Jurnal Teknik Informatika (JUTIF)
Vol. 6, No. 1, February 2025, pp. 75-86

DOI: https://doi.org/10.52436/1.jutif.2025.6.1.4173
p-ISSN: 2723-3863
e-ISSN: 2723-3871

PERFORMANCE ANALYSIS OF EXTRACT, TRANSFORM, AND LOAD METHODS FOR BUSINESS INTELLIGENCE IN E-LEARNING SYSTEMS USING PENTAHO DATA INTEGRATION

Aulia Kukuh Saputra[1], Kusuma Ayu Laksitowening[2], Anisa Herdiani[3]

[1,2,3]Informatics, Faculty of Informatics, Universitas Telkom, Indonesia
Email : [1]auliakukuhs@student.telkomuniversity.ac.id, [2]ayu@telkomuniversity.ac.id, [3]anisaherdiani@telkomuniversity.ac.id,

(Article received: December 4, 2024; Revision: January 7, 2025; published: February 20, 2025)

Abstract

The rapid adoption of Learning Management Systems (LMS) in higher education has resulted in the generation of large and complex datasets, posing significant challenges for efficient data integration and analysis. The urgency to address these challenges is driven by the growing demand for real-time analytics and data-driven decision-making in educational institutions. This study advances the field of computer science by evaluating and comparing the performance of three Extract, Transform, and Load (ETL) methods—Table Output, Sync After Merge, and Switch Case—using Pentaho Data Integration (PDI). The study introduces novel insights into ETL optimization techniques, focusing on execution time as the primary metric, critical for ensuring timely and reliable insights in Business Intelligence (BI) systems. Performance testing was conducted with synthetic datasets ranging from 150 to 1,000,000 records across five scenarios: data addition, synchronization, insertion, deletion, and combined operations. Results reveal that Sync After Merge outperformed other methods, achieving up to 35% faster execution times, particularly with large datasets. These findings contribute significantly to the advancement of data integration techniques in computer science, enabling institutions to optimize their BI systems, enhance system responsiveness, and support data-driven decision-making processes effectively. The research provides valuable insights for developing scalable ETL solutions in educational technology systems.

Keywords: *Business Intelligence, Data Warehouse, ETL performance, LMS, Pentaho Data Integration.*

1. INTRODUCTION

The adoption of Learning Management Systems (LMS) in higher education has grown rapidly, generating vast amounts of data related to student activities, assessments, and interactions [1]. When analyzed effectively, this data can provide valuable insights to enhance learning outcomes and support data-driven decision-making [2]. However, the increasing volume and complexity of LMS data present significant technical challenges, particularly in data integration and analysis processes [3].

One of the most critical challenges is in the transformation stage of the Extract, Transform, and Load (ETL) process. This stage involves sophisticated techniques to handle missing attributes, remove irrelevant columns, perform data typecasting, and aggregate data [4]. The complexity becomes even more significant when processing educational datasets, which often contain diverse student activities and assessment records.

Educational institutions often struggle to manage and process LMS data effectively. Traditional approaches to data integration are insufficient for handling the volume and variety of educational data, especially when real-time analysis is required [5]. Business Intelligence (BI) systems offer solutions for data-driven decision-making, but their implementation relies heavily on efficient data integration processes [6]. While open-source BI tools such as Pentaho have shown promise in addressing these challenges cost-effectively, their adoption in Indonesian educational institutions remains limited due to a lack of structured and integrated data access [7].

At the heart of BI systems lies the data warehouse (DWH), which consolidates and organizes data from multiple sources into multidimensional formats suitable for analysis [8]. The construction of a DWH depends on ETL mechanisms, which serve as the backbone of data integration. Research has shown that properly designed ETL processes are crucial for successful data integration and analysis in enterprise data warehouses [9]. Tools like Pentaho have demonstrated significant improvements in handling large-scale data integration tasks when implemented correctly [10]. These ETL processes are essential for preparing data for advanced analytics tools, such as dashboards and Online Analytical Processing (OLAP) [11].

Several strategies have been proposed to improve ETL performance, including optimizing the data extraction process, parallel processing, and

distributed computing [12]. Studies have also emphasized the importance of data governance and management practices to optimize ETL performance. Recent comparisons between ETL and E-LT approaches highlight their unique advantages depending on specific data integration needs [13]. These approaches have proven effective in improving ETL performance, particularly for large-scale educational datasets.

As the volume of LMS data continues to grow exponentially, optimizing ETL performance has become increasingly important. Execution time has emerged as a crucial metric for ensuring timely BI insights. Modern ETL techniques, such as parallel processing and incremental updates, are necessary to handle large datasets efficiently [14]. Rigorous ETL testing is also essential to ensure reliability and scalability, particularly in environments with frequent data updates [15]. Effective ETL preprocessing is critical for integrating data from multiple academic systems, such as Student Information Systems (SIS) and LMS, to ensure data quality and consistency [16].

This study evaluates the execution time of three ETL methods—Table Output, Sync After Merge, and Switch Case—implemented using Pentaho Data Integration (PDI). Testing was conducted on datasets ranging from 150 to 1,000,000 records to assess the scalability and efficiency of each method in transforming LMS transactional data into a DWH-ready format. By focusing on execution time, this research provides actionable insights for educational institutions to optimize their ETL processes and improve BI operations. Furthermore, it contributes to the understanding of ETL method selection, scalability challenges, and system responsiveness in e-learning environment.

2. RESEARCH METHODOLOGY

This research employs a systematic approach to evaluate Extract, Transform, Load (ETL) performance in e-learning data warehouse implementations. The methodology encompasses multiple interconnected phases, as illustrated in Figure 1.

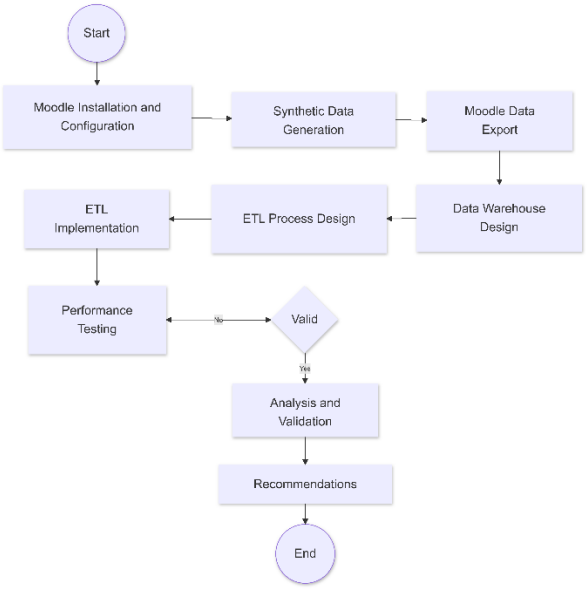

Figure 1. Research Methodology Flow for ETL Performance Analysis in E-learning Data Warehouse

The research methodology follows a structured sequence as depicted in Figure 1, demonstrating the interconnected phases from initial system setup through final analysis and recommendations. As shown in Figure 1, the process begins with Moodle installation and configuration, progresses through data preparation and ETL implementation, and concludes with analysis and recommendations. This sequential yet iterative approach ensures continuous validation throughout the research process.

2.1. Research Flow Overview

The research methodology follows a structured sequence designed to ensure comprehensive evaluation and reliable results. The process flow, as depicted in Figure 1, demonstrates the interconnected phases from initial system setup through final analysis and recommendations. This sequential yet iterative approach allows for continuous validation and refinement throughout the research process.

2.2. Moodle Installation and Configuration

The initial phase involves the implementation of Moodle version 4.1 as the foundation for educational data generation. This version was selected specifically for its comprehensive logging capabilities and widespread adoption in educational institutions. The system utilizes MySQL 8.0 database configuration, which provides essential performance optimization features and robust handling of large datasets. The configuration process includes the implementation of necessary educational plugins and modules to support diverse academic activities.

The system configuration encompasses the creation of hierarchical course structures that accurately reflect real academic organizations. User roles are implemented to match typical educational

institution hierarchies, while activity modules are configured to support various educational tasks including assignments, quizzes, and forums. The grading schema implementation aligns with standard academic evaluation practices.

2.3. Synthetic Data Generation

The data generation phase focuses on creating realistic educational datasets that accurately mirror real-world scenarios. Course-related data is generated following standard academic nomenclature, with activity patterns designed to reflect typical semester timelines. The assessment structures incorporate common academic evaluation methods to ensure data authenticity.

The data volume configuration implements a multi-tier approach to testing. The base dataset contains 150 records for initial validation, while medium-scale testing utilizes datasets of 15,000 and 50,000 records. Large-scale testing employs datasets of 500,000 and 1,000,000 records to evaluate system performance under significant data loads. These synthetic datasets include comprehensive student activity patterns, assignment submissions, and assessment records that simulate actual educational environments.

2.4. Data Warehouse Design

The data warehouse design phase implements a dimensional model optimized for educational data analysis. The design utilizes a star schema architecture that effectively captures the multidimensional nature of educational data. Fact tables are structured to record student activities, assessments, and performance metrics, while dimension tables maintain course information, temporal data, student details, and activity classifications. The implementation incorporates Type 2 slowly changing dimensions to preserve historical accuracy in tracking changes over time.

2.5. ETL Process Design

The ETL process design incorporates three distinct methods selected for their specific capabilities in handling educational data. The Table Output method serves as a baseline approach, providing direct data transfer functionality with simplified error tracking mechanisms. The Sync After Merge method offers advanced capabilities for incremental updates and efficient change detection, particularly valuable for ongoing data synchronization. The Switch Case method enables conditional data routing and handles complex transformation scenarios commonly encountered in educational data processing.

Three distinct ETL methods are implemented and tested in this research:

1. Table Output Method

The Table Output method represents the simplest data transfer mechanism in ETL processes. As shown in Figure 2, this method involves direct data movement from the source table (e.g., LMS Moodle - Course Sections) to the staging table (Staging Course Sections). Figure 2 illustrates how this straightforward approach manages data transfer through direct table connections. The method supports batch processing, with configurable batch sizes typically set to 1,000 records to balance efficiency and system resource utilization. Comprehensive error handling is embedded at key stages to monitor data flow integrity.

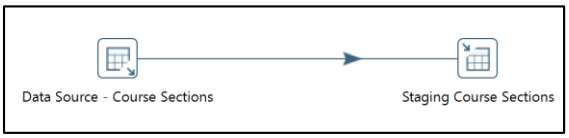

Figure 2. Table Output Method Implementation

2. Sync After Merge Method

The Sync After Merge method, as depicted in Figure 3, employs a sophisticated data synchronization approach. Figure 3 demonstrates how this method processes source and target datasets simultaneously, ensuring that only updated or new records are synchronized. The transformation involves multiple steps:

a. Sorting rows from both source and staging datasets.
b. Merging rows to identify differences between the datasets.
c. Applying synchronization through selective updates or inserts.

This approach minimizes redundancy while maintaining record-by-record comparisons, which enhances the accuracy of data synchronization.

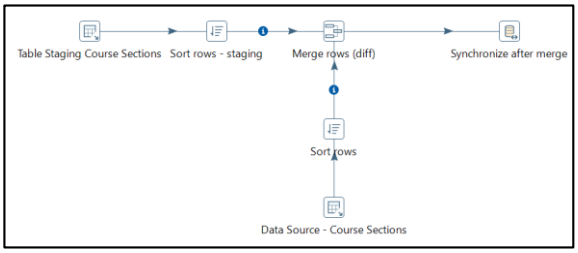

Figure 3. Sync After Merge Method Workflow

3. Switch Case Method

The Switch Case method, as visualized in Figure 4, uses conditional logic to handle multiple processing scenarios efficiently. Each record is examined for its status (e.g., add, update, or delete) and routed to the appropriate processing path:

a. Add New Rows for records not present in the staging table.
b. Update for modified records.
c. Delete for obsolete records.

This method includes dedicated error recovery procedures to ensure reliability and strategic placement of performance monitoring points to optimize the ETL process.

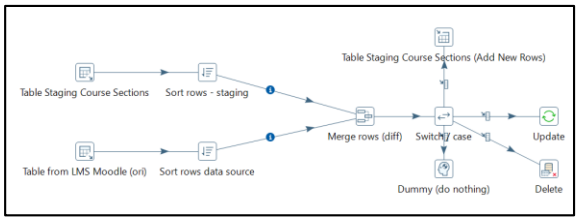

Figure 4. Switch Case Method Workflow

2.6. Testing Scenarios

The testing process implements five distinct scenarios designed to evaluate ETL performance under different operational conditions typically encountered in e-learning data warehouse environments.

1. Add Data Scenario

This scenario evaluates the performance of each method when adding new data to an empty staging area. The process begins with a clean staging environment and measures the time required to load various volumes of data, starting from 150 records up to 1,000,000 records. This scenario establishes baseline performance metrics for each ETL method.

2. Without Difference Scenario

This scenario tests performance when source and target data are identical, with no changes requiring synchronization. This test is crucial for evaluating each method's efficiency in handling verification processes and their ability to recognize unchanged data states without unnecessary processing overhead.

3. Insert New Data Scenario

The insert scenario measures performance when adding new records to an existing dataset in the staging area. This test simulates real-world conditions where new data must be integrated with existing records, evaluating each method's capability to handle incremental data loading efficiently.

4. Delete Data Scenario

This scenario assesses the performance of data removal operations, measuring the time required to delete varying volumes of records from the staging area. The test evaluates each method's efficiency in handling data removal while maintaining data integrity.

5. Combination Operations Scenario

The combination scenario tests performance under complex conditions where multiple operation types occur simultaneously. This includes a mix of insertions, updates, and deletions, providing insights into each method's capability to handle diverse operations concurrently while maintaining consistent performance.

2.7. Performance Testing Framework

The testing environment utilizes specific hardware and software configurations selected to simulate enterprise-level educational systems. The hardware configuration includes an Intel Core i7 processor, chosen for its parallel processing capabilities, complemented by 16 GB of RAM to handle large dataset operations efficiently. The storage system employs SSD technology to minimize I/O bottlenecks during intensive data processing operations.

Testing scenarios encompass five distinct cases designed to evaluate different aspects of ETL performance. The Add Data scenario assesses initial load capabilities across varying dataset sizes. Without Difference testing evaluates system efficiency in handling unchanged data, while Insert and Delete operations measure the performance of incremental updates. The Combined Operations scenario tests system resilience under complex, multi-operation conditions.

Each testing scenario is executed with datasets of varying sizes to evaluate scalability and performance characteristics. The dataset sizes are strategically chosen to represent different scales of e-learning operations:

a. Small-scale testing: 150 records
b. Medium-scale testing: 15,000 and 50,000 records
c. Large-scale testing: 65,000 records
d. Enterprise-scale testing: 500,000 and 1,000,000 records

2.8. Analysis and Validation

The analysis phase implements comprehensive statistical evaluation of performance data collected during testing. Execution times are analyzed across different scenarios and data volumes to identify performance patterns and potential bottlenecks. The validation process ensures data consistency and accuracy through multiple verification steps, including cross-scenario result comparison and error rate analysis.

3. RESULT AND DISCUSSION

3.1. Performance Analysis Results

The performance analysis presents execution times for each ETL method across different testing scenarios. The results are organized by method and scenario to enable clear comparison.

1. Add Data Performance

The performance analysis for the Add Data scenario evaluates the execution times of three ETL methods—Table Output, Sync After Merge, and Switch Case—across varying dataset sizes. The results, as summarized in Table 1 and visualized in Figure 5, reveal significant differences in how these methods handle increasing data volumes.

Table 1. Add Data Performance Results (in seconds)

	150 Data	15.000 Data	50.000 Data	65.000 Data	500.000 Data	1 mio Data
Table Output	0.2	0.2	127	170	1502	3373
Sync After Merge	0.4	0.6	74	113	968	2176
Switch Case	0.4	0.6	77	116	988	2239

The Sync After Merge method demonstrates superior performance across all dataset sizes, particularly for larger datasets. For example, it processes 1,000,000 records in 2,176 seconds, approximately 35% faster than Table Output, which requires 3,373 seconds. Switch Case shows comparable performance to Sync After Merge for smaller datasets but becomes slightly slower for datasets exceeding 500,000 records.

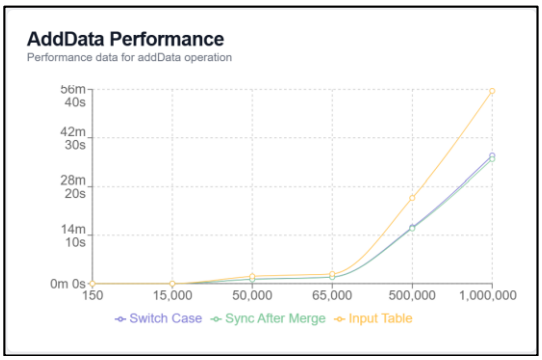

Figure 5. Add Data Performance Comparison Across Methods

This graph highlights the performance trends of the three ETL methods for Add Data operations. It demonstrates the exponential increase in execution time for Table Output, as represented by the yellow line, while Sync After Merge (green line) and Switch Case (blue line) maintain relatively stable performance.

Sync After Merge and Switch Case exhibit near-linear scalability, making them ideal for handling large-scale data integration tasks in real-world scenarios. In contrast, Table Output suffers from exponential performance degradation as dataset size increases. The performance gap between Table Output and the other methods becomes significant at 50,000 records, where Table Output starts to exhibit drastic increases in execution time.

2. Without Difference Performance

Table 2. Without Difference Performance Results (in seconds)

	150 Data	15.000 Data	50.000 Data	65.000 Data	500.000 Data	1 mio Data
Table Output	15.3	15.4	129	-	-	-
Sync After Merge	0.6	0.4	0.7	0.6	3.3	7
Switch Case	0.4	0.4	0.7	3.6	3.7	8.2

The Without Difference scenario examines performance when the source and target datasets are identical, requiring no changes to the data. The results, visualized in Figure 6, highlight the superior stability of Sync After Merge and Switch Case compared to Table Output. Sync After Merge processes 1,000,000 records in only 7 seconds, closely followed by Switch Case at 8.2 seconds, while Table Output fails to complete operations beyond 50,000 records due to inefficiencies in detecting unchanged data.

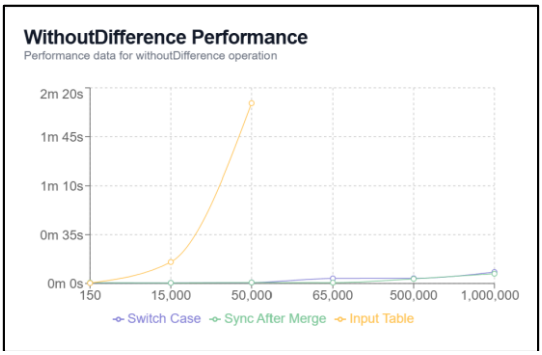

Figure 6. Without Difference Performance Comparison Across Methods

Figure 6 illustrates that Sync After Merge and Switch Case maintain consistent performance across all dataset sizes, while Table Output's execution time increases drastically even for small datasets. These results demonstrate the inability of Table Output to handle synchronization tasks efficiently, making it unsuitable for scenarios requiring frequent data comparisons.

3. Insert New Data Performance

Table 3. Insert New Data Performance Results (in seconds)

	150 Data	15.000 Data	50.000 Data	65.000 Data	500.000 Data	1 mio Data
Table Output	0.3	22.3	140	-	-	-
Sync After Merge	0.4	0.4	8.7	9.7	10.6	14.4
Switch Case	0.4	0.4	8.5	10	10.6	44.6

Insert New Data operations assess the efficiency of adding new records to existing datasets. The results, shown in Figure 7, further underscore the strengths of Sync After Merge. For 1,000,000 records, Sync After Merge completes the operation in 14.4 seconds, compared to Switch Case, which requires 44.6 seconds. Table Output, on the other hand, fails to complete operations beyond 50,000 records.

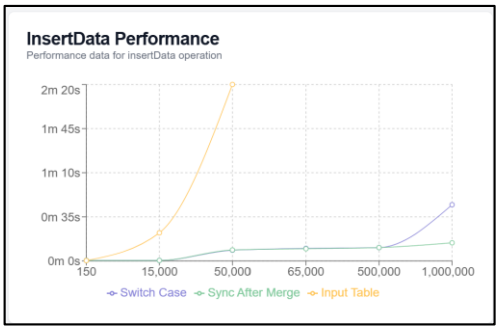

Figure 7. Insert New Data Performance Comparison Across Methods

As shown in the graph, Sync After Merge demonstrates consistent efficiency across all dataset sizes, attributed to its row-level comparison and selective processing mechanism. Switch Case, although initially comparable, begins to show a decline in performance for larger datasets due to its reliance on conditional logic, which adds processing overhead.

4. Delete Data Performance

Table 4. Delete Data Performance Results (in seconds)

	150 Data	15.000 Data	50.000 Data	65.000 Data	500.000 Data	1 mio Data
Table Output	0.2	23	146	-	-	-
Sync After Merge	0.4	0.4	1.3	1.8	10.9	25.4
Switch Case	0.4	0.4	0.5	1.5	12.9	81

The Delete Data scenario evaluates the efficiency of removing records from datasets. As visualized in Figure 8, Sync After Merge maintains efficient performance across all dataset sizes, completing 1,000,000 records in 25.4 seconds. Switch Case, while capable, requires significantly more time, taking 81 seconds for the same operation. Table Output fails to handle datasets beyond 50,000 records, further emphasizing its limitations in scalability.

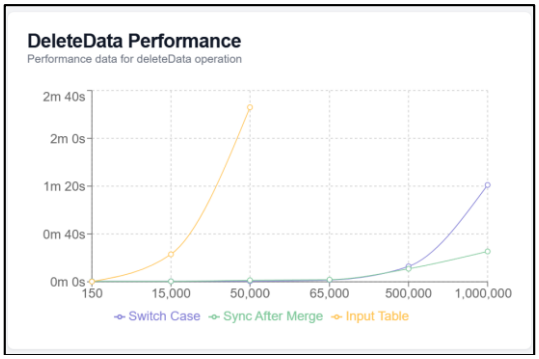

Figure 8. Delete Data Performance Comparison Across Methods

Figure 8 highlights the consistent performance of Sync After Merge, which is able to efficiently identify and remove obsolete records without processing unnecessary data. In contrast, Switch Case shows slower execution due to the increasing computational overhead associated with large-scale deletions.

5. Combination Data Performance

Table 5. Combination Data Performance Results (in seconds)

	150 Data	15.000 Data	50.000 Data	65.000 Data	500.000 Data	1 mio Data
Table Output	0.2	22.8	145	-	-	-
Sync After Merge	0.4	0.4	0.9	1	4	9.2
Switch Case	0.4	0.5	0.9	1.3	5	39.8

Combination Data operations involve simultaneous insertions, updates, and deletions, making this the most complex scenario to evaluate. The results, visualized in Figure 9, show that Sync After Merge processes 1,000,000 records in just 9.2 seconds, significantly outperforming Switch Case, which takes 39.8 seconds. Table Output fails to complete operations for datasets larger than 65,000 records.

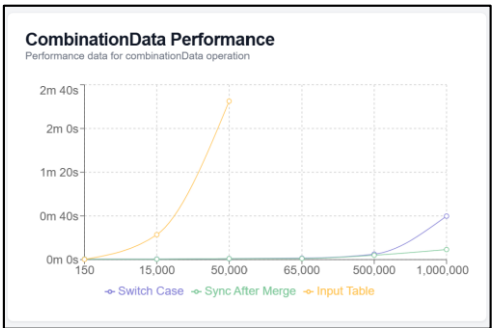

Figure 9. Combination Data Performance Comparison Across Methods

As illustrated in the graph, Sync After Merge's ability to handle mixed operations efficiently highlights its suitability for complex ETL tasks. Switch Case, while maintaining stable performance for smaller datasets, shows a noticeable decline as data volumes increase, attributed to the method's reliance on additional computational layers to handle multiple operations. Table Output's inability to process larger datasets further emphasizes its limitations in scalability.

3.2. Detailed Performance Analysis

A statistical analysis of execution times reveals a strong positive correlation (r=0.94r = 0.94r=0.94) between data volume and processing time for all methods. Table 6 and Figure 10 illustrate the scaling behavior of each method when processing datasets of varying sizes.

Table 6. Performance Scaling Factors Across Methods

Data Volume	Table Output	Sync After Merge	Switch Case
Small (150-15K)	Linear	Linear	Linear
Medium (50K-65K)	Exponential	Linear	Linear
Large (500K-1M)	-	Near-Linear	Sub-exponential

As shown in Table 6, Table Output exhibits exponential degradation in performance as dataset size increases. By contrast, Sync After Merge maintains near-linear scalability, even for datasets up to 1,000,000 records. Switch Case demonstrates similar scaling characteristics to Sync After Merge for small and medium datasets but shows minor inefficiencies as data volumes grow beyond 500,000 records.

3.3. Performance Optimization Factors

The superior performance of Sync After Merge can be attributed to several key factors:

1. Efficient Data State Management

Sync After Merge efficiently maintains data state information, reducing unnecessary comparisons and operations. This optimization is evident in the significantly lower execution times observed in the Without Difference scenario.

2. Scalability Characteristics

The scalability analysis in Table 7 demonstrates the ability of Sync After Merge to handle increasing data volumes effectively. Table 7 presents the execution times for each method in the Add Data scenario.

Table 7. Execution Time Comparison for Add Data Scenario (All execution times in seconds)

Data Volume	Table Output	Sync After Merge	Switch Case
150	0.2	0.4	0.4
15,000	0.2	0.6	0.6
50,000	127	74	77
65,000	170	113	116
500,000	1502	968	988
1,000,000	3373	2176	2239

As illustrated in Table 7, Sync After Merge requires 2,176 seconds to process 1,000,000 records, representing a 35.5% improvement over Table Output, which takes 3,373 seconds. While Switch Case performs similarly for smaller datasets, its efficiency declines slightly for larger volumes.

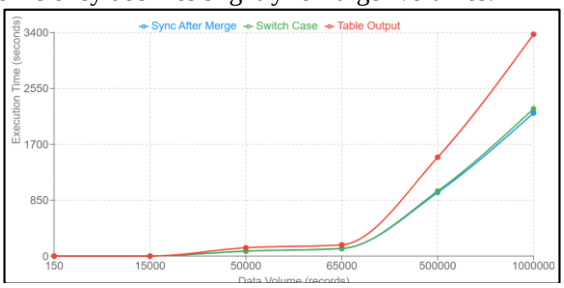

Figure 10. Scalability Analysis Showing Performance Trends Across Data Volumes

The scaling patterns indicate that Sync After Merge maintains the most consistent performance ratio as data volume increases, making it particularly suitable for large-scale e-learning data processing requirements.

3.4. Statistical Performance Metrics

This section highlights execution times for 1,000,000 records across various scenarios. Table 8 and Figure 11 provide a detailed comparison of performance metrics for Add Data, Without Difference, Insert Data, Delete Data, and Combination Data scenarios.

Table 8. Execution Time Comparison for ETL Methods with 1,000,000 Records

Method	Add Data	Without Difference	Insert Data	Delete Data	Combination Data
Table Output	3373	-	-	-	-
Sync After Merge	2176	7	14.4	25.4	9.2
Switch Case	2239	8.2	44.6	81	39.8

Figure 11 highlights the superiority of Sync After Merge across all scenarios, particularly in complex operations like Combination Data, where it completes the task in 9.2 seconds, significantly faster than Switch Case (39.8 seconds). Table Output fails to handle datasets of this size in most scenarios.

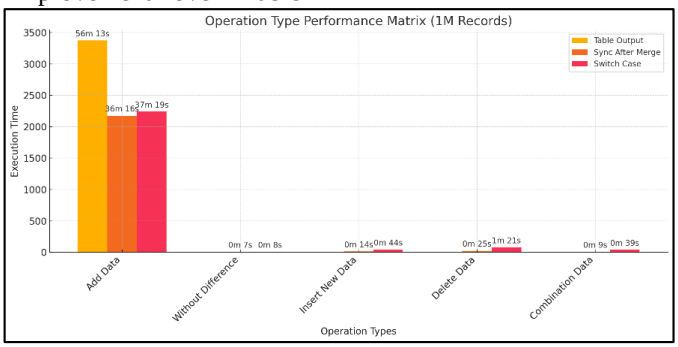

Figure 11. Operation Type Performance Matrix

3.5. Data Volume Thresholds

The analysis identifies optimal volume thresholds for each method:

a. Table Output is suitable for datasets smaller than 50,000 records. Performance degrades significantly for larger volumes.
b. Switch Case demonstrates efficiency when handling datasets up to 500,000 records but starts to show performance limitations beyond this size.
c. Sync After Merge consistently performs optimally across all tested data volumes, making it the best choice for large-scale ETL processes.

3.6. Implications for E-Learning Data Warehouse Implementation

The findings from this study have significant implications for implementing ETL processes in e-learning environments. These implications can be categorized into three main areas:

1. Operational Efficiency

The performance characteristics of Sync After Merge method demonstrate particular advantages for e-learning data warehouses. This method's efficient handling of large datasets aligns well with the periodic data synchronization needs of academic institutions, particularly during peak periods such as semester transitions or academic year closings. The method's ability to process 1,000,000 records in significantly reduced time enables institutions to maintain more frequent data updates without impacting system availability.

2. Resource Optimization

The analysis reveals optimal configurations for e-learning data warehouse implementations, as shown in Table 9. These configurations ensure that institutions of different sizes can effectively manage their datasets, with Table 9 providing specific recommendations for both small and large institutions. The configurations presented demonstrate how standardizing on Sync After Merge enables consistent performance across varying workloads.

Table 9. Recommended Configurations for E-Learning Data Warehouse

Parameter	Small Institution	Large Institution
ETL Method	Sync After Merge	Sync After Merge
Batch Size	1,000 records	1,000 records
Processing Frequency	Daily	Real-time/Hourly
Data Volume Range	<50,000 records/batch	>500,000 records/batch
ETL Method	Sync After Merge	Sync After Merge

4. DISCUSSION

Building on previous research in ETL optimization for complex data warehouse schemas [17], this study demonstrates the critical importance of method selection for e-learning data warehouse performance, particularly when handling large-scale datasets. Among the methods evaluated, Sync After Merge demonstrated superior performance across all scenarios, consistently outperforming Switch Case and Table Output in terms of execution time and scalability.

In scenarios such as Add Data and Combination Data, Sync After Merge completed operations in 36m 16s and 9.2m, respectively, while Switch Case required significantly more time (39.8m for Combination Data), and Table Output failed to process datasets beyond 65,000 records. Research has shown that ETL tools like Pentaho can effectively address data integration challenges when properly implemented [18]. Research on ETL tools has shown that proper data validation and transformation processes are critical for successful data integration, particularly when dealing with diverse data sources and formats [19]. Our findings on ETL method performance align with this perspective, as demonstrated by the superior results of Sync After Merge in handling complex data transformations and validations.

While Switch Case performed moderately well for mid-sized datasets, its efficiency declined significantly as data volumes grew, limiting its scalability. Table Output, on the other hand, proved inefficient and impractical for large-scale operations. These results validate findings in earlier studies, which highlighted the bottlenecks in traditional ETL methods when applied to large datasets.

Research has highlighted the critical importance of thorough ETL testing to ensure data warehouse quality and reliability. Recent studies demonstrate that functional testing approaches, including data quality and balancing tests, are essential for validating ETL implementations and detecting potential faults before they impact warehouse data [20]. Our experimental results align with these findings, showing how proper testing methodologies can help identify performance issues across different ETL methods.

A unique contribution of this study is its operation-specific analysis. For example, Sync After Merge excelled in all operation types, from Without Difference synchronization (7 seconds for 1,000,000 records) to complex deletion tasks (25.4 seconds). These results are particularly relevant for educational institutions aiming to optimize their BI systems by reducing ETL execution times [21]. Additionally, the use of Pentaho Data Integration provides flexibility in designing ETL workflows, supporting diverse operational needs in e-learning data warehouses [22].

Recent research on ETL tools and frameworks has demonstrated various approaches to data integration in enterprise environments [23]. While our study focuses on Pentaho Data Integration, similar findings regarding ETL performance

optimization and data validation have been observed across different integration tools, reinforcing the importance of proper ETL method selection for efficient data processing.

To further illustrate the relevance of this study, Table 4 presents a comparative summary of these prior works.

Table 10. Comparative Summary of Related Studies

Researcher	Method	Dataset	Focus of Study	Notes
S. Vučetić et al. (2023)	SQL vs. SSIS	Multisource academic data	Performance benchmarking of ETL tools	Emphasizes execution speed optimization for diverse datasets
L. Dinesh (2024)	Cloud-based hybrid ETL	Cloud architecture datasets	Performance analysis of hybrid ETL models	Highlights scaling benefits using hybrid optimization, relevant for large-scale datasets
A. Winnetou (2017)	Delta Extraction	Synthetic large datasets	Improving ETL efficiency via delta methods	Offers an approach to historical data synchronization, which overlaps with Sync After Merge

The results of this study have significant implications for the implementation of Business Intelligence (BI) systems in the education sector. By demonstrating the efficiency and scalability of the Sync After Merge method, this research provides actionable insights for academic institutions aiming to optimize their data integration processes.

Frequent and efficient data updates, as enabled by Sync After Merge, allow institutions to maintain up-to-date data warehouses, which are critical for generating timely insights. For example, during critical academic periods such as enrollment or semester transitions, the ability to process large volumes of data quickly ensures uninterrupted operations and informed decision-making. Furthermore, the use of efficient ETL processes reduces system downtime and operational costs, making BI systems more accessible to institutions with limited resources.

Despite these contributions, the study is limited by its exclusive focus on execution time. Other performance metrics, such as memory usage and error handling, were not evaluated. Future work could explore these aspects while also investigating alternative ETL tools like Apache Nifi or AWS Glue, which have been gaining popularity for big data processing [24]. Moreover, the integration of advanced ETL techniques, such as machine learning-based data transformations, could further improve performance and scalability [25].

Additionally, the dataset used in this study was synthetically generated to simulate educational data. While this approach ensures consistency and control over variables, it may not fully capture the complexities and variability of real-world data. Future studies could validate these findings using actual datasets from educational institutions to assess the practical applicability of the proposed methods.

5. CONCLUSIONS

This study analyzed the performance of three ETL methods—Sync After Merge, Switch Case, and Table Output—in the context of e-learning data warehouses. The findings reveal that:

This study demonstrates the superior performance of Sync After Merge in optimizing ETL processes for large-scale educational data. Key findings include:

1. Sync After Merge outperformed Switch Case and Table Output in all scenarios, particularly in execution time and scalability. This method proved to be the most efficient for handling incremental updates and complex operations such as data deletion.
2. The use of Sync After Merge enables educational institutions to maintain real-time data warehouses, ensuring timely insights and efficient decision-making. This is especially critical during peak periods such as enrollment and semester transitions.
3. By reducing processing times and improving resource utilization, Sync After Merge offers a cost-effective and scalable solution for managing growing volumes of educational data.

The recommendations for future work emphasize expanding the scope of evaluation to include additional performance metrics, such as memory usage, CPU consumption, and fault tolerance, to provide a more comprehensive analysis of ETL methods. Future studies should validate the findings with real-world datasets to assess their practical applicability and relevance in diverse educational environments. Moreover, exploring the integration of advanced ETL techniques, such as cloud-based ETL frameworks or machine learning-enhanced data transformations, could further enhance performance and scalability. Investigating alternative ETL tools, such as Apache Nifi or AWS Glue, which are increasingly popular for big data processing, may also offer additional benefits in terms of efficiency and scalability.

REFERENCES

[1] H. D. Surjono, *Membangun Course E-Learning Berbasis Moodle*, 2nd ed. Yogyakarta: UNY Press, 2013.

[2] J. Cole and H. Foster, *Using Moodle: Teaching with the Popular Open Source Course Management System*, 2nd ed. O'Reilly Media, 2007.

[3] M. I. Afandi and E. D. Wahyuni, "Data Warehouse Implementation for University Executive Information System with Speech Command Feature," Apr. 2019, doi: 10.11594/nstp.2019.0222.

[4] L. Dinesh and K. G. Devi, "An efficient hybrid optimization of ETL process in data warehouse of cloud architecture," *Journal of Cloud Computing*, vol. 13, no. 1, p. 12, 2024, doi: 10.1186/s13677-023-00571-y.

[5] I. Wilarso, "Pemanfaatan Data Warehouse Di Perguruan Tinggi Indonesia," *Jurnal Sistem Informasi*, vol. 4, no. 1, pp. 47–54, 2008, doi: 10.21609/jsi.v4i1.244.

[6] V. Rainardi, *Building a data warehouse: With examples in SQL server*. 2008. doi: 10.1007/978-1-4302-0528-9.

[7] A. B. Winnetou, S. A. Wicaksono, and A. Pinandito, "Analisis Peningkatan Performa Proses ETL (Extract, Transform, Dan Loading) Pada Data Warehouse Dengan Menerapkan Delta Extraction Menggunakan Historical Table," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 4, pp. 1366–1371, Aug. 2017.

[8] A. Walha, F. Ghozzi, and F. Gargouri, "Data integration from traditional to big data: main features and comparisons of ETL approaches," *J Supercomput*, vol. 80, no. 19, pp. 26687–26725, 2024, doi: 10.1007/s11227-024-06413-1.

[9] W. Fana, R. Sovia, R. Permana, and M. Islam, "Data Warehouse Design With ETL Method (Extract, Transform, And Load) for Company Information Centre," *International Journal of Artificial Intelligence Research*, vol. 5, Jan. 2021, doi: 10.29099/ijair.v5i2.215.

[10] O. Baker and C. N. Thien, "A New Approach to Use Big Data Tools to Substitute Unstructured Data Warehouse," in *2020 IEEE Conference on Big Data and Analytics (ICBDA)*, 2020, pp. 26–31. doi: 10.1109/ICBDA50157.2020.9289757.

[11] D. Seenivasan, "Improving the Performance of the ETL Jobs," *International Journal of Computer Trends and Technology*, vol. 71, pp. 27–33, Jan. 2023, doi: 10.14445/22312803/IJCTT-V71I3P105.

[12] R. Harris, "Data Warehousing and Decision Support System Effectiveness Demonstrated in Service Recovery During COVID19 Health Pandemic," Jan. 2020, pp. 1–5. doi: 10.1109/ICOSST51357.2020.9333019.

[13] E. M. Haryono, Fahmi, A. S. Tri W, I. Gunawan, A. Nizar Hidayanto, and U. Rahardja, "Comparison of the E-LT vs ETL Method in Data Warehouse Implementation: A Qualitative Study," in *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 2020, pp. 115–120. doi: 10.1109/ICIMCIS51567.2020.9354284.

[14] S. Vučetić, S. Ilić, S. Savić, and S. Ilić, "Comparison of ETL execution speeds when using SQL code and when using SSIS built-in components," pp. 120–126, Jan. 2023, doi: 10.70456/QMXD2316.

[15] D. Seenivasan, "Exploring Popular ETL Testing Techniques," *International Journal of Computer Trends and Technology*, vol. 71, pp. 32–39, Jan. 2023, doi: 10.14445/22312803/IJCTT-V71I2P106.

[16] G. Mhon and N. Kham, "ETL Preprocessing with Multiple Data Sources for Academic Data Analysis," Jan. 2020, pp. 1–5. doi: 10.1109/ICCA49400.2020.9022824.

[17] H. Qin, X. Jin, and X. Zhang, "Research on Extract, Transform and Load(ETL) in Land and Resources Star Schema Data Warehouse," *2012 Fifth International Symposium on Computational Intelligence and Design*, vol. 1, pp. 120–123, 2012, [Online]. Available: https://api.semanticscholar.org/CorpusID:16045132

[18] R. Nath, O. Romero, T. Pedersen, and K. Hose, "High-level ETL for Semantic Data Warehouses," *Semant Web*, vol. 13, pp. 85–132, Jan. 2021, doi: 10.3233/SW-210429.

[19] N. Prasath and J. Sreemathy, "A New Approach for Cloud Data Migration Technique Using Talend ETL Tool," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021, pp. 1674–1678. doi: 10.1109/ICACCS51430.2021.9441898.

[20] H. Homayouni, "Testing Extract-Transform-Load Process in Data Warehouse Systems," in *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2018, pp. 158–161. doi: 10.1109/ISSREW.2018.000-6.

[21] A. Qaiser, M. Farooq, S. M. N. Mustafa, and N. Abrar, "Comparative Analysis of ETL Tools in Big Data Analytics," p. 2023, Jan. 2023.

[22] F. Sudarto, D. Aryani, and Y. Yulianto, "Pengembangan Bussiness Intelegence (BI) Untuk Perusahaan Dalam Membangun Solusi Bisnis Berbasis Open Source," *Journal Sensi: Strategic of Education in Information System*, vol. 1, no. 1, pp. 1–8, Aug. 2015.

[23] J. Sreemathy, I. Joseph V., S. Nisha, C. Prabha I., and G. Priya R.M., "Data Integration in ETL Using TALEND," in *2020 6th International Conference on Advanced Computing and Communication*

Systems (ICACCS), 2020, pp. 1444–1448. doi: 10.1109/ICACCS48705.2020.9074186.

[24] Q. T. Minh, D. T. Thai, B. T. Duc, and N. H. Phat, "Designing a Data Warehouse Framework for Business Intelligence," in *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*, 2022, pp. 498–502. doi: 10.1109/ICDABI56818.2022.10041706.

[25] B. Khan, S. Jan, and W. Khan, "An Overview of ETL Techniques, Tools, Processes and Evaluations in Data Warehousing," *Journal on Big Data*, vol. 6, pp. 1–20, Jan. 2024, doi: 10.32604/jbd.2023.046223.