An Automatic Tool for Anchor Model Data Warehouse Development

Humasak Simanjuntak, Marcella Tambunan, Kamaruddin Manullang, Hanna Panjaitan

Abstract— **Anchor modeling is a modern, agile, and 6NF normalization for data warehouse modeling technique suited for information that changes over time both in structure and content (historization of data store modification). At the moment, there is no tool used to generate anchor modelling structure from data sources relational automatically. Therefore, proposed AncRel v1.0 algorithm to produce anchor model structure based on relational data sources is presented. The algorithm is started by reading all components on the relational data source, identifying anchor component, and producing a complete anchor model structure. The evaluation shows that AncRel v1.0 generated complete anchor model structure successfully.**

Index Terms—**anchor model, data warehouse, 6NF normalization, historization.**

I. Introduction

Data warehouse is a collection of integrated data that subject oriented, time-variant which can be used to support decision making [1][2]. On the other hand, Data Warehouse may be used for predicting trends and simulating a virtual business scenarios [3]. This kind of data processing is often called the what-if analysis. A data warehouse need to be modeled before created in Database Management System (DBMS). Data warehouse modelling can be done in several ways such as normalization, dimensional, and the most recently model is the combination of normalized and dimensional. Some examples of data warehouse modeling are Entity Relationship Model, Anchor Modeling, Data Vault, Star Schema, Snow Flake, and Fact Constellations [4][5].

Maintenance of data warehouse by using any modeling technique needs to be done so that data is useful for organization. Maintenance is quite complex because data warehouse requires stability and consistency for storing data in different time. Therefore, data warehouse structure or data warehouse modeling must be flexible and able to handle data modification without deleting the current data [3][6]. Data modification can be categorized as: (1) content changes, i.e. insert/update/delete records, and (2) schema changes, i.e. add/modify/drop an common attribute as reported in [7][8]. Both types of modifications may lead to schema changes in a data warehouse.

Data warehouse modeling which support normalization concept (such as ER modeling, Star Schema, Snowflake, and Fact Constellations) have a weakness in terms of flexibility and the ability to handle data modification while Data Vault as a modern data warehouse modeling handle data modification with some limitations.

At the moment, there are some researches about data modification in data warehouse. As in the study [3] proposed multi version of data warehouse that handle schema changes. There are two different kinds of versions: (1) real versions, which handle changes made to data source, and (2) alternative versions, which handle changes made by a user directly in a data warehouse for the purpose of applying the what-if analysis. Others research also proposed multi version data warehouse based on multidimensional structure enhanced by temporal and versioning extensions. It requires several integrity constraints to maintain the consistency of its structure and data. A set of multi version data warehouse constraints are defined under three classes: structural constraints, temporal constraints and versioning constraints [9].

The popular research for data modification management in data warehouse is Anchor modeling. Anchor modeling is an agile and modern data warehouse modeling technique which suitable for information that changes over time, either the structure or content [10]. Anchor modeling (normally in a sixth normal form (6NF)) offers a method that extend or add information to the data, but does not give damage to the structure of data, as well as the accuracy and flexibility of data. A key benefit of Anchor Modeling is that modification in a data warehouse environment only require extensions, not modifications to the data warehouse. Such changes, therefore, do not require immediate modifications of existing applications, since all previous versions of the database schema are available as subsets of the current schema. Anchor modeling has four basic symbols, are [10]:

1) Anchor: An anchor represents a set of entities that is used to deal with the identity of an entity in the data warehouse.
2) Attribute: Attributes are used to describe the properties of the anchor. There are four types of attributes are static attributes, historized attributes, knotted static attributes, and knotted historized attributes. Static attributes are property of the entity (anchor) which does not require to keep historical changes of any attribute values. Historized attribute indicates that attribute value modification need to be recorded. Knotted static attribute indicates the relationship between the anchors and knots. Knotted historized shows the relationship with knots value is unstable and may change over time.

Manuscript received August 7, 2015. This paper with title "An Automatic Tool for Anchor Model Data Warehouse Development" was supported by Del Institute of Technology.

Humasak Simanjuntak, Marcella Tambunan, Kamaruddin Manullang, Hanna Panjaitan are with the Information Systems Program Study, Del Institute of Technology, Laguboti, Indonesia. e-mail: humasak@del.ac.id, if10016@gmail.com,kamaruddinmanoellang@gmail.com, hannaif10070@gmail.com.

3) Knot: A knot shows a property with fixed value, usually small and do not change over time. Knot are used to manage properties that are shared by many instances of several anchors. Example knot is GENDER property, which only has two values of women and men. In anchor modeling, Knot minimize redundancy of fixed property.

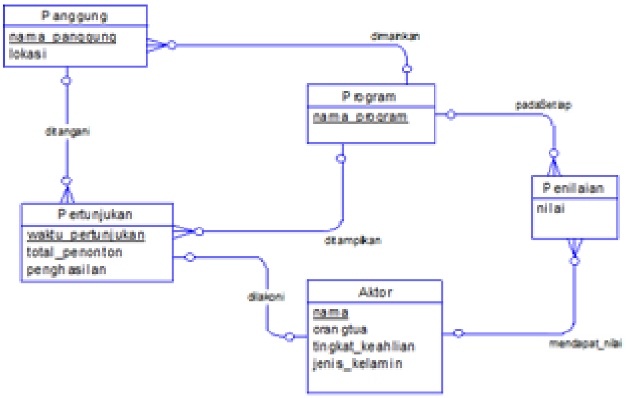


Fig. 1. Example of Conceptual Data Model

4) Tie: A tie representing a relationship between two or more anchor entities and optional knot entities. There are four types of tie are static, historized, knotted static, and knotted historized. Please see the example of conceptual data model (CDM) of stage setting in Fig. 1.

Based on anchor modeling guideline, the anchor modeling design for CDM in Fig. 1 can be seen in Fig. 2.

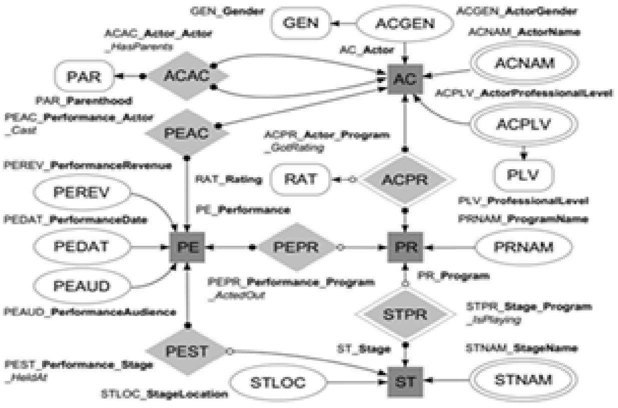


Fig. 2. Anchor modelling design for CDM in Fig. 1

It can seen in Fig. 2 that there are 4 anchors modeling design: PE_Performance, ST_Stage, PR_Program, and AC_Actor which describe its entity. The attributes consist of PR_NAM_Program_Name and PE_DAT_Performance_Date which describe the property of entity. Some attributes AC_NAM_Actor_Name and ST_NAM_Stage_Name as historized is used to illustrate the changes of facts. Fact in Actor that have Gender with two possible values depicted with knot GEN_Gender and a knotted attribute AC_GEN_Actor_Gender. Professional Level is an attribute with static value so that PLV_Professional Level decide as knot and AC_PLV_Actor_Professional Level as knotted attribute with applying historized to record all changes in the attribute [10].

Relationship between anchors is depicted through tie. In Fig. 2, all ties are PE_in_AC_was Cast, PE_was Held_ST_at Location, PE_at_PR_was played, ST_at Location_PR_is_Playing, AC_part_PR_in_RAT_got, and AC_parent_AC_child_PAT_having. Historized tie is ST_at Location_PR_is Playing was used to stored fact about stage changes for each program. AC_part_PR_in_RAT_got is tie knotted which showed Actor whose got rating for each program played and historized stored the changes of rating [10].

Anchor modeling offers a solution to the problem of data warehouse, which is not shared by other modeling. Particularly in insertion process, modification of model are potential to raise problem for the system.

Generally, data source or transactional data for data warehouse stored in a relational database management system. Data warehouse designer needs more time for designing a data warehouse based on the structure of data source (relational database). At the moment, there is no concepts, algorithms, and any tools to transform relational database to an anchor modeling automatically. The only available tool is an Anchor Modeler [11]. The Anchor Modeler is an open source database modeling tool. It is developed by Lars Rönnbäck and Ville Krumlinde and is licensed under the MIT Open Source License.

Based on this fact, data warehouse designer have difficulties in generating data warehouse model. Therefore, in this research, the concepts, algorithms, and tool to transform relational database (transactional data) to anchor modeling called AncRel v1.0 is proposed.

II. RESEARCH METHOD

Type of our research is mainly analytical research to find solution for helping user build an anchor model based on the transactional relational database easily. Our research was conducted by applied approaches by following steps: (i) Conducting literature study on relational model and data warehouse concept, specifically on anchor modeling. At this stage, learning and understanding of those concepts was by reading and understanding papers, text book that available on those fields. (ii) Analyzing the process for generating anchor model from transaction relational database to implement several findings in defining solution to answer the research problem. (iii) Developing a tool which consists of basic software engineering steps such as application analysis, design, and implementation to build the final application. (iv) Evaluating tool by using existing databases in SQL Server. We used three databases as the sources for data warehouse to be built. The evaluation was done by analyzing the anchor model result based on transactional database which was entered into the application. Based on those results, we were evaluating the result manually whether the anchor model is accurate or not. (v) Concluding the evaluation result in order to summarize the research question and problem was able to answer or not.

III. PROPOSED ALGORITHM

In this section, we explain our analysis results and proposed algorithm to generate anchor model from transaction relational database automatically. The proposed algorithm was resulted based on the rules for translating an anchor schema into a relational database schema [12].

By paying attention to the components of anchor modeling that described in section 1, the process of transforming relational tables to anchor model can be done. First, the transformation process is done by getting the database information through the information schema in SQL Server. The information are table, attributes, relations between tables with primary key and foreign key, and relation cardinality. Based on analysis, the process of transformation has been done by following some steps below:

A. Determine Anchor

In accordance with the rules of anchor modeling, any entity or any transaction which have attribute is potential to become an anchor. In a relational database, the entity is database tables. A table can be a strong entity, weak entity, and entity from many to many tables. Based on analysis, the following rule is used to create Anchor component: Strong entity will be the anchor in anchor modeling; Weak entity which depend on two tables will be the anchor with the tie connected to its related tables; Table formed by the many to many relationship will be a tie, if it hasn't attribute except of primary key and/or foreign key. If it has other attributes then it will be an anchor. By the rules of anchor, the naming convention for anchor is the mnemonic addition.

B. Determine Attribute

Creating attribute requires all table attributes from data source. Anchor modeling attribute is divided into static attribute and historized attribute. Furthermore, historized attribute is divided into four parts: static, historized, knotted static, and knotted historized. The attributes need to accommodate so that any changes should be made into historized attribute, while attribute with a fixed value are grouped into static attribute. Based on analysis, the following rule is used to create historized attribute: Attributes that value is increasing or decreasing (example: weight and height of human); Attributes that its value has to change regularly. In anchor modeling, foreign and primary key won't be included as attributes. The intervention of user that understand the business process is needed to decide static and historized attributes.

C. Determine Knot

A Knot represent an attribute with constant value or attribute with a small range value. In relational table, Knot also can be a reference table. Reference table always store constant value (example is gender). But, it is also possible a reference table has some different values. In determining Knot from relational data, we propose to use threshold percentage of number different value of attribute. This threshold will be inputted by user. After threshold inputted by user, then: Calculation of knot percentage applied for all attributes in a table except primary and foreign key attribute; The Calculation of threshold percentage is by comparing possible value in one column with all number of records in this table. Example: gender column has 2 possible values (male and female) and number of records is 100, then the percentage is 2%. The probability of one attribute as a Knot even greater if the percentage is getting smaller.

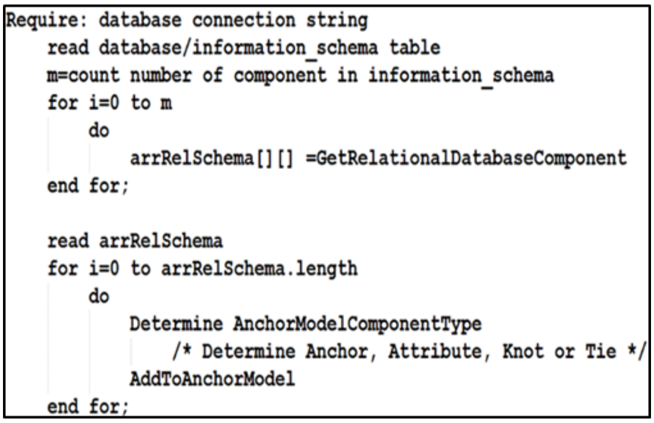
D. Determine Tie

There are several types of cardinality in a relational table. Those diversity effects on tie formation. Relationships with cardinality 1 : N or N : 1 form a tie that contains the primary key of the related entity. Table that appear because of cardinality of N : M and only have foreign key as attributes will form a tie. If table has other attributes then it will be an anchors that connected to the tie.

E. Determine Historized or Static Attribute and Historized or Static Tie

Attribute and tie grouped into static, historized, knotted static, and knotted historized. However, in this case only historized or static attributes and historized or static tie to be applied because: The value of attribute is growth/increase or decrease (eg., weight or height, the other example is the change in the status of quite a good example); Attributes that are considered necessary to change its value at regular intervals.

Historized attribute depends on the needs of the user itself (according to the business process), so in this case it takes the input from user. In application, all attributes (except primary key attributes and foreign keys) will be listed as a candidate of the historized attributes. Then the candidate is shown to the user, and user will select the historized attribute. The primary key is an unique value so it is not necessary to change the data, as well as the foreign key (primary key in another table) will not be the attribute in the anchor modeling. The overall pseudocode of algorithm can be seen in the Fig. 3.

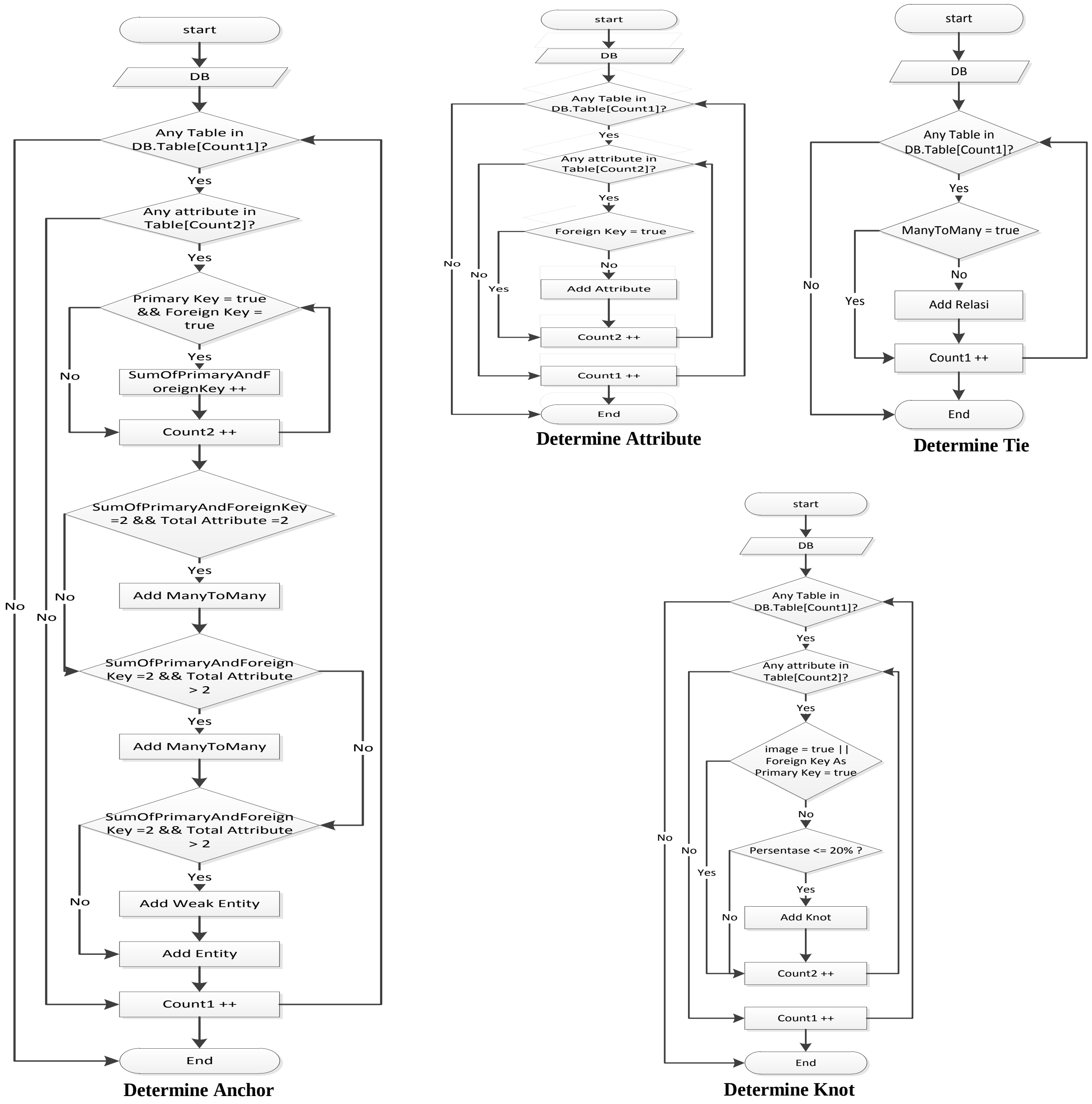


```
Require: database connection string
    read database/information_schema table
    m=count number of component in information_schema
    for i=0 to m
        do
            arrRelSchema[][] =GetRelationalDatabaseComponent
    end for;

    read arrRelSchema
    for i=0 to arrRelSchema.length
        do
            Determine AnchorModelComponentType
                /* Determine Anchor, Attribute, Knot or Tie */
            AddToAnchorModel
    end for;
```

Fig. 3. Algorithm for generating Anchor Model from Relational Database

The input of algorithm is a database connection string. Database connection string is a transactional database which need to be transformed in anchor modeling. All relational database components will be identified by reading the information schema table and stored in array variable. Then, all anchor modeling component type will be decided based on data in array variable. The detail algorithm to determine Anchor, Attribute, Knot and Tie can be seen in flowchart in Fig. 4.


start
DB
Any Table in DB.Table[Count1]?
Yes
Any attribute in Table[Count2]?
Yes
Primary Key = true && Foreign Key = true
Yes
SumOfPrimaryAndForeignKey ++
No
Count2 ++
SumOfPrimaryAndForeignKey =2 && Total Attribute =2
Yes
Add ManyToMany
No
No
No
SumOfPrimaryAndForeign Key =2 && Total Attribute > 2
Yes
Add ManyToMany
No
SumOfPrimaryAndForeign Key =2 && Total Attribute > 2
Yes
Add Weak Entity
No
Add Entity
Count1 ++
End
start
DB
Any Table in DB.Table[Count1]?
Yes
Any attribute in Table[Count2]?
Yes
Foreign Key = true
No
Add Attribute
No
No
Yes
Count2 ++
Count1 ++
End
start
DB
Any Table in DB.Table[Count1]?
Yes
ManyToMany = true
No
Add Relasi
No
Yes
Count1 ++
End
start
DB
Any Table in DB.Table[Count1]?
Yes
Any attribute in Table[Count2]?
Yes
image = true || Foreign Key As Primary Key = true
No
Persentase <= 20% ?
Yes
Add Knot
No
No
No
Yes
Count2 ++
Count1 ++
End

Determine Anchor

Determine Attribute

Determine Tie

Determine Knot

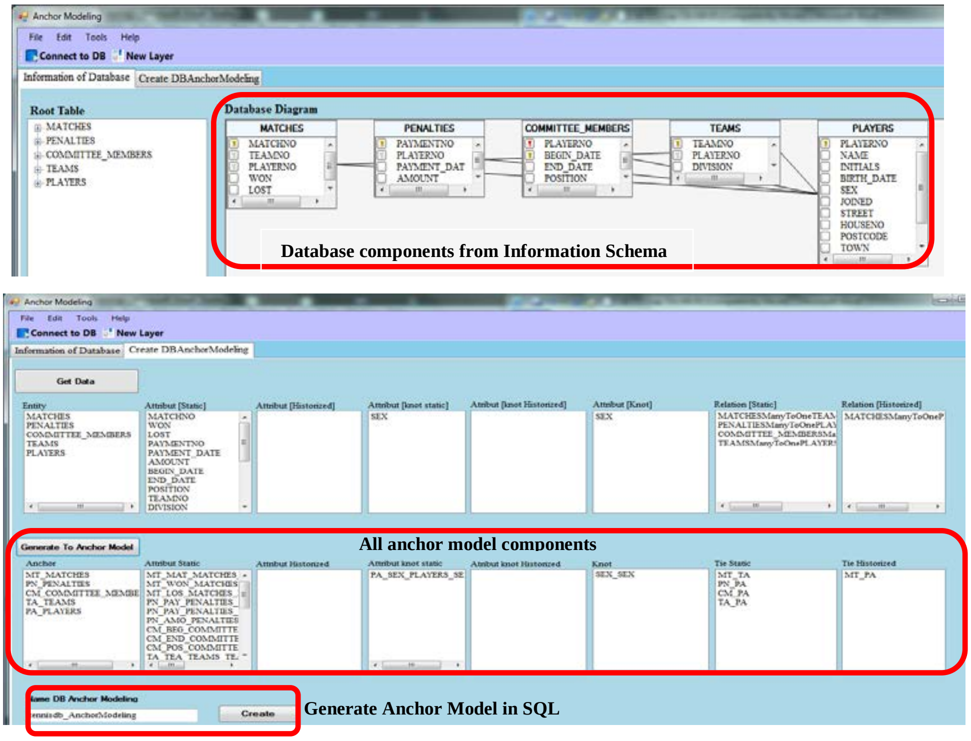
Fig. 4. Algorithm flowchart to determine anchor, attribute, knot, and tie

Fig. 4 shows each step to build anchor model data warehouse which are generate the anchor component and determine the attribute, knot, and tie, respectively. Each step read relational database component from transactional data source that already stored in array variable arrRelSchema. Finally, if all transactional data source components have been transformed, then anchor model will be displayed to the user.

IV. EMPIRICAL INVESTIGATION DESIGN

Since the focus of this research is to proposed algorithm and tool that generate anchor modeling component from relational database, we conducted a set of experiments using 3 relational databases. The summary of each database are employee database has 2 table with one to many relationship, Tennis DB has 5 tables and Northwind has 13 tables including one weak entity (Order Details) and 2 tables with many to many relationship (only have 2 foreign key attributes).

We designed a C#-based tool called AncRel v1.0 that takes a database in SQL Server as an input. Tool read information schema to get all relational database components. The tool implemented the proposed algorithm. The interface of designed AncRel v1.0 can be seen in the Fig. 5



Database components from Information Schema

All anchor model components

Generate Anchor Model in SQL

Fig. 5. The example interface of application AncRel v1.0

V. RESULT AND ANALYSIS

Based on experimental result, the application of the designed AncRel v1.0 algorithm to generate the anchor model from database relational was successfully achieved. Based on the three cases, each database component was converted to anchors, attributes, tie, and knots. Anchor was generated from entity, attributes from foreign key attributes apart, tie from a previous relationship and the entity from the database produced by Cardinality M : N. Knot component were resulted from the anchor attribute whose value is constant. The tool proposed a candidate knots component and then asked the user whether the candidate was declared as knots or not. Table I presents all anchor model component which generated by AncRel v1.0 and applied for each database. The number of Knot component can be different according to user input in the application.

TABLE I. COMPARISON RELATIONAL MODEL COMPONENT BETWEEN ANCHOR MODEL COMPONENT GENERATE BY AncRel v1.0

Database Name	Relational Model		Anchor Model			
	Tables	Attributes	Anchor	Attribute	Tie	#Knot
Employee	2	6	2	5	1	1
TennisDB	5	28	5	23	5	1
Northwind	13	87	11	76	8	1

Based on data in Table 1, there are some analysis that can be concluded:

1) More table appear in Anchor Model

AncRel v1.0 generated more tables for Anchor model than in the transactional relational database. The purpose is to historize all the data that stored in the table. Therefore, each entity, attribute and relationship stored as a table to support historization. The comparison of the table number in anchor model with relational database can be seen in Table 1. One of anchor modeling characteristics is Normalization 6NF. In normalizing process, the higher levels of normalization then more tables are created. In the Table 1, it can seen that the anchor model was generated by the application by applying the 6NF rule.

2) Usage Knot

It is possible to have same attributes in a different table in a database. For example: in the each database table (Northwind, Supplier, Employees and Customers tables) has city attribute, respectively. The designed application gave flexibility to a user to decide which attribute that should be as Knot. Then the application generated one Knot City and some attributes depends on number of the selected attribute city.

3) The validity of the algorithm

The anchor model which generated by AncRel v1.0 was verified with the anchor model that designed using free web application in www.anchormodeling.com. Anchor model from this tool was converted to SQL and created in the relational database. All of the components which resulted by AncRel v1.0 is similar with anchor model component from www.anchormodeling.com. But, it is also possible that the generated anchor model by AncRel v1.0 will be different with obtained from www.anchormodeling.com, especially for Historized or Static Attribute and Historized or Static Tie. These attributes and Ties are depend to the user input.

VI. CONCLUSION AND FUTURE WORK

In this paper, an algorithm and a tool of AncRel v1.0 has been proposed to generate anchor model from relational database. Based on three simple cases, it was obtained that the AncRel v1.0 algorithm could successfully generated an anchor model, however it is still need to verify the proposed method with different and complex cases.

Determining Knot and historized attribute was difficult. In the future work, we will try to apply the statistical method or Natural Language Processing to decide a Knot and Historized attribute automatically.

REFERENCES

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

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

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

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

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

[6] Knowles Curtis. 6NF Conceptual Models and Data Warehousing 2.0. Proceedings of the Southern Association for Information Systems Conference. Atlanta: 2012.

[7] Rundensteiner E., Koeller A., and Zhang X. Maintaining Data Warehouses over Changing Information Sources. Communications of the ACM, vol. 43, No. 6, 2000

[8] Sjøberg D. Quantifying Schema Evolution. Information Software Technology 35: 1993; p. 35-54.

[9] Turki I.Z., Jedidi F.G., Bouaziz R. Multiversion Data Warehouse Constraints, Proceedings of the ACM 13th international workshop on Data warehousing and OLAP, New York: 2010; p. 11-18

[10] Rönnbäck L, Regard O. Anchor modeling - Agile Information Modeling in Evolving Data Environments, Data & Knowledge Engineering journal, Sweden: 2010; p. 1229-1253.

[11] Rönnbäck L, Krumlinde V. Anchor Modeler tool, http://www.anchormodeling.com/modeler/latest/, MIT Open Source License, 2014.

[12] Rönnbäck L, Regardt O, Bergholtz M, Johannesson P, Wohed P. From Anchor Model to Relational Database, Sweden: 2010

Humasak Simanjuntak was born in Pematang Siantar, Indonesia 26 April 1983. He granted the bachelor degree of engineering informatics from Bandung Institute of Technology, Indonesia in 2007 and master degree of information system development from HAN University of Applied Sciences, The Netherlands in 2010.

He has been working as a lecturer in information system department at Del Institute of Technology, Indonesia since 2007 till present. He also assigned as the head of information system department since 2014, his interests are data/text mining, database system, data warehousing, information extraction, and information system development.