IMPLEMENTATION OF EXTRACT, TRANSFORM, LOAD (ETL) ON UNIVERSITY DATABASE USING STATE-SPACE PROBLEM

Ramanti Dharayani[1], Kusuma Ayu Laksitowening[2], Amarilis Putri Yanuarfiani[3]

[1, 2, 3] Fakultas Teknik Elektro, Universitas Telkom, Bandung
[1]dharayanii@gmail.com, [2]ayu@telkomuniversity.ac.id,
[3]amarylis.fiani@gmail.com,

Diterima pada 12 Desember 2023; disetujui pada 24 Juli 2024; dan diterbitkan pada 1 Agustus 2024.

Abstrak

Extraction, Transformation, and Load (ETL) adalah salah satu proses dalam data warehouse yang mengumpulkan data dari berbagai sumber. ETL mengolah data mentah menjadi data yang bersih sesuai dengan ketentu-an data warehouse. Proses tersebut umumnya terdiri atas beberapa aktivitas dan membutuhkan waktu yang lama dan memori yang besar. Penelitian ini melakukan implementasi ETL dengan menggunakan workflow masalah state-space pada kasus database universitas. Masalah state-space menggambarkan aliran proses ETL dan menemukan urutan aktivitas dalam proses ETL. Dari hasil uji ETL, rangkaian kegiatan diubah menggunakan transisi graf dan diperoleh hasil yang lebih optimal.

Kata Kunci: extract transform and load, etl, state-space problem, data warehouse, oracle warehouse builder

Abstract

Extraction, Transformation, and Load (ETL) are one of the processes in the data warehouse. The process of ETL is to collect data from various sources. ETL is a process-to-process data into clean data by the provisions of the data warehouse. The ETL process generally consists of multiple activities and requires a significant amount of time and memory. In this final project, an ETL implementation will be carried out using a state-space problem workflow in the case of a university database. The state-space problem describes the ETL process flow and finds the sequence of activities in the ETL process. From the results of the ETL test, the series of activities was changed using graph transitions, and more optimal results were obtained.

Key Words*: *extract transform and load, etl, state-space problem, data warehouse, oracle warehouse builder

1. Introduction

The Higher Education Database (PDPT) is a source of information for Indonesia's higher education quality assurance system. PDPT has activities to collect and process data and information regarding the implementation of higher education in all tertiary institutions by the Directorate General of Higher Education. Currently PDPT change into PDDikti. Based on the university vision, Indonesia Higher Education Hold onto: Teaching, Research-ing, and Community Service, the basic data will be the same. These activities are used to oversee the administration of higher education by the government. To facilitate the activities of collecting, processing data and information regarding the implementation of higher education by the Higher Education, it is necessary to design strategic information such as CIF (Corporate Information Factory) to support evaluation and planning activities by the head of the institution to improve the quality of the institution.

CIF is an information ecosystem structure formulated by WH Inmon. According to Inmon, CIF is a logical architecture that aims to generate business intelligence and business management capabilities derived from data generated by the company's business operations. Among the components of the CIF architecture, the data warehouse is the central point of data integration-the early stages of managing data into information [1]. Data warehouse is a process that involves three major processes, namely basic business processes, ETL (Extract, transform, Load), and business dimensions.

Historical data is needed in the higher education database data warehouse (PDPT) process, not the latest data. Historical information and recent data can be located in various data sources or different databases. Data sources that consist of several types can lead to duplicated data or have other data formats. To handle different formats or duplicated data, ETL (Extract,

Transform, Load) process is needed so that the final data is of high quality. When the information is entered into the data warehouse, the data is in the same format. The ETL process aims to trans-form data according to the data warehouse requirements. The ETL process involves more than one data source. One or more transformation activities are needed to obtain quality final data results when undergoing the ETL transformation process. In general, the ETL process has several problems. The ETL process requires a long execution time. The ETL process also requires significant memory, and changes in the data structure in the ETL cause the ETL process to be repeated. The existence of an ETL process that requires a long time and large memory, there are several choices of activity sequences that will reduce time, cost, and memory in the ETL process. To describe the activities of the ETL process, state-space problems are used to change the sequence of activities performed by transitioning graphs on the previously designed state-space problem.

Because there can be one or more activities in the ETL process, it should be represented by a state-space problem. The state-space problem is one of the general formulas for the problem space intelligence action. The state contains information from the process that causes the effect of an effort to determine the next state and final state [2]. A state-space problem is a collection of sets with configurations and a solution to the problem reached. ETL in university databases can involve many input data processed and is reasonably complex. It is better to use the state-space problem analysis method. State-space problem is an analytical method used to describe a process in detail to document each ETL process. An ETL process that is quite complex can be described with a state-space problem by depicting activities as a graph so that the ETL process is more structured. A structured ETL process can make it easier to detect an error so that any repairs can be handled immediately. In the state-space problem, all activities are depicted in the form of a graph as a node and the direction of the process as an edge. To get op-timal results, nodes can be transitioned, called a graph transition. Input or input from the data source used to meet the PDPT data warehouse's needs is quite complex because it can consist of various data sources. The complexity of the data can be designed using the state-space problem analysis method so that all activities can be seen in detail what processes are carried out to perform ETL in the PDPT data warehouse.

2. Methodology

2.1 Data Warehouse

A data warehouse is a system that can retrieve and consolidate data periodically from various sources into a da-ta store dimension that has been normalized. A data warehouse can store a history of data with queries that will be used in business intelligence or other analytical activities [3]. Business intelligence and other analytical activities need integrated data that can be store in data warehouse [4]

A data warehouse is a relational database designed to implement query and transaction processing analysis. A data warehouse contains historical data derived from transaction data. The data collected in the data warehouse can also come from other sources, such as .sql, .xls, .txt files, etc. Data from various sources can be integrated and have a relationship with one another.

In addition to relational databases, a data warehouse environment includes extraction, transportation, trans-formation, and loading (ETL) solutions, Online Analytical Process (OLAP), client analytic tools, and other appli-cations that manage data and deliver it to business users [5].

2.2 Dimensional Model

A Dimensional model is a table that contains the data structure of the target ETL process. The Dimensional model contains dimension tables and fact tables. The tables are between the back and the front of the process. The dimensional model is a physical mapping from the previous step of moving the table to the end-user environment. The dimensional model is better known as the data structure used by the end-user for querying and analy-sis. Dimensional models are very stable in changes in data and easy to understand by users; dimensional models contain data structures and are relatively fast when querying available relational databases. Dimensional models are used as the basis for building cubes in OLAP [5]. A dimensional model needs to supporting details logical in-formation that distributed in different area [6]

2.3 ETL

ETL (Extract, Transform, Load) is the primary system of a data warehouse. A good ETL design of a data source extraction system, prioritizing data quality and consistent standards, data from separate sources is appropriate to be integrated to provide a data format to be represented. The ETL system is a backbone activity that is not visible to the end-user of the data warehouse. ETL fulfills 70 percent of the resources required to implement and maintain a data warehouse [5]. ETL is also a collection of data preparation processes from OLTP (Online Transaction Process). ETL is the data processing phase from the source data into the data warehouse. The purpose of ETL is to collect, filter, process and combine relevant data from various sources to be stored in a data warehouse.[7] To ensure that the data warehouse that is built has quality data, ETL can be used to ensure that the data being transformed is of good

quality [4].

2.4 State Space Problem

An acyclic graph will model state-space with two elements: nodes and edges, like a regular graph. The nodes in the chart will represent all activities and record sets. A record set is any data store that can provide a flat record schema. In this study, the data stores used are relation tables and record files. At the same time, the edge shows a relationship between providers and users of data, which can connect between the record set and activity or activity with activity. Each node has different characteristics depending on the schemata it has. A definite record set will only have one schema, namely input schemata or output schema, while an activity has at least two schemata, at least one for each schema. An input schema is responsible for bringing data records into the action for processing, and an output schema is responsible for getting data to the following data user. An activity with only one input schema is called unary, while one with two input schemas is binary.

Assume A is a set of activities, RS is a set of record sets, and Pr is a set of provider relationships so that the workflow graph can be denoted G (V, E), where V = A RS and E = Pr. The record set can be divided into two types: RS, which is the data provider (source), denoted by RSs, and RS, which is the data user (target), represented by RSt. An activity A is formally defined by A = (Id, I, O, S), where:

1. The id is a unique identifier for the activity,
2. I is a set of input schemata,
3. O is the output schemata set, and
4. S is an expression in relational algebra or a function that describes the task of an activity. The algebraic operators that will be used include: selection (θ), projection (π), Cartesian product ($\times$), join ($\propto$), aggregation (γ), ordering (θ), union ( ) , difference ($-$), and function application (f). [8]

2.5 Conceptual Model

This section aims to present a conceptual model for ETL activities. The goal is to use object-oriented entities to capture ETL processes. The first step is to define a graphical notation and a metamodel. Then in detail it will explain all the entities in the metamodel. Which consists of one end node and one coordinator node; it is enough to use transparent mode. As for networks composed of many nodes, the use of API mode is a must. The following is a further explanation of transparent mode and API mode.

2.6 Logical Model

The logical model focuses on data flow from the data source to the target or data warehouse. The analytical model describes the technical solution more than implementing the entire ETL process. The overall scenario of ETL involves activities, a set of records, and functions that can be used together with graphs in a linear series and executed sequentially, which is called an architecture graph. The following is the notation of the architecture graph.

2.7 Graph Transition

Graph transition is essential in getting optimal results in the ETL process. Graph transitions function to change the arrangement of nodes in the graph to produce a more optimal workflow. The graph transitions that will be used in this study are as follows [7]:

1. Swap (Figure1(a)).

   This transition can be applied to two unary activities by changing the order of the two activities. The order of A1.A2 activities will change to A2.A1 after swapping and is denoted by SWA(A1, A2).

2. Factorize and distribute (Figure 1(b)).

   Factorize is implemented by involving a binary activity and at least two unary activities with the same functionality but on different paths, and both meet in the binary action. The factorized transition will combine the two unary activities placed after the binary activity. While distribution is the opposite of factorizing, a unary training will be split into two or more actions and placed before the binary activity on a different path. These factorize and distribute swap transitions that change unary and binary activities. Suppose there is a binary activity Ab that gets data from A1 and A2 with the notation (A1.Ab). (A2.Ab). Then factorize is applied to A1 and A2 to become Ab.A(1+2), where A1 and A2 are combined into A(1+2) and then placed after Ab. Factorize is denoted by FAC(Ab, A1, A2), and distribute is represented by DIS(Ab, A(1+2)).

3. Merge and split (Figure 1(c)).

   Merge is implemented by merging two different activities that can be linked based on constraints in the ETL workflow, and the split is applied to separate the merged activities again after the transition applica-tion is complete. For example, if activities A1 and A2 are merged, they will become A(1+2), with the notation MER(A(1+2), A1, A2). And vice versa with split, the notation becomes SPL(A(1+2), A1, A2).

3. Developed System

The system that will be built in this research is the process of extracting, transforming, and loading data from the data source into the data warehouse.

Figure 2 is a specification of the process ETL system carried out in this study. The process takes data from the data source to be extracted into the ETL process; after the data is removed, the data is transformed according to the required requirements. After the data has been changed, the information is entered into the loading process data warehouse.

3.1 Developed System

Step 1

1. Analyzing Data Warehouse Needs with BAN-PT Forms

   In this study, the BAN-PT form is used as information to reference the needs needed to build a data warehouse for Higher Education Databases. The BAN-PT form has seven standards for the Higher Education Database, so it will be analyzed which part is the dimensional scheme and which part is the relational scheme.

2. Define Dimensional Table

   An analysis was carried out as described previously to determine the dimensional table. From the results of the analysis of questions on the BAN-PT form, eight fact tables and 13-dimension tables were selected. The dimensional table schema used in this study is a galaxy schema because one dimension table can be used for more than one fact table.

3. Analyzing Data Source

   After determining the dimensional table, the data source is analyzed. This analysis is carried out to find what data is needed in the data warehouse. The data source is taken from the Telkom University information system and localhost.

4. Creating Dimensional Tables in the Data Warehouse

   After determining which parts are dimensional, a dimensional data warehouse schema is created a dimension table made in Oracle DBMS.

Step 2

1. Creating Conceptual Models and Logical Models

   Before creating a logical model, first, create a conceptual model. When creating a conceptual analysis model, instead, the data sources needed to meet the needs of the data warehouse. When creating a conceptual model, we also analyze the

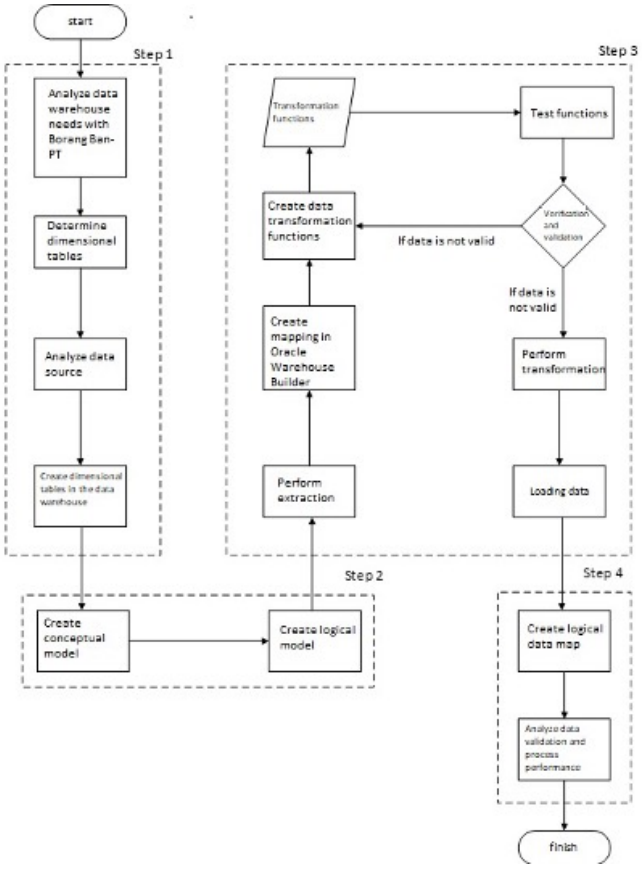

Figure 1. System Design Flow

transformations needed to meet the needs of the data warehouse. From the conceptual model that has been made, the next step is to convert the conceptual model from the dimension table.

Step 3

This step explains the ETL process from extracting data into the Oracle warehouse builder tools to loading data into the target (data warehouse).

1. Perform Data Extraction

   Data is taken from the source and extracted into the oracle warehouse builder in the form of a table. The information is extracted from the data source and then in the form of a table stored in the oracle warehouse builder.

2. Creating a Data Transform Function

   The functions used to perform the transformations are created in the oracle warehouse builder. There are 16 functions used for change. Six of these functions are already available in the oracle warehouse builder.

3. Testing the Data Transformation Function

   The function created to perform the transformation is tested to whether the process can be used to perform the conversion. First, the part that has been made is validated whether there is an error. If there are no errors, the function is used in the mapping so that ETL can be run directly.

4. Mapping in Oracle Warehouse Builder

   After the function has been tested, the procedure is applied to the Oracle Warehouse Builder mapping. The ETL process is performed on Mapping in Oracle Warehouse Builder. Mapping is an implementation of the Con-ceptual and Logical Model. Mapping the ETL process from extracting and transforming to loading data into the data warehouse is carried out.

5. Doing Transformation and Loading

   After completing the mapping in the oracle warehouse builder, the mapping that has been made is validated. If there are no mapping errors, it can be deployed to make sure the mapping we created can be run or not and ensure the oracle warehouse builder service is running. After finishing deploying the mapping, it can be run. In the run process, the transformation and loading are carried out.

Step 4

This step is the last step to analyze the processes that have been carried out and create a logical data map.

1. Creating a Logical Data Map

   Logical data map is the flow used in the ETL process, where the data is obtained and how to transform it. The analytical data map was created because it could be interrupted at the time of implementation if it was not de-signed properly. The analytical data map explains where the data can be obtained from the target and how to fill in the data.

2. Analyze data validation and process performance.

   At this stage, I analyzed data validation and saw the version of the process carried out.

4. Evaluation

This section describes the objectives and test scenarios carried out in this study.

1. Testing Purpose

   The testing in this study aims to determine the feasibility of the previously designed process. At the same time, the tests carried out focus on two things, namely the validity of the process data and process performance.

2. Testing Scenario

   The tests carried out in this study will focus on two things as described previously. The explanation of the test scenario is as follows:

   - Testing the validity of the data mapping before the graph transition and after the graph transition: validation will be carried out on the mapping designed in the oracle warehouse builder. This stage is the primary stage to determine whether the logical model design is a valid and equivalent design to choose the next test.
   - Process performance testing: at this stage, the performance analysis of the ETL process will be ana-lyzed. The performance seen is in terms of time, cost, and memory capacity used when the process is carried out and by visiting whether the selected row in the data source is input all into the data warehouse.
   - Perform comparisons between designs with each other when performing ETL on the dimension table. Of the two or more methods made, a comparison of which way is more optimal is carried out. The designs compared are testing and analysis.

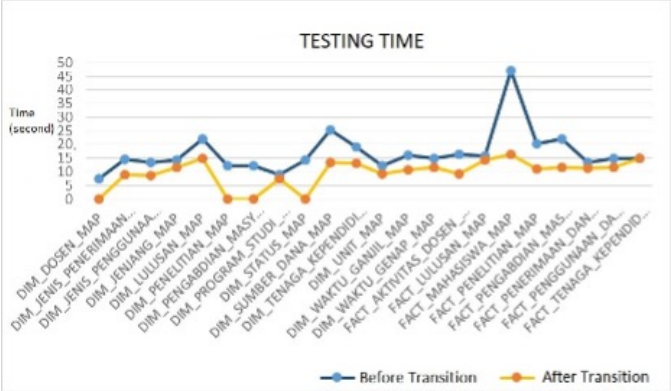

Figure 2. Graph of ETL test results by the time

4.1 Data Evaluation Test

After testing the data validation on the mapping made previously, the next step is to test the data validation for the mapping in which the logical model has been transitioned. Of the 22 analytical models that have been created, 15 logical models can be transitioned.

From the results of data validation tests carried out, all data that has been extracted and transformed is entered into the loading process. Mapping with a logical model before and after the transition and after the change is an equivalent scheme. The results of the ETL evidence this was carried out before the schema is transitioned and after it produces the same results.

4.2 Time-Based ETL Performance Test

After implementing the ETL on the Oracle Warehouse Builder and testing it based on time, 15 mappings can be used for graph transitions in the form of a logical model.

Figure 4 of the 1st time test shows that the blue line is the timeline before the transition, and the yellow line is the line after the change. On the yellow line, there are dots at 0.

4.3 ETL Performance Testing Based on Process Cost

The cost value is obtained from the generalization of the statement when executing the previously designed mapping. The following is the result of ETL performance based on the cost after graph transition. Figure 5 is a graph that illustrates the comparison of tests based on cost.

4.4 Process Performance Testing based on memory usage.

Figure 6 is a graph that shows constant memory usage during ETL.

Figure 7 is a graph showing the memory usage when performing ETL. The chart shows a similar picture because persistent memory and runtime memory is

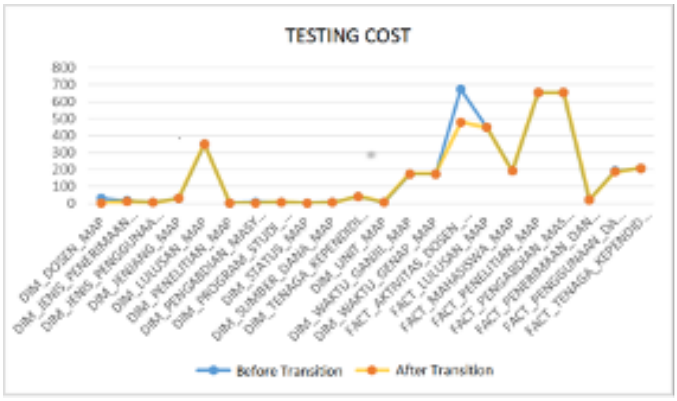

Figure 3. Graph of ETL test results based on cost.

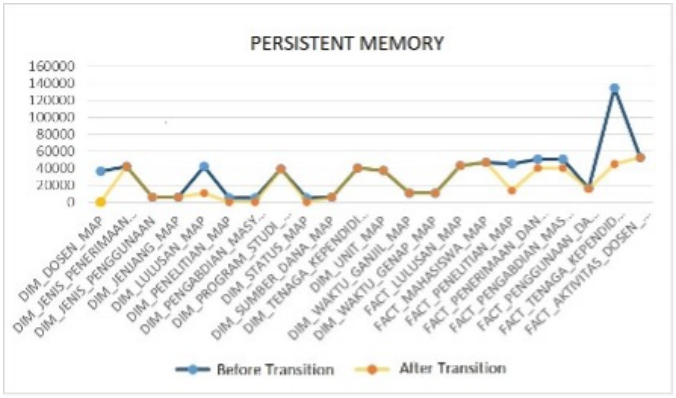

Figure 4. Graph of ETL test results based on persistent memory.

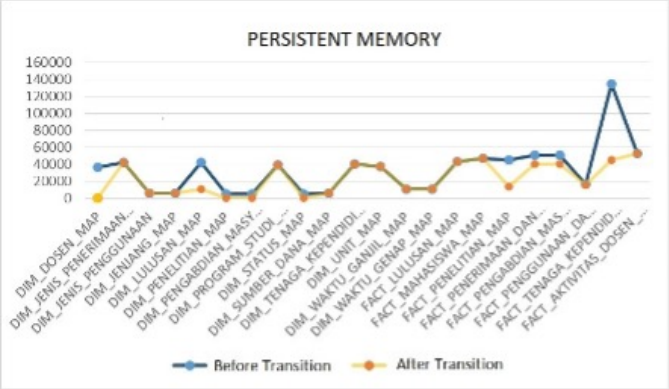

Figure 5. Graph of ETL test results based on runtime memory.

directly proportional. The use of runtime and persistent memory does not have a striking difference. This is because persistent memory depends on the number of columns in the query, while runtime memory depends on the command execution. The more columns executed in an uncompromising memory statement; the more memory will be used. The effect of runtime memory is that the more complex an account is, the bigger the memory used.

4.5 Activity Change Analysis

The tests carried out previously, namely testing based on time, cost (in bytes), and memory usage, show a graph transition in the logical model design, which results in more optimal results than the previous design. In the earlier tests, it can be seen that the time used to carry out the ETL process requires less time when graph transitions are carried out.

The change in time during the graph transition is the effect of the activity on the ETL process. These activities consist of transformation activities which have been described in Chapter II. Query Processing influences the fast or slow ETL process, where the more records accessed, the longer the processing time will be. This happens in the ETL process design that has been made previously.

Figure 8 is an example of a sequence of activities before a graph transition is performed. In this activity, NN (Not Null) activity is an activity to remove null values, and UN(Unique Value) is an activity to remove redundant values. The activity will be faster if a SWAP(NN, UN) graph transition is carried out, namely by changing the sequence of activities by prioritizing the removal of redundant values; this is because if the redundant values are removed first, it will speed up the ETL process. as said before, the more records in a circle, the longer the process and the greater the cost, by changing the sequence of activities, the time used will be faster. The memory usage will be more optimal. Changes in the sequence of activities must be adjusted to other activities not to change the shape of the logical model design. This research will implement with oracle warehouse builder tools while [9] is builded by Pentaho Data Integrator. Both of the tools can handling null value. Previous research focusing on handling null value while in this research focusing which model that have a better model to build.

4.6 Analysis of Test Results

This research focuses on modeling ETL activities as state-space problems by making nodes as activities in the ETL. Because it represents a node as an activity from ETL, a graph transition is carried out on the logical model created previously for ETL implementation to get optimal results. An analytical model that is created must be equivalent. The rational model is comparable if it has the same output and the two schemes for spreading data from source to target are identical. Therefore, the tests contained in section 4.2 ensure that the scheme is equivalent. The comparable scheme results have been obtained from the test results in subsection 4.2; further testing can be carried out. A graph transition on a state-space problem with a relatively complex logical model can be done. An analytical model is complicated if it has more than one sequential activity to reach a goal. For example, it takes more than one process to fulfill a need for one or more attributes in the fact table.

When performing a graph transition, there is a change in the order of execution of the ETL process. To achieve a graph transition, the first thing to do is check the input and output of each state. Then from the states that have been designed, look for alternative sequences with an equivalent scheme. After the new equivalent scheme, graph transitions can be performed. To find out the graph results are more optimal, namely running from the implementation that has been designed. From the results of the tests that have been carried out in section 4.3, section 4.4, and section 4.5, graph transitions on state-space problems can reduce execution time and memory usage during execution from 33.33% to 94.73% of the overall mapping performed by graph transitions. This is caused by the number of records that are executed less in the mapping that has been done with the graph transition: the fewer rows that are accessed, the more optimal the process. In the tests carried out, when the graph transition is carried out, the difference can be seen in the execution time; after the graph transition, the execution time of the ETL process decreases a lot. Meanwhile, the cost and memory usage tend to be the same. This ETL process didn't have a complex query so that the graph transition can decrease execution time of ETL process. However, if we have many processes, and many jobs the ETL process will need distributed process

tools [10]. Based on the [9] they use IFNULL() expression in the Pentaho data integrator script while in oracle warehouse builder Use Not Null Activity. Both of the tools can be used to handling null value.

5. Conclusion

Based on the research that has been done above, the following conclusions can be drawn. The State-space problem method can be implemented in Higher Education Databases' Extract, Transform, and Load (ETL) process. In the ETL process, there are several provisions for the needs of the data warehouse, one of which is eliminating the null value so that there is no null value in the data warehouse. The state-space problem in this implementation applies to a graph transition process created from a logical model. The analytical model after the graph transition is said to be valid if the logical model is equivalent. An analytic model is equivalent if it produces the same data. After testing, it is proven that the mapping is identical before and after the transition is carried out. After the graph transition is done, the performance results of the logical model after the change are more optimal, reaching 46.67-53.33%. In the future, it is necessary to adjust the transformation function in different environments. This research can be developed into performance optimization of ETL based on storage using a similar method. We recommend that you consider using database memory for more than 60GB if you have complex processes and large amounts of data. This research can be developed using databases from different hosts with various file formats.

Daftar Pustaka

[1] C. Imhoff and R. Sousa, *Corporate information factory*. John Wiley & Sons, 2002.

[2] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

[3] S. Server, "Building a data warehouse," 2008.

[4] M. Souibgui, F. Atigui, S. Zammali, S. Cherfi, and S. B. Yahia, "Data quality in etl process: A preliminary study," *Procedia Computer Science*, vol. 159, pp. 676–687, 2019.

[5] R. Kimball and J. Caserta, *The data warehouse ETL toolkit*. John Wiley & Sons, 2004.

[6] M. M. KirMani, "Dimensional modeling using star schema for data warehouse creation," *Oriental journal of computer science and technology*, vol. 10, no. 4, pp. 745–754, 2017.

[7] L. Heilig and S. Voß, "Information systems in seaports: a categorization and overview," *Information Technology and Management*, vol. 18, pp. 179–201, 2017.

[8] A. Simitsis, "Modeling and optimization of extraction-transformation-loading (etl) processes in data warehouse environments," *National Technical University of Athens: PhD Thesis, Athens, Greece*, 2004.

[9] S. Choudhari and M. R. Agrawal, "Optimization design of query processing performance using appropriate materialized view selection & preservation," *Optimization*, vol. 3, no. 12, 2014.

[10] A. A. Yulianto, "Extract transform load (etl) process in distributed database academic data warehouse," *APTIKOM Journal on Computer Science and Information Technologies*, vol. 4, no. 2, pp. 61–68, 2019.