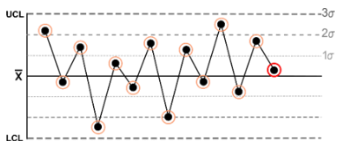
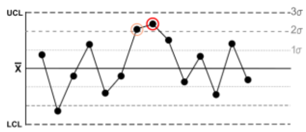
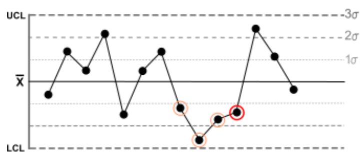
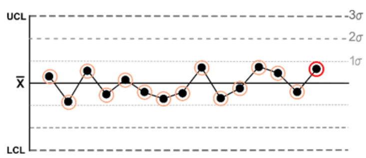
x	:	Average
x_1	:	Each data value
k	:	Data Sum
s	:	Standart Devitation
n	:	Sum of the data *sample*

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

b. Nelson Rules

Nelson Rules are used in this study to detect significant deviations in the database performance testing data. These rules help identify patterns or anomalies in response time, resource utilization, or capacity that indicate the system is operating outside of its normal range. In the context of this testing, Nelson Rules are applied to examine the results from the Control Quality Chart, to identify unusual spikes or fluctuations in performance that may signal issues or potential areas for improvement in the database system. [5], [9], [10]. The graph analysis using Nelson's Rules can be seen in Table 3.

Table 3. Nelson Rules

Rules	**Descriptions**	**Kegunaan**
Rules 1	There is a single point located more than three standard deviations from the mean.	Indicates a single anomaly in the process
		
Rules 2	There are nine (or more) consecutive points on the same side of the mean	Indicates a process shift around the average value
		
Rules 3	Six (or more) consecutive points that are continuously increasing or decreasing	Indicates a trend or tendency in the process
		
Rules 4	Fourteen (or more) points that alternately increase and decrease in value.	Indicates systematic effects such as changes in the machine, system, or operator
		
ules 5	Two (or three) out of three consecutive points are beyond the mean in the same direction.	Indicates a shift in the process average or an increase in standard deviation
		
Rules 6	Four (or five) out of five consecutive points are within the mean zone and in the same direction.	Indicates a shift in the process mean
		
Rules 7	Fifteen consecutive points fall within one zone of the mean on both sides.	Indicates subgroup stratification when observations come from different sources or are collected in various ways.
		
Rules 8	Eight non-consecutive points, with values alternating on	Indicates subgroup stratification when observations

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

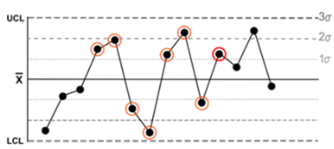
	both sides of the mean	within a single subgroup come from the same source but are collected through different methods
		

Table 4 presents the identified causes and corresponding explanations based on the graphs obtained from the measurement results.

Table 4. Possible Causes

Rules	**Possible Causes**
1,2	Configuration error Errors in measurement and planning Incomplete operation New personnel performing the task A process step was skipped A process step was not completed
3,4	Change in supporting equipment Change in work instructions New or repaired tools introduced Change in work procedures Change in the system New operator assigned Change in maintenance program
5,3	Using tools or certain machine parts Operator fatigue Temperature factors: the machine is too hot or too cold Inadequate maintenance
6	Lack of control after process improvement A mixture of more than one process Strong tendency to fall slightly out of control
7	Producing varying outcomes More than one process is invol.ved
8	Overcontrol or misgrouping (subgroups taken from different sources) Operator interference or disruption

Overall, this analysis aims to provide a comprehensive overview of system performance and to identify areas that require improvement or optimization.

3. RESEARCH RESULT

The results of this study are presented below, based on measurements conducted using the specified sub-characteristics.

3.1 Time Behavior

Based on the measurement results, one query was found to have a response time exceeding the average, reaching 1.16 seconds. This finding is categorized as an anomaly according to Nelson Rule 1, which identifies data points that fall beyond three standard deviations from the mean. This condition indicates that the query is running inefficiently and may become a source of system delays if not addressed promptly.

To resolve this issue, it is recommended to optimize the query by reviewing its SQL structure, including the possible use of complex joins, nested subqueries, or filters that do not utilize indexes. Query tuning strategies, such as adding appropriate indexes, simplifying query logic, or utilizing stored procedures, can help reduce response time. Additionally, reviewing system configuration and process flows is essential to ensure there are no misconfigurations or unfinished executions. These efforts aim to maintain system performance stability, especially under high user load conditions.

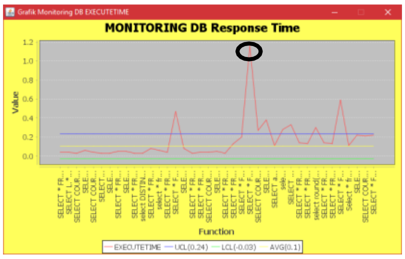
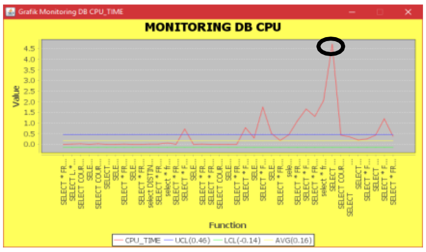


Figure 3. Graph of result time behavior

3.2 Resource Utilization

The resource utilization testing was conducted using CPU utilization and memory capacity metrics. The test results showed an average CPU time of 0.16 seconds. In Figure 4, one data point exceeds the Upper Control Limit (UCL), indicating a significantly high CPU usage of 3.18 seconds.

p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom



.

Figure 4. Graph of CPU *utilization*

.

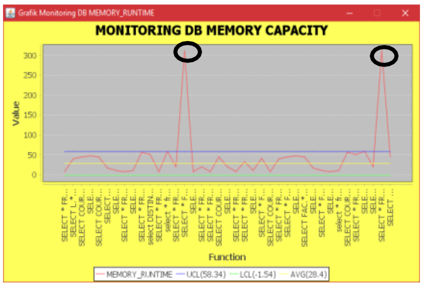


Figure 5. Graph of the result: memory utilization

According to the ISO/IEC 25010 standard, the Resource Utilization sub-characteristic assesses how efficiently a system utilizes the necessary resources to perform its functions. When CPU or memory usage exceeds expected thresholds, it indicates inefficiency and increases the risk of performance degradation under a heavier user load. Based on Table 4, possible contributing factors may include inadequate system maintenance, improper equipment use, or operational inconsistencies (as described in Rules 3, 4, and 5). Therefore, further query profiling and optimization are strongly recommended to ensure stable and efficient system performance.

3.3 Capacity

Capacity testing was conducted using memory usage. The results showed an average maximum I/O value of 0.3%, indicating stable performance without significant deviations.

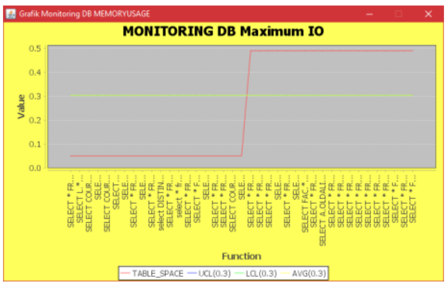


Figure 6. Graph of the result memory usage

However, the graph in Figure 6 illustrates that while the values remain under the Upper Control Limit (UCL), some points approach the threshold, suggesting potential strain under heavier loads.

The Capacity sub-characteristic in ISO/IEC 25010 refers to the extent to which a system can maintain optimal performance under maximum workload conditions. Although Nelson Rules detected no anomalies, the measurement results suggest that adjustments in resource allocation should be considered. This is especially important for anticipating future scalability needs. Strategies such as vertical scaling (adding more resources to the existing server) or horizontal scaling (adding more nodes) may help ensure consistent system performance under increasing demand.

Based on the evaluation and analysis of performance quality measurements, the test results indicate that most queries can deliver optimal response times under light to moderate loads. However, some queries experience significant spikes in response time as user load increases. The components that influence database performance are summarized in Table 3, based on the success percentage for each sub-characteristic.

Table 5. Success Percentage of Measurement

Sub-Karakteristik	Persentase
Time Behavior	Defect = (7/40) x 100% = 17,5 %
	Success = (33/40) x 100% =82,5%
Resource Utilization	Defect = (8/40) x 100% = 20%
	Success = (32/40) x 100% = 80%
Capacity	Defect = (50/40) x 100% = 50%
	Success = (50/40) x 100% = 50%

The detected defects represent the number of elements that impact performance, and the table also presents the average percentage results from all measurements conducted on each component.

As shown in Table 5, the analysis results indicate that for time behavior, 82.5% of queries performed successfully, while 17.5% experienced response time issues, suggesting the need for improvement in specific queries. For resource utilization, 80% of the queries ran smoothly; however, 20% still consumed excessive CPU and memory resources, indicating a need for optimization. Regarding capacity, 50% of the queries were successful, while the remaining 50% were hindered due to discrepancies in maximum size and byte allocation for user object segments, which require adjustment.

Based on these results, it is recommended that queries be optimized and

p-ISSN :1858-3911 , *e-ISSN : 2614-5405*
https://journal.fkom.uniku.ac.id/ilkom

resource allocation adjusted to improve overall system performance.

4. DISCUSSION

The performance testing results of the database system within an academic information system revealed several significant findings that can serve as a reference for improving system quality. Based on the Time Behavior measurements, the system generally demonstrated good response times, with the majority of queries executing in under one second. However, some queries exhibited higher-than-average response times, indicating potential inefficiencies that require query optimization to enhance response efficiency.

In terms of Resource Utilization, the findings showed that some queries consumed more CPU and memory resources than expected. The average CPU usage was recorded at 0.16 seconds, while memory usage reached 28.4 KB. This high resource consumption suggests inefficiencies in specific queries, which can be improved by optimizing them to reduce memory and CPU usage.

On the other hand, the Capacity measurements indicated that the system could handle a maximum I/O of up to 0.3%, demonstrating strong capacity in managing data loads. There were no significant issues related to capacity, as the maximum I/O values remained within stable limits, suggesting that the system can manage larger data volumes without substantial performance degradation.

Overall, the test results identified several areas requiring improvement, particularly in response time and resource utilization. Therefore, optimizing specific queries and enhancing resource management are recommended to improve the overall system performance.

5. CONCLUSION

Based on the performance testing results, the database system demonstrated exemplary performance across most queries, with optimal response times under light to moderate loads. However, several queries require further optimization in terms of response time and resource utilization, particularly in terms of CPU and memory usage. Although I/O capacity did not present any issues, the testing identified the need for improved resource management in specific queries to enhance system efficiency. Therefore, query optimization and improved resource management are expected to enhance overall system performance, particularly under high-load conditions.

6. SUGGESTION

Future research is expected to expand its scope by testing database systems under more diverse conditions, such as deployment on cloud platforms or distributed systems. Further studies may also focus on automated query optimization analysis and the development of machine learning techniques for dynamic performance prediction and improvement.

Machine learning methods, such as decision trees, reinforcement learning, or neural networks, can be applied to detect and optimize inefficient query patterns automatically. Testing may also include evaluating the impact of horizontal and vertical scaling on database performance, particularly for applications that handle large volumes of data. This research is expected to provide deeper insights into database management in more complex and dynamic environments.

REFERENCES

[1] I. Akbar, “Analisis Dan Perancangan Website SMKN 13 Bandung.” [Online]. Available: https://journal.fkom.uniku.ac.id/ilkomTerakreditasiSINTA5

[2] M. Berlianti Akbar and T. Sutabri, “Analisis Manajemen Layanan Penerimaan Siswa Baru Menggunakan Framework ITIL Versi 3 Pada Domain Service Design di SMA Negeri 22 Palembang.” [Online]. Available: https://journal.fkom.uniku.ac.id/ilkom

[3] L. Indah Sari, W. Aribowo Probonegoro, I. Atma Luhur, and S. Informasi ISB Atma Luhur, “Analisis Sistem Informasi Akademik Menggunakan Metode FAST Pada SDIT Al-Qudwah Pangkalpinang.” [Online]. Available: https://journal.fkom.uniku.ac.id/ilkom

[4] T. Taipalus, “Database management system performance comparisons: A systematic literature review,” *Journal of Systems and Software*, vol. 208, Feb. 2024, doi: 10.1016/j.jss.2023.111872.

[5] V. Vukovic, V. Vuković, J. Đurković, and J. Trninić, “Defining Performance Criteria and Planning Performance Tests for the Exam Registration Software,” 2014. [Online]. Available:

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

https://www.researchgate.net/publication/325366605

[6] T. Taipalus, "Database management system performance comparisons: A systematic literature review", doi: 10.48550/arXiv.2301.01095.

[7] S. Rochimah, U. Laili Yuhana, and S. Balqis Hidayat, "INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION journal homepage : www.joiv.org/index.php/joiv INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION Software Quality Measurement for Functional Suitability, Performance Efficiency, and Reliability Characteristics Using Analytical Hierarchy Process." [Online]. Available: www.joiv.org/index.php/joiv

[8] N. Srivastava, U. Kumar, P. Singh, and P. Singh, "Software and Performance Testing Tools," *Journal of Infor-matics Electrical and Electronics Engineering*, vol. 02, no. 001, pp. 1–12, 2021, doi: 10.54060/JIEEE/002.01.001.

[9] E. Peters and G. K. Aggrey, "An ISO 25010 based quality model for ERP systems," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 2, pp. 578–583, Apr. 2020, doi: 10.25046/aj050272.

[10] R. Mansharamani, A. Khanapurkar, B. Mathew, and R. Subramanyan, "Performance testing: Far from steady state," in *Proceedings - International Computer Software and Applications Conference*, 2010, pp. 341–346. doi: 10.1109/COMPSACW.2010.66.

[11] N. Husufa and I. Prihandi, "Optimizing JMeter on Performance Testing Using the Bulk Data Method," *Journal of Information Systems and Informatics*, vol. 3, no. 1, 2022, [Online]. Available: http://journal-isi.org/index.php/isi