An Automatic Tool to Transform Star Schema Data Warehouse to Physical Data Model

Humasak Simanjuntak, Ardo Nainggolan, Dameria Simatupang, Delia De Venty Manurung
Institut Teknologi Del, Faculty of Electrical Engineering and Informatics,
Information System Department, Sitoluama, Indonesia.
humasak@del.ac.id

***Abstract*—Data warehouse is used to store very large data for supporting company to perform data analysis. Star schema is data warehouse model most widely used by companies today. Sometimes, data stored in star schema need to be exported to conventional model so that others may use them without knowing the OLTP (Online Transaction Processing) or source model, particularly for backup and recovery case. Therefore, this research aimed to transform star schema data model to physical data model. Two cases have been identified case, which are: 1) the star schema with simple star schema and the multi-fact star schema (standard case); and 2) the multi star schema (nonstandard case). There are five processes to build the physical model from the star schema model, namely: 1) finding fact table, 2)finding dimension table, 3) deleting time dimension table, and adding date attribute to fact table, 4) changing fact table to relational table, and 5) changing dimension table to relational table. The prototype was built to implement this phase, and it was tested using some cases. The prototype transformed star schema to physical data model properly (complete design with table, attribute, relation, data type). Some results were different (were not consistent) from the source model because there are many possibilities of star schema for one model, and there is no metadata that are stored when the star schema model was built.**

***Index Terms*—Data Warehouse; Star Schema; Fact Table; Relational Table.**

I. INTRODUCTION

Data warehouse is a collection of integrated data that are subject oriented, time-variant, and integrated, which can be used to support decision making [1] [2]. On the other hand, Data Warehouse can be used for predicting trends and simulating a virtual business scenarios [3]. A data warehouse needs to be modeled before it is created in the Database Management System (DBMS). Data warehouse modelling can be done in several ways, such as normalization, dimensional, and the most recently used at the moment is the combination of normalized and dimensional. Some examples of data warehouse modeling are the Entity Relationship Model and the Dimensional Modeling. There are three forms of dimensional modeling, which are the Snow Flake Schema, Fact Constellations, and Star Schema [4][5].

Snowflake schema is a normalization form of star schema where dimension tables may be connected to other dimension tables besides the fact table. Fact constellation schema is a modeling that consists of multiple fact tables that use one or more dimension tables simultaneously. Star schema is a data warehouse modeling with a fact table centrally connected with several dimension tables. Star schema is the most frequently used models by companies today. Star schema is useful in minimizing join process and speed up queries [6]. The data source for data warehouse is the Online Transactional Processing (OLTP) data. In Enterprise, OLTP is frequently defined in terms of entity-relationship diagrams (ERD). Furthermore, ERD will be converted to Physical Data Model which contains some components, such as table, attribute, primary key, foreign key, etc. Physical data model is created in Database Management Systems and this model will be used by data warehouse designer to design star schema. Finally, data from OLTP will be imported to star schema model after some ETL (Extract, Transform, and Loading) processes.

In implementation, developers only access star schema model. However, sometimes they need to see the physical data model. The physical data model assists developers in understanding the business process that exists in a system. In addition, backup and recovery process can be done quickly in case there are problems with the data warehouse. If developer did not have access to the physical data model, then it is impossible to do these. One way to allow this is data reverse engineering [16]. At this time, there are researches about transformation physical data model into a star schema [7, 8], but there is no research that conducts reverse engineering from star schema to physical data model. The existing research mainly discussed the concept of translating Star Schema into Entity-Relationship Diagrams [6], but its implementation has yet to be conducted.

This research conducted data reverse engineering to transform star schema to physical data model. Reverse Engineering is expected to produce the original physical data model. The result of the transformation is expected to replace the OLTP design, if there is a problem with the original data structure. Moreover, this research showed the consistency between physical data model produced from the star schema.

II. RELATED WORK

Methods on transforming physical data model to data warehouse model and vice versa was implemented by researchers. Krippendorf, M. and Song Il-Yeol has conducted research to translate star schema into Entity Relationship Diagram. This research is still at its conceptual stage (did not implemented yet), although it has already defined some rules to transform star schema with standard case (simple star schema with one fact table, multiple fact table) and star schema with Non-Standard case (multi-star schema) [6]. Rules defined by Krippendorf, M. and Song Il-Yeol were used as a primary rule for transforming star schema to physical data model.

Yen-Ting Chen, Ping-Yu Hsu proposed an algorithm to translate an entire ER Diagram into multidimensional with hierarchical snowflake structures. Some points about the algorithm are: (1) the algorithm systematically performs ER Diagrams to multidimensional model translation, given the identified fact table, (2) the algorithm guarantees that adding a new entity to the structure does not change the grain of the existing entities, (3) the snowflake structure proposed by the algorithm takes the fewest relationship to connect dimensions and the given fact table [9].

Golfarelli, M proposed a semi-automated methodology for building data warehouse from the pre-existing Entity/Relationship schemes based on the description of a database [10]. The model consists of tree-structured fact schemes whose basic elements are facts, attributes, dimensions and hierarchies. In this research, some features represented on the fact schemes are the additivity of fact attributes along dimensions, the optionality of dimension attributes and the existence of non-dimension attributes.

Furthermore, related research regarding automatic transformation tools of star schema such as SAMSTAR [11], BIRST [12] are available. Simanjuntak et al. also did a transformation from Entity Relationship Model into Star Schema. This research built rules and tools by identifying fact table from physical data model derived from Entity Relationship Model. The Fact table was used as a basic knowledge for dimension table [6]. Lumbantoruan et al. offered an approach to automatically generate star schema from user business key(s). It started by a syntactical parsing process of user business to identify noun words used to generate dimension table candidates and a fact table [8].

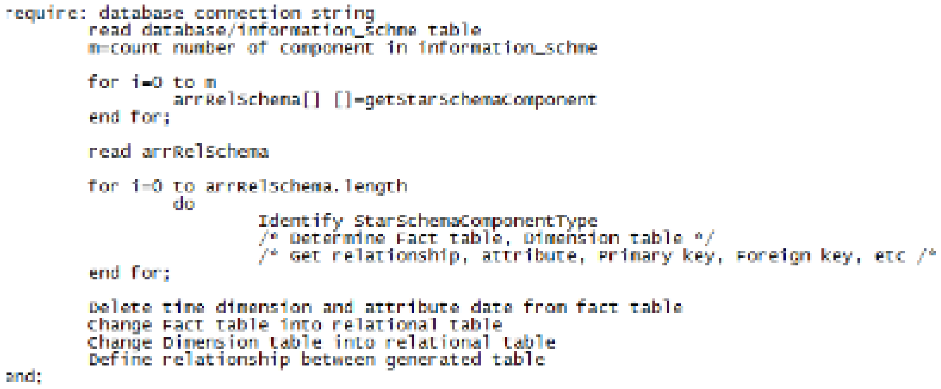
III. METHODOLOGY

In this paper, we proposed a method to transform star schema into physical data model. This method uses rule generated in paper [6] as the main reference. The rule was improved so that it can perform transformation from star schema to physical data model. The differences between the proposed method and the method that had already been developed in paper [6] can be seen in Table 1 below.

Table 1
Differences between the proposed method and the method in paper [6]

No	Proposed Method	Method in paper [6]
1	Transform star schema into physical data model	Transform star schema into ER Diagram
2	Identify fact table and dimension table	Does not identify fact table and dimension table
3	Delete time dimension	Does not handle time dimension
4	Translate Fact table as a transactional relational table.	Translate each fact table as an n-ary relation to associate the principle dimension tables
5	Translate Dimension table as a relational table directly connected to the fact table/transactional table	Translate each dimension table into an entity
6	In non-standard case, there is a new table as a connector between dimension as owner of weak entity and fact table itself.	In non-standard case, there is a new relation between dimension as owner of weak entity and fact table itself.
7	Implement in a tool as prototype.	Only concept to ER Diagram. Has not been implemented.

The overall algorithm of the proposed method can be seen in Figure 1.



```
require: database connection string
        read database/information_schme table
        m=count number of component in information_schme

        for i=0 to m
                arrRelSchema[] []=getStarSchemaComponent
        end for;

        read arrRelSchema

        for i=0 to arrRelSchema.length
                do
                        Identify StarSchemaComponentType
                        /* Determine Fact table, Dimension table */
                        /* Get relationship, attribute, Primary key, Foreign key, etc /*
        end for;

        Delete time dimension and attribute date from fact table
        Change Fact table into relational table
        Change Dimension table into relational table
        Define relationship between generated table
and;
```

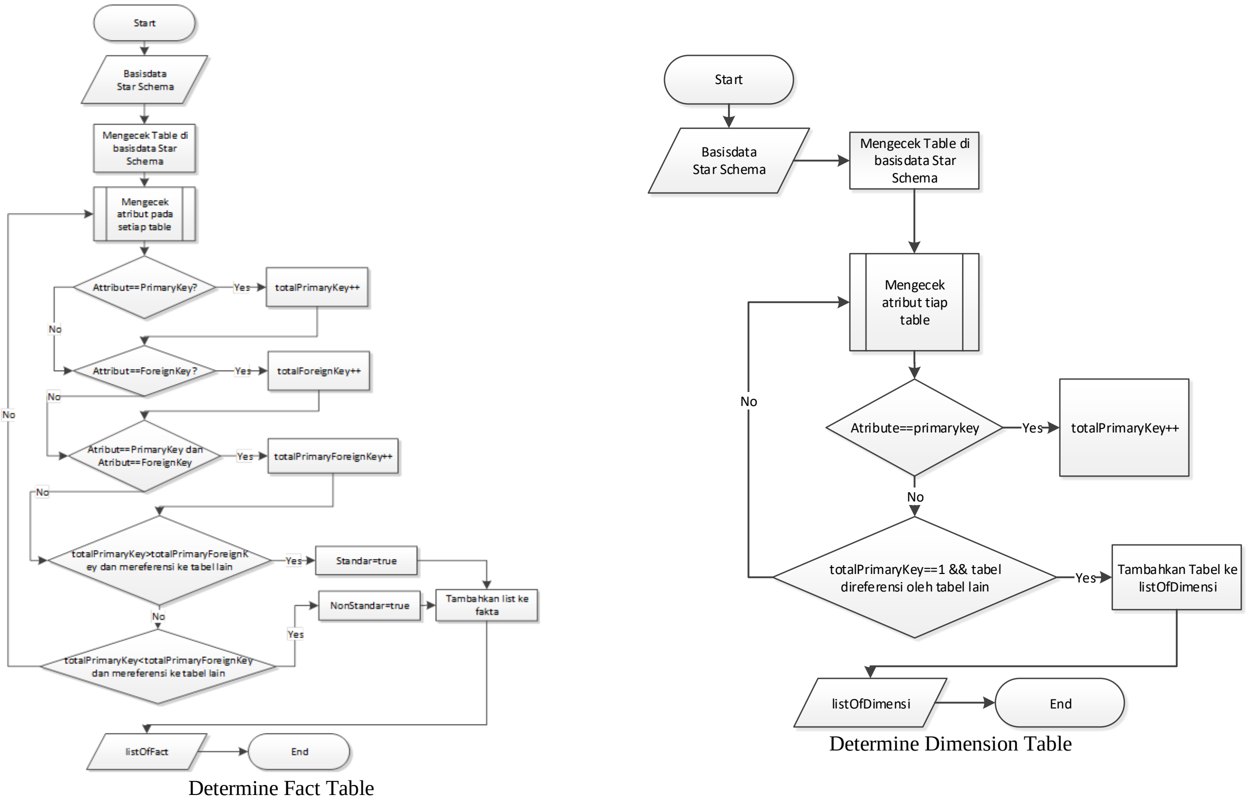
Figure 1: Overall algorithm to transform star schema into physical data model

Database connection string is an input for the algorithm. It is a database that contains star schema model. All table, relation, and constraint in star schema will be identified by reading the information schema table and stored in an array variable. Information schema is a metadata table created by SQL Server when users create new database and new object in database. Then, the component of star schema will be decided based on the rule. The proposed algorithm deletes the time dimension and date attribute from the fact table. All tables identified as fact will be transformed into a transactional table in the physical model and all dimension table will be transformed as a table. The detailed algorithm or rule to identify the fact table and dimension table from star schema can be seen in Figure 2.

The flowchart in Figure 2 shows that the star schema database becomes an input of the algorithm. All table will be read and the constraint in each table attribute is checked, whether the attribute is a primary key, foreign key, or it plays for both primary and foreign key. If the number of primary key is equal to the number of attribute that plays for both primary and foreign key, the table refers to another table, then the table is included in a list of fact and it is concluded as a standard case. If the number of primary key is less than the number of attribute that plays for both the primary and foreign key, the table will be included in a list of fact and it is concluded as a non-standard case. The output of this algorithm is the list of Fact table.

Identifying Dimension table starts by reading all tables in the star schema database. The constraint for each attribute will be checked. If the number of primary key attribute in star schema database is equal to one, the table is referred by another table. This table also has one or more foreign keys from the other tables, and it will be included in the list of primary dimension. Furthermore, the proposed method also tries to find a secondary dimension table (dimension that directly connected with first dimension table).

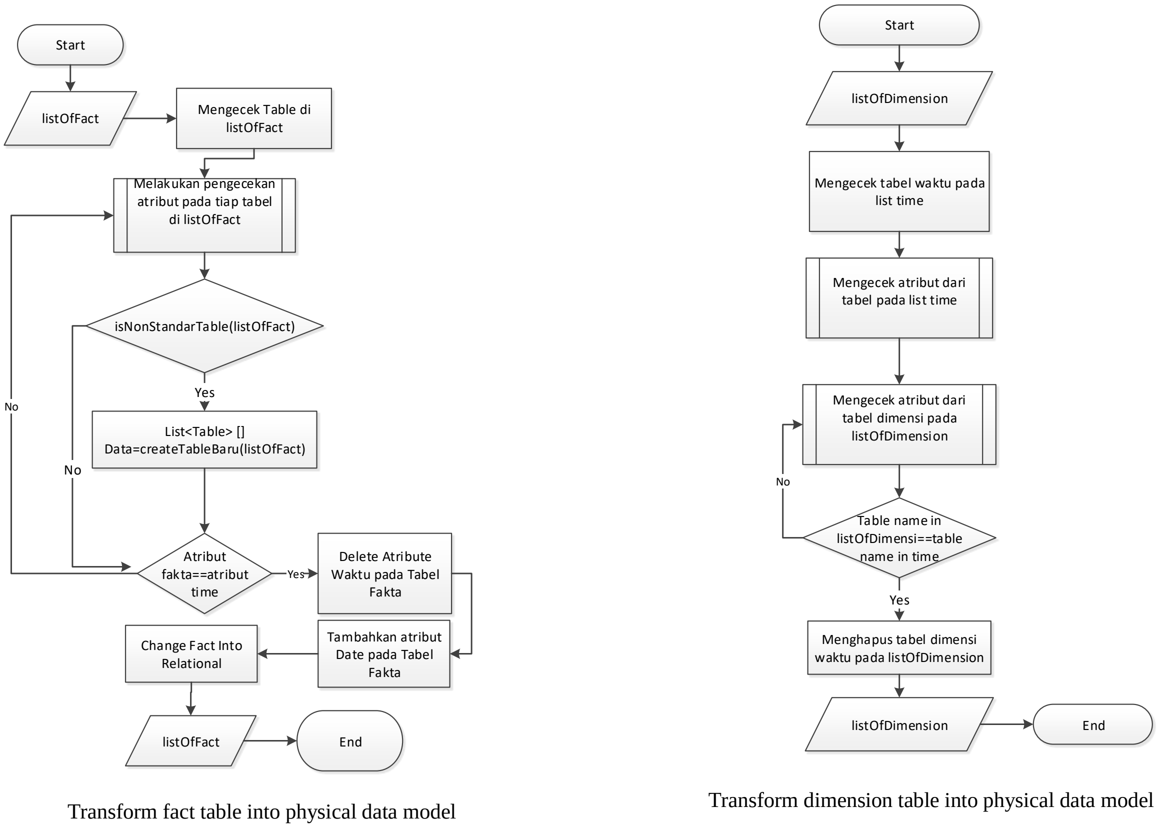
After the fact and dimension table is identified, it will be transformed into a relational table. The algorithm to transform star schema data model to physical data model (relational model) can be seen in Figure 3.


Start
Basisdata Star Schema
Mengecek Table di basisdata Star Schema
Mengecek atribut pada setiap table
Attribut==PrimaryKey?
Yes
totalPrimaryKey++
No
Attribut==ForeignKey?
Yes
totalForeignKey++
No
No
Atribut==PrimaryKey dan Atribut==ForeignKey
Yes
totalPrimaryForeignKey++
No
totalPrimaryKey>totalPrimaryForeignKey dan mereferensi ke tabel lain
Yes
Standar=true
No
NonStandar=true
Yes
Tambahkan list ke fakta
totalPrimaryKey<totalPrimaryForeignKey dan mereferensi ke tabel lain
listOfFact
End
Start
Basisdata Star Schema
Mengecek Table di basisdata Star Schema
Mengecek atribut tiap table
No
Atribute==primarykey
Yes
totalPrimaryKey++
No
totalPrimaryKey==1 && tabel direferensi oleh tabel lain
Yes
Tambahkan Tabel ke listOfDimensi
listOfDimensi
End

Determine Fact Table

Determine Dimension Table

Figure 2: Algorithm flowchart for identifying Fact and Dimension table in Star Schema


Start
listOfFact
Mengecek Table di listOfFact
Melakukan pengecekan atribut pada tiap tabel di listOfFact
isNonStandarTable(listOfFact)
Yes
List<Table> [] Data=createTableBaru(listOfFact)
No
No
Atribut fakta==atribut time
Yes
Delete Atribute Waktu pada Tabel Fakta
Tambahkan atribut Date pada Tabel Fakta
Change Fact Into Relational
listOfFact
End
Start
listOfDimension
Mengecek tabel waktu pada list time
Mengecek atribut dari tabel pada list time
Mengecek atribut dari tabel dimensi pada listOfDimension
No
Table name in listOfDimensi==table name in time
Yes
Menghapus tabel dimensi waktu pada listOfDimension
listOfDimension
End

Transform fact table into physical data model

Transform dimension table into physical data model

Figure 3: Algorithm flowchart for transforming fact and dimension table into physical data model

Figure 3 shows each step to transform star schema model to physical data model. The first step is to transform the fact table into physical data model. This algorithm uses the list of fact resulted from the algorithm to determine the fact table. On a standard case, the primary key of the fact table is a collection of foreign keys from the dimension table related to the fact table. This characteristic is similar with the characteristic of the table that appears from the many-to-many relationship in the physical data model. So, the fact table is directly drawn as a table in physical data model.

IV. Results and Discussion

Since the focus of this research is to propose algorithm and tool that automatically transform star schema into physical data model, we conducted a set of experiments using some examples of star schema. The summary of each model is:

(i) Standard case
We used 4 examples of star schema: Northwind Star Schema, Voyage Star Schema, Library star schema, and Project star schema.

(ii) Non-Standard case
We used 3 examples of star schema: product transaction star schema, restaurant star schema, and Complex Voyage Star Schema.

We designed a C#.Net-based tool that takes database connection string as input. The database must be created in SQL Server. The tool read information schema to get all metadata of star schema table. The tool implemented the proposed algorithm. The example of tool interface can be seen in Figure 3 and the example of physical data model generated from Complex Voyage Star Schema can be seen in Figure 4.

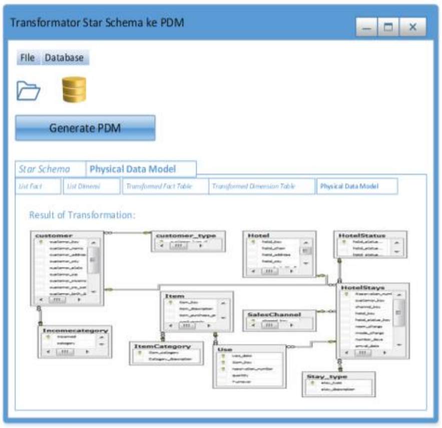


Figure 4: The interface of tool for transforming star schema into physical data model

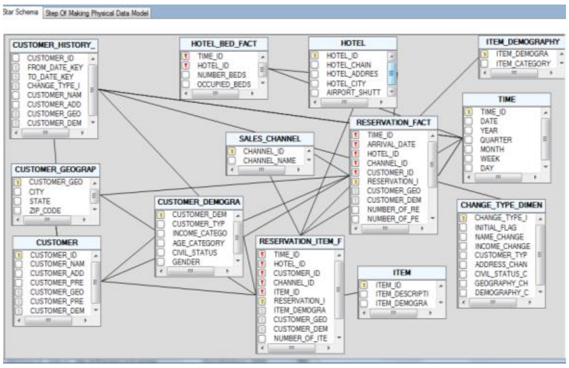


Figure 5: The example of Complex Voyage Physical data model that generated from Complex Voyage star schema.

The experiments of application using the seven examples of star schema have resulted in 10 tests. The result can be seen in Table 2.

Table 2
Experiment Result

No. Test	Star Schema	Experiment Result
A1, A2, A3, A6	Multiple Fact Table Sales, Reservation System Star Schema, Library Star Schema	Result accepted. Physical data model has same structure with OLTP data source. Time dimension deleted and date attribute added to the fact table.
A4, A5, A7	Northwind Star Schema, Voyage Star Schema, Project Star Schema	Result not accepted. Physical data model has different structure with OLTP data source. Time dimension deleted and date attribute added to the fact table.
A8, A9, A10	Product Transaction Star Schema, Restaurant Star Schema, Complex Voyage Star Schema	Result not accepted. Physical data model has different structure with OLTP data source. Time dimension deleted and date attribute added to the fact table. Add new table for weak entity.

Based on the experiment result, there are a few things that must be considered in the tool for transforming star schema to physical data model:

(i) Standard Case
1. The number of attribute and the name of attribute in physical model is similar with star schema.
2. The number of table in physical data model is different with the number of table in star schema, because time dimension table was deleted in physical data model
3. All tables in physical data model have relation to the other table.
4. The physical data model conforms with relational data model theory

(ii) Non-Standard Case
1. The number of attribute and the name of attribute in the physical model is similar with star schema.
2. The number of table in physical data model is different from the number of table in star schema, because time dimension table was deleted in physical data model
3. All tables in physical data model have relation to the other table.
4. The physical data model conforms with relational data model theory

There was some experiment results that showed the test results were not accepted for some case (A4, A5, A7, A8, A9, A10). This situation occured because the transformation result was different from the original OLTP data model. The proposed algorithm cannot give perfect transformation for complex star schema that has multiple fact table and multi star schema. Test A1, A2, A3, and A6 showed the result was accepted, because it was a simple star schema and simple multi fact star schema, hence the algorithm can easily transform the physical data model.

V. Conclusion

In this paper, an algorithm and a tool for transforming star schema to physical data model automatically has been suggested. Based on the experiments, the tool successfully transformed star schema to physical data model. However, the application did not give consistent physical data model for each case of star schema. This is because there was no metadata that stored information about the original data source (OLTP). When the data warehouse designers generate the star schema, they should store the metadata of data source, so that consistent physical data model for different star schema can be easily achieved.

In the data warehouse, data warehouse designer can add new attributes based on the attribute from the data source. Therefore, it is possible to get different physical data model for this star schema, with the exception that this process is stored in metadata

In the future, an intelligent application to get a consistent physical data model from star schema is needed. The application will enable user to transform physical data model to star schema and vice versa. When transforming the physical data model to star schema, all process will be stored in metadata. This metadata will help data warehouse designer to do reverse engineering from the star schema to physical data model to get the original data source model.

References

[1] Inmon, W.H. *Building the Data Warehouse*. 3rd ed. 2008. New York: Wiley Publishing Inc.

[2] Chauduri S, Dayal U. 1997. An Overview of Data Warehousing and OLAP Technology. *Microsoft Research*. ACM SIGMOD Record, New York: 65-74.

[3] Bebel B, Eder J, Koncilia C, etc. 2004. Creation and Management of Versions in Multiversion Data warehouses. *SAC '04 Proceedings of the 2004 ACM symposium on applied computing.* New York: 717-723.

[4] Rainardi Vincent. 2008. *Building a Data warehouse: With Examples in SQL Server 2008*. New York: Apress.

[5] Ballard Chuck, Herreman D, Schau D, Bell R. 1998. *Data Modeling Techniques for Data Warehousing.* California: IBM Corp.

[6] Krippendorf, M. and Song Il-Yeol. 1997. The Translation of Star Schema into Entity-Relationship Diagrams. DEXA '97 *Proceedings of the 8th International Workshop on Database and Expert Systems Applications.* 390.

[7] Simanjuntak Humasak, Pangaribuan Andreas, Nababan Daniel, Sihotang Rina. 2012. Transformator Entity Relationship Model to Star Schema. *National Proceeding in National Conference ICT-M Politeknik Telkom*. ISSN: 2302-1896.

[8] Lumbantoruan Rosni, Sibarani Elisa, Sitorus Monica, Mindari Ayunisa, Sinaga Suhendrowan. 2014. An Approach for Automatically Generating Star Schema from Natural Language. *TELKOMNIKA Journal*. 12(2): 501-510.

[9] Yen-Ting Chen, Ping-Yu Hsu. 2005. An Efficient and Grain Preservation Mapping Algorithm: from ER Diagram to Multidimensional Model. *Proceedings of the 5th international conference on Advanced Distributed Systems*. Springer-Verlag Berlin: 331-346.

[10] Golfarelli, M. Maio, D. Rizzi, S. 1998. Conceptual Design of Data Warehouses from E/R Schema. *Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences. IEEE Computer Society Washington*. 7: 334.

[11] Song IY, Khare R, Dai B. 2007. SAMSTAR: A Semi-Automated Lexical Method for Generating Star Schemas from an Entity-Relationship Diagram. 10th ACM Int'l Workshop on Data Warehousing and OLAP (DOLAP 2007). ACM New York: 9-16.

[12] http://www.birst.com/product/technology/data-warehouse-automation. Automatic star schema generation, BIRST.

[13] Il Yeol Song. Ritu Khare. Bing Dai. 2007. SAMSTAR: A Semi-Automated Lexical Method for Generating Star Schemas from an Entity-Relationship Diagram. DOLAP '07 Proceedings of the ACM tenth international workshop on Data warehousing and OLAP. ACM New York: 9-16.

[14] Luca Cabibbo . Riccardo Torlone. 1998. A Logical Approach to Multidimensional Databases. *EDBT '98 Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology*. Springer-Verlag London: 183-197.

[15] James Dullea. Il-Yeol Song. Ioanna Lamprou. 2003. An analysis of structural validity in entity-relationship modeling. *Journal Data & Knowledge Engineering*. 47(2): 167 – 205.

[16] Muller Hausi, Jahnke Jens. 2000. Reverse Engineering: A Roadmap. *Proceedings of the Conference on the Future of Software Engineering*. ACM New York: 47-60.