

Implementation of Data Backup and Synchronization Based on Identity Column Real-Time Data Warehouse

I Gede Adnyana^{a1}, I Made Dwi Jendra Sulastra^{b2}

^aComputer System Department, STMIK STIKOM Indonesia
Denpasar, Indonesia
¹adnyana.nakkuta@gmail.com

^bAccounting Department, Bali State Polytechnic
Denpasar, Indonesia
²dwijendrasulastra@gmail.com

Abstract

Failure in the process of loading data from the Online Transactional Processing(OLTP) system to the Normalized Data Store (NDS) database can occur. This caused by a disruption in the network so that the OLTP system is unable to save data to the OLTP and NDS databases. Backup and synchronization data scenarios are needed to maintain data consistency and data availability. In this research, the process of data backup and synchronization is done by providing an identity column for the table in the OLTP database. An identity column is used to give status data, value '0' if the inserting process fails, and value '1' if successful. Data backup is done by storing temporary data into a CSV file format, then the CSV file is read, and an insert process is carried out into the OLTP database. After the insertion process into the OLTP database is successful, it continues with the synchronization process between the OLTP database and the NDS. Data synchronization between OLTP and NDS databases is done by checking the value of the Identity Column in each table in the OLTP database.

Keywords: Normalized Data Store, Real-Time Data Warehouse, Backup, Synchronization, Identity Column

1. Introduction

Data processing using information systems provides benefits, including speed up processing time due to automation. Information systems can also process data accurately, thereby reducing the risk of human error. Problems arise when information systems cannot process data with very large volumes, and data is scattered in different systems with diverse database structures [1][2][3][4][5][6]. An organization will experience difficulty in decision making when it encounters conflicting reports due to the lack of consistency of data from the various data sources used. Therefore we need an integrated data processing model that can process heterogeneous transactional data called data warehouse (DWH) [1][2][3][4][5][6][7][8][9][10][11].

The current trend, an organization needs the latest information in decision making. The real-time data warehouse has different characteristics from the classic data warehouse [1][2][5][7][11]. To achieve real-time data warehouse is very dependent on the process known as ETL (Extract, Transform, Loading) [1][2][5][7][10][11][12][13][14][15][16][17]. There are several ETL approaches to realizing real-time data warehouse between processing data that only undergoes updating or known as the Change Data Capture (CDC) concept [1][2][5][10][11][16][18]. Another approach is to accelerate the frequency of data extraction [2][11][12][15][19]. Both approaches aim to reduce the data processing time lag so that real-time data warehouse is realized.

In recent years Real-Time Data Warehouse (RTDW) has become a trend worldwide. ETL Data Warehouse that used to be run once a day is now running every hour, even every 5 minutes

(mini-batch). This can be done by using two approaches, namely push approach and pull approach. By using a push approach, the source system pushes data into the Data Warehouse. The data warehouse will be updated as soon as the data in the source system changes. Changes to the source system will be detected using database triggers. The pull approach method uses time intervals to update data in the data warehouse. Changes to the source system are detected using the timestamp or identity column method. An identity column is a column (field) in a table whose value will be used as a benchmark when data will be pulled from the source system and then stored in a data warehouse [2].

Some research has been carried out of RTDW, such as research on RTDW modeling using the CDC [5][10][11][16][18]. There is also research on making RTDW using the CDC Event-Driven Programming method [11]. Then there is research on the architecture of RTDW and how the process of making RTDW from the Traditional model [7]. The process of making RTDW is very much determined by the ETL process. To optimize the process of making RTDW, several studies have been conducted on the ETL process [11][12][13][14][15][16][17][18]. From these, several studies all focus on making RTDW without discussing the possibility of failure in the process of transferring data from the source system to the RTDW database.

To realize RTDW, data from the source system is loaded into a staging area or Normalized Data Store (NDS) which is used as a temporary storage place for data from various sources and ETL process sites before it is loaded in the Dimensional Data Store [1][2][5][11][12][13][14][15][16][17]. In one study on RTDW that implemented the use of CDC based on Event-Driven Programming [11], data was parallel stored in the OLTP database and the NDS database. In the process of loading data from the source system into the NDS, it is possible for the failure to occur due to the failure of the OLTP system to save data to the OLTP database and the NDS database. Failures can be caused by disruptions on the network, causing connection failures between the OLTP system and the OLTP database and the NDS database. To maintain data consistency between the source database and the NDS database and data availability, backup and synchronization data scenarios are needed.

In this research, the data backup process is carried out by storing temporary data in the form of a file with the format of Comma Separated Values (CSV), which is then followed by the process of synchronizing data. The data synchronization process is carried out by giving identity columns to each table in the source database. Furthermore, checking is done by checking the value of the identity column.

This research is expected to be able to handle the failure of the data transfer process from the OLTP system to the Real-Time Data Warehouse.

2. Research Methods

The stages in this research are generally shown in Figure 1. In Figure 1, the research stage starts with defining the problem where the problem that arises is the possibility of failure data transfer from the source system to the formation of RTDW. Then the next stage is the study of literature to obtain supporting literature for solving the problem. The backup scenario design is done by designing a backup scenario model of the problem if there is a failure of data transfer from the source system to the data warehouse. The next step is to design a data synchronization scenario between the source database and the NDS database using the identity column method. Data backup and synchronization models are then developed and implemented. Then do the system functionality test and analysis of test results. Then from the analysis of the test results, conclusions are made.

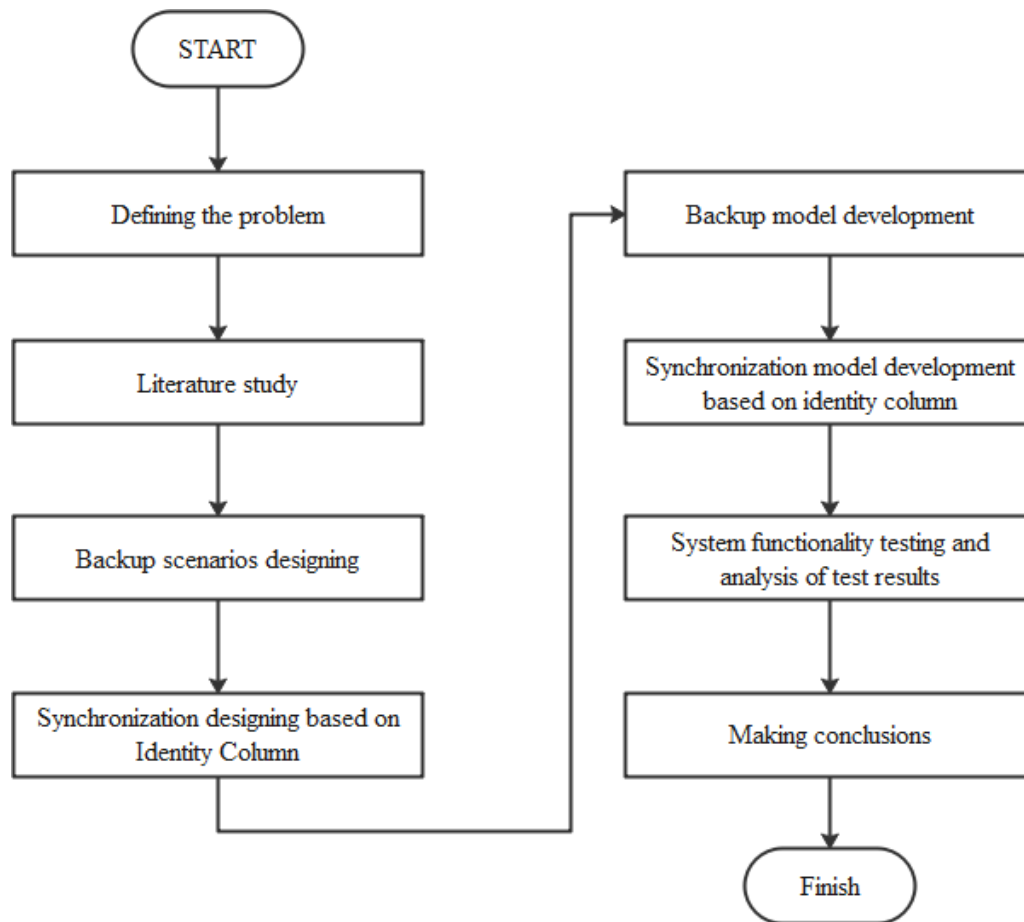


Figure 1. General Stages of Research

2.1. Real-Time Data Warehouse

The data warehouse is a system that retrieves and consolidates data periodically from source systems into dimensional or normalized data stores [2]. The data warehouse is a collection of data that has a subject-oriented, integrated, time-variant, and non-volatile nature of data collection in support of the management decision-making process [2]. The data warehouse is a system that extracts, cleans, adapts, and sends data sources into a dimensional data storage and then supports the implementation of queries and analysis for decision-making purposes. Information in the data warehouse is always presented in the form of dimensions and facts [1][2].

Classic data warehouse usually updates the data every day or every week. In accordance with business requests that require up to date or real-time data processing, the concept of the real-time data warehouse was created. In a real-time data warehouse, the process of updating data is carried out dynamically continuously with a break time that is almost close to zero [2]. To create a real-time data warehouse highly dependent on the Extract, Transform, Load (ETL) process. Another approach to reducing the lag time is to only process data that has been updated or known as the Change Data Capture concept [1][2].

2.2. Change Data Capture (CDC)

Change Data Capture (CDC) is an innovative approach to data integration, based on identifying, capturing, and sending changes made by data sources. By processing only the changes, the CDC makes the data integration process more efficient and reduces costs by reducing latency [1][2].

CDC is designed to maximize the efficiency of the ETL process. Without CDC, all data in the Online Transaction Processing (OLTP) database will be moved to the data warehouse whenever needed, while with CDC, only data changes that occur in the OLTP database will be moved. Therefore, the CDC can minimize the restore that is used to move data changes and minimize the latency of sending information, so this will save costs. There are two CDC scenario models integrated with ETL tools: [2].

- a. Batch-Oriented CDC (Pull CDC) Scenario: It is processing a set of data that only experiences periodic changes in high or low frequency.
- b. Live CDC Scenarios (Push CDC): Are sending data changes to the ETL tool after the changes occur. It can be done with an event-delivery mechanism or messaging middleware.

2.2.1. CDC Based on Event-Driven Programming.

Event-driven programming is a programming technique where all program execution flows are determined by an event. When the program starts, it will wait for user input events. For each event that appears, the program will run the syntax to respond. The flow of program execution is determined by the order in which events occur [11].

In CDC based event-driven programming, when a user runs an event by clicking the button on the GUI, the data that has been filled in the GUI will be stored in the OLTP database, and the data will also be sent to the Normalized Data Store (NDS) database for further process. Data that has been inputted in a parallel GUI will be stored in two databases, namely the OLTP database and the NDS database [11].

2.3. Extract, Transform, Loading (ETL)

Extract, Transform, Loading (ETL) is a very important process in the data warehouse, with this ETL data from the operating system can be entered into the data warehouse. The purpose of ETL is to collect, filter, process, and combine data from various sources to be stored in a data warehouse [1][2].

2.3.1. Extract

Most of the data in the source system are very complex, so determining the relevant data is very difficult. Efforts to design and create extraction processes are very consuming time [1][2]. Raw data originating from the source system can usually be directly stored in the staging area with minimal restructuring to maintain the authenticity of the data. There are three methods for extraction that are commonly used, namely [2]:

- a. Whole table every time, this method extracts all rows in the table (full extraction). This method is suitable if the table size is small and consists of only a few rows.
- b. Incremental extract, this method extracts only the changed rows, not extracts the entire table. In getting a changed row, you can use the timestamp column, identity column, transaction date, triggers, or combinations.
- c. Fixed range, this method extracts several records or extracts with a certain time period; for example, the last six months of data.

2.3.2. Transform

The transformation phase applies a set of rules or functions to data taken from the source to get data to be sent to the final target. Some data sources will require very little or no data manipulation [1][2].

2.3.3. Loading

The process of loading or also known as the process of delivering, is a process in which the transformed data is ready to be entered into a data warehouse, where the design of the table structure of data to be loaded (load) is made in the dimensional design process. The data from the loading process is ready to be queried and presented by the data warehouse. Therefore the

dimensional data warehouse design will determine the speed of the query process performed [1][2].

2.4. Data Backup and Synchronization Based on Identity Column

Data backup is needed to maintain the availability of data if there are problems with the system; for example, there are database connection problems in the system. In this research, data backup is done by storing temporary data in the form of files in the format of Comma Separated Values (CSV). If the connection problem has been resolved, then the CSV file reading process is then performed then the insert process is carried out in the OLTP database. If the insert process in the OLTP database is successfully continued with the process of synchronizing the data between the OLTP database and the NDS database

Data synchronization is needed to ensure data consistency is maintained. Data flow in the data warehouse, data from the source system is loaded into a staging area storage or Normalized Data Store (NDS). It is possible that the data failed to be loaded into the NDS, therefore synchronization of the data between the OLTP database as the source and the NDS database as the destination. The technique used to check records that fail to load on an NDS basis is the Identity Column, which is a column with a certain status value such as '0' or '1'. If the record fails to load in the NDS database, then the Identity Column record will have a value of '0'. Then the record with the Identity Column value of '0' is carried out by inserting into the NDS database. If the insert process is successful, then the process of updating the Identity Column value will be made to a value of '1'.

3. Results and Discussion

The test is carried out using a Customer Relationship Management System (CRM) Online Transaction Processing (OLTP) simulation application that is useful for recording any customer complaints telecommunications services.

3.1. CDC Event Programming

When users enter new data in the CRM system, the data will be sent in parallel to the OLTP database and NDS database. CDC event programming is triggered by events that occur in the CRM system. The data input process on the CRM system can be seen in Figure 2.

The screenshot shows a 'Create Case' form with the following fields:

- Case: CS180919094423
- Date of Complain: Wednesday, September 19, 2018
- Caused By: [Empty]
- MSISDN A#: 62817600777
- Status: Dispatch
- Date of Dispatch: Wednesday, September 19, 2018
- Complain: Tidak bisa SMS
- Dispatch to: Fault VAS
- Action: [Empty]
- Area: Bali
- Date of Closed: Monday, April 27, 2020

Customer's Data:

- No. Identitas: 5103010022194513
- Alamat: Badung
- Tgl. Lahir: 9/10/1999
- Nama: I Kadek Ariawan
- Tempat Lahir: Denpasar
- Jenis Kelamin: Laki-laki

Simcard Information:

- SMSO: 001
- MO: 001
- UMB: 001
- Balance: 100000
- Registration Date: 8/15/2018
- SMST: 001
- MT: 001
- Profile: Prepaid
- HLR: HBDG2
- Expired Date: 11/15/2018

Offline Action: Save, Update

NoCase	MSISDN	JnsKomplain	TglKomplain	TglSolved	CauseBy	IDPegawai	Status	DispatchKe	TglDispatch
CS180914062733	62817600710	Tidak bisa sms	9/14/2018 6:27 ...	9/14/2018 7:15 ...	Pulsa tdk ckup	EMP900001	Closed	Fault VAS	9/14/2018 6:27
CS180914063828	62817600711	Tidak bisa UMB	9/14/2018 6:38 ...	10/16/2018 11:5...	Internal error	EMP900001	Closed	Fault VAS	9/14/2018 6:38
CS180914064521	62817600721	Tidak bisa call	9/14/2018 6:45 ...	10/16/2018 11:5...	Network error	EMP900001	Closed	Fault VAS	9/14/2018 6:45

Figure 2. CDC Event Programming on CRM Systems

Figure 3 shows the data has been successfully stored in the OLTP database in the TComplain table.

NoCase	MSISDN	JnsKomplain	TglKomplain	TglSolved	CauseBy	IDPegawai	Status	DispatchKe
CS180919094423	62817600777	Tidak bisa SMS	9/19/2018 9:44 ...			EMP900001	Dispatch	Fault VAS

Figure 3. Results of CDC Event Programming in OLTP Database

Figure 4 shows the data has been successfully stored in the NDS database in the TNDSComplain

Kuncikomplain	nocase	MSISDN	JnsKomplain	TglKomplain	DispatchKe	TglDispatch	Active
11	CS180919094423	62817600777	Tidak bisa SMS	2018-09-19 00:00:00.000	Fault VAS	2018-09-19 00:00:00.000	1

Figure 4. Results of CDC Event Programming in the NDS Database

3.2. Data Backup

Data backups are needed to ensure that every transaction remains recorded when a database connection problem occurs. Transactions will be saved in a file format with Comma Separated Values (CSV) format and will be processed into the OLTP database when the database connection is back to normal. The process of inputting data on an OLTP system while offline can be seen in Figure 5.

The screenshot shows a 'Create Case' form with the following fields and values:

- Case: CS181111020607
- MSISDN A#: 62817600963
- MSISDN B#: 628193231987
- Complain: Cannot SMS
- Area: Bali
- Date of Complain: Sunday, November 11, 2018
- Status: Dispatch
- Date of Dispatch: Sunday, November 11, 2018
- Dispatch to: Fault VAS
- Date of Closed: Sunday, November 11, 2018
- Customer's Data: No. Identitas, Nama, Alamat, Tempat Lahir, Tgl. Lahir, Jenis Kelamin
- Simcard Information: SMSO, SMST, MO, MT, UMB, Profile, Balance, HLR, Registration Date, Expired Date

Buttons: Save, Cancel, Close, Offline Action (Save, Update)

Figure 5. Create Case Form Input

If there is a database connection problem, the save and update buttons will be enabled on the offline action feature. In this research, the connection failure is simulated with a scenario of giving a false (closed) value to the connection variable in the OLTP system so that the connection with the OLTP database and the NDS database fails. Since the connection cannot be made, all transaction data will be saved in the CSV file shown in Figure 6.

```

FileCreateCase.csv
1 CS181111020340,62819800710,Cannot call,11/11/2018 12:00:00 AM,,EMP900001,Dispatch,Fault MSC,11/11/2018 12:00
2 CS181111020445,62819800715,Cannot access UMB,11/11/2018 12:00:00 AM,,EMP900001,Dispatch,Fault VAS,11/11/2018
3 CS181111020607,62817600963,Cannot SMS,11/11/2018 12:00:00 AM,,EMP900001,Dispatch,Fault VAS,11/11/2018 12:00:
4 CS181111020916,62817600965,Cannot MBanking,11/11/2018 12:00:00 AM,,EMP900001,Dispatch,Fault VAS,11/11/2018 1
5 CS181111021246,62818100270,Cannot SMS,11/11/2018 12:00:00 AM,,EMP900001,Dispatch,Fault VAS,11/11/2018 12:00:
6 CS181111021331,62818100325,Cannot acces data,11/11/2018 12:00:00 AM,,EMP900001,Dispatch,Fault GPRS/IN,11/11/
    
```

Figure 6. Transaction Data in CSV Format

When the database connection has returned to normal, the data file in CSV format will first be read into a tabular form, and then the process is inserted into the TComplain table in the OLTP database. The process of reading a CSV file and inserting data into an OLTP database is shown in Figure 7.

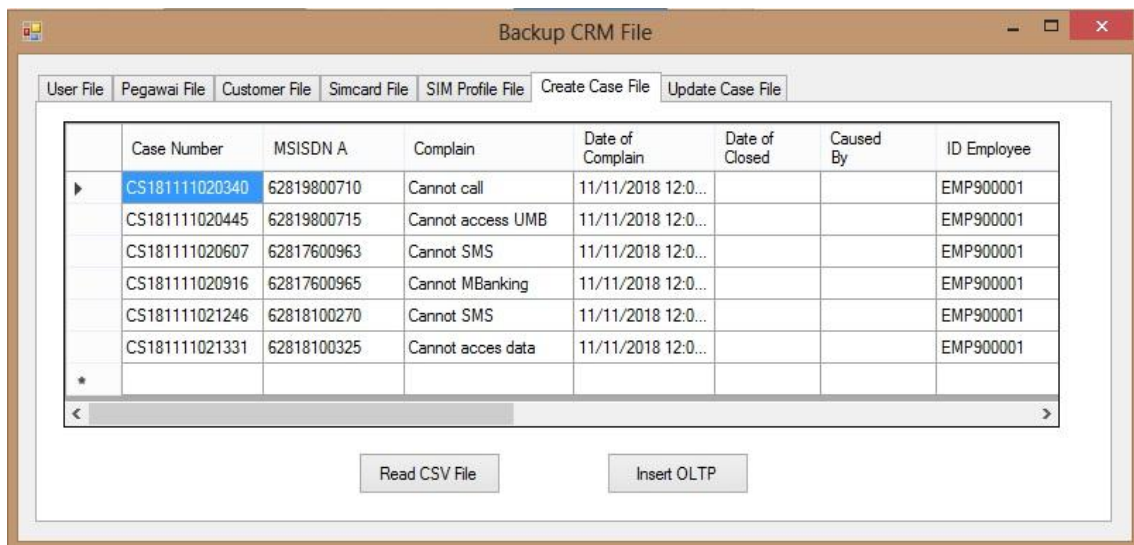


Figure 7. CSV File Format Reading

If the OLTP insert key is pressed, the data stored in the CSV file will be inserted into the TComplain table in the OLTP CRM database. The results of adding data to the TComplain table in the OLTP CRM database are shown in Figure 8.

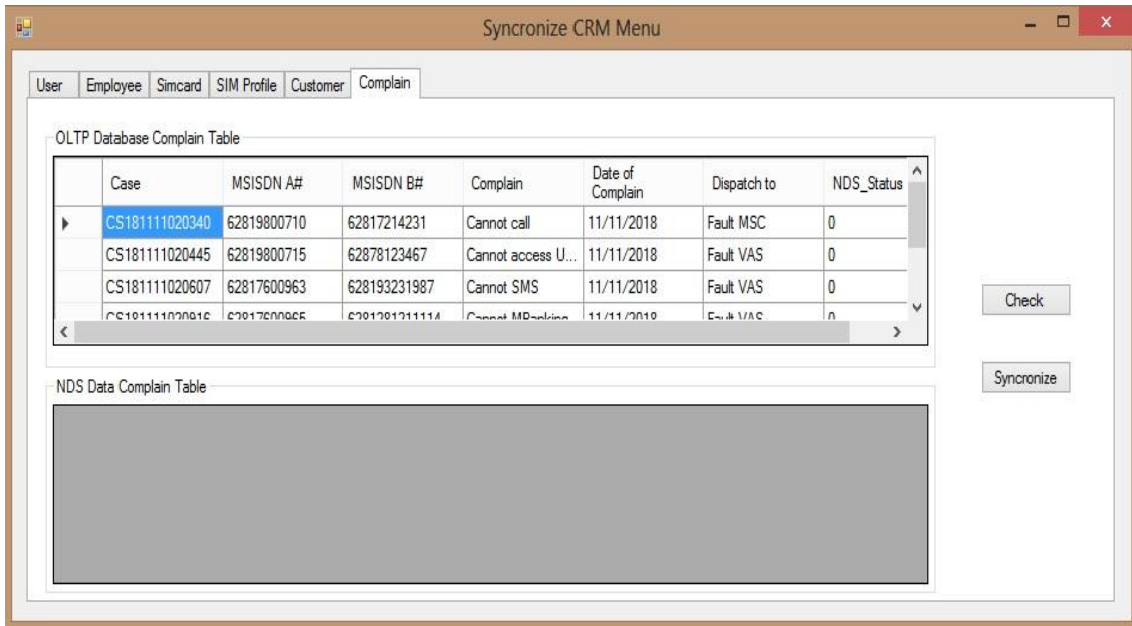


Figure 8. CRM Synchronization Menu

Adding data from files in CSV format is done in the OLTP CRM database, which then requires the process of synchronizing data between the OLTP database and the NDS database shown in Figure 9.

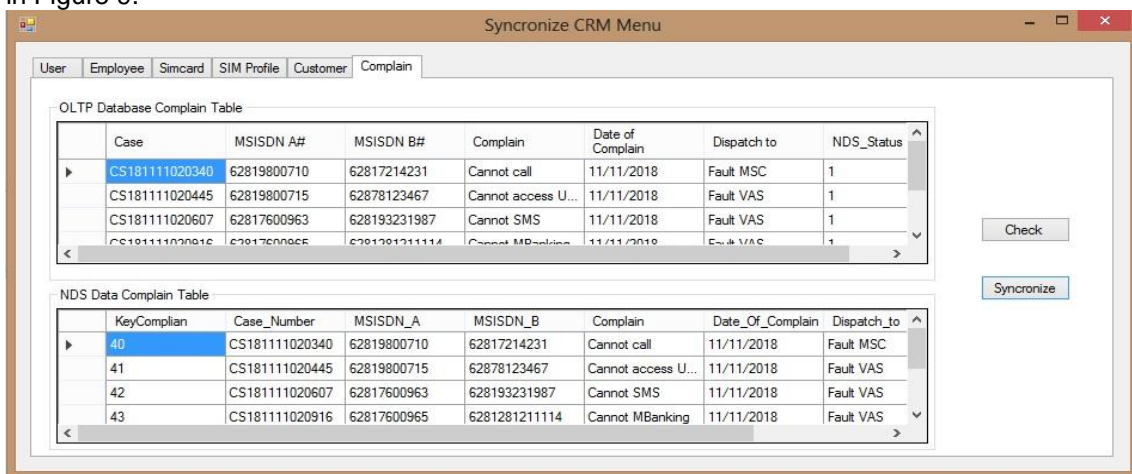


Figure 9. Synchronization between TComplain Table and TNDSComplain Table

Then every 30 seconds interval, the scheduler runs the data extraction process in the NDS database. Data in the NDS database is sent to the database warehouse using the Incremental Extraction based Timestamp method. This method can process only the latest data contained in the NDS database. The results of the process of inserting into a data warehouse database with the Incremental Extraction method based on Timestamp are shown in Figure 10.

ComplainKey	Case_Number	MSISDN_A	MSISDN_B	Complain	Date_of_Complain	Dispatch_to	Effective_date
1	40	CS181111020340	62819800710	62817214231	Cannot call	2018-11-11 00:00:00.000	2018-11-11 03:01:19.570
2	41	CS181111020445	62819800715	62878123467	Cannot access UMB	2018-11-11 00:00:00.000	2018-11-11 03:01:19.570
3	42	CS181111020607	62817600963	628193231987	Cannot SMS	2018-11-11 00:00:00.000	2018-11-11 03:01:19.570
4	43	CS181111020916	62817600965	6281281211114	Cannot MBanking	2018-11-11 00:00:00.000	2018-11-11 03:01:19.570
5	44	CS181111021246	62818100270	628189988165	Cannot SMS	2018-11-11 00:00:00.000	2018-11-11 03:01:19.570
6	45	CS181111021331	62818100325	628561113562	Cannot acces data	2018-11-11 00:00:00.000	2018-11-11 03:01:19.570

Figure 10. Results in the Data Warehouse

3.3. Data Synchronization

In the CDC based event-driven programming method, data inputted on the parallel CRM system GUI is stored in the OLTP database and the NDS database. Data synchronization is needed to ensure the data in the OLTP database is the same as the NDS database. Checking and synchronizing data in the OLTP database can be seen in Figure 11.

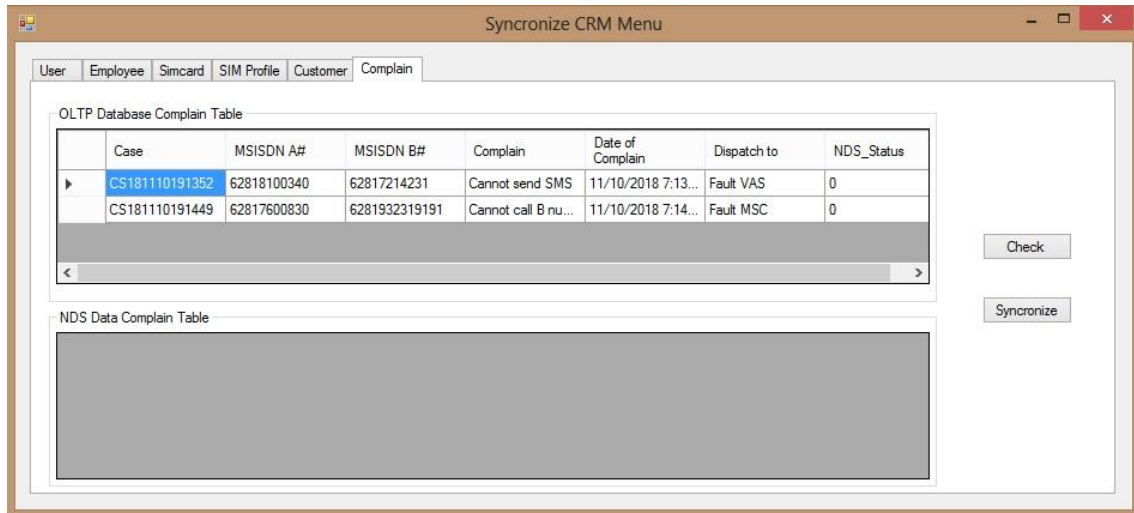


Figure 11. CRM Synchronization Menu

Figure 11 shows the results of checking synchronization in the OLTP CRM database. Two data are not synchronized. Both of these data failed to be stored in the NDS database, and the data was stored in the OLTP CRM database. Asynchronous data is found by checking the status of the Identity Column in the TComplain table, which functions as an Identity Column, the NDS_Status column, which, if not synchronized, has a value of '0'.

If data that has not been synchronized found, then the data synchronization process is carried out by pressing the Synchronize button. When the Synchronize button is pressed, insert the asynchronous data into the TNDSComplain table in the NDS database. If the data is successfully inserted into the NDS database, the NDS_status column value is updated to '1'. The process of synchronizing data between the TComplain table and the TNDSComplain table is shown in Figure 12.

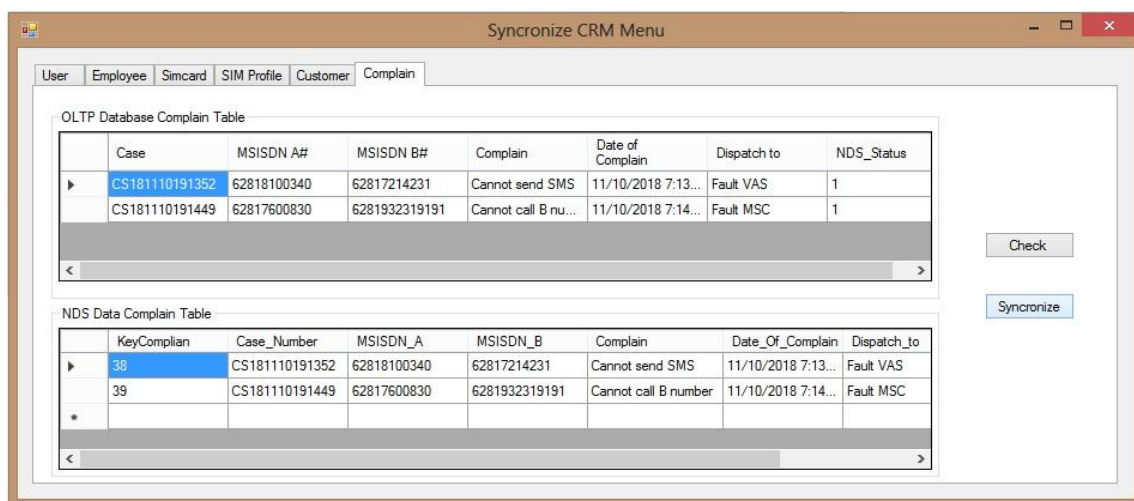


Figure 12. Synchronizing data between TComplain table and TNDSComplain table

Then every 30 seconds interval, the scheduler runs the data extraction process in the NDS database. Data in the NDS database is sent to the database warehouse using the Incremental Extraction based Timestamp method. This method can process only the latest data contained in the NDS database. The results of the process of inserting into a data warehouse database with the Incremental Extraction method based on Timestamp are shown in Figure 13.

ComplainKey	Case_Number	MSISDN_A	MSISDN_B	Complain	Date_of_Complain	Dispatch_to	Effective_date
38	CS181110191352	62818100340	62817214231	Cannot send SMS	2018-11-10 19:13:52.000	Fault VAS	2018-11-11 01:32:17.547
39	CS181110191449	62817600830	6281932319191	Cannot call B number	2018-11-10 19:14:49.000	Fault MSC	2018-11-11 01:32:17.547

Figure 13. Results in the TFact_Complain table

4. Conclusions

Data backup is done by storing temporary data in a file in the format of Comma Separated Values (CSV), then the CSV file is read, and the process is inserted into the OLTP database. After the insertion process into the OLTP database is successful, it continues with the synchronization process between the OLTP database and the NDS database. Data synchronization between the OLTP database and the NDS database is done by checking the value of the Identity Column in each table in the OLTP database. If the value of the Identity Column is '0', then the process of inserting the data into the NDS database is carried out. If the data is successful, it is loaded into the NDS database, then the value of the Identity Column is updated to be '1'. Data in the NDS database is loaded into the Dimensional Data Store with the Incremental Extraction based Timestamp method to create the Real-Time Data Warehouse.

References

- [1] R. Kimball and M. Ross, *The Data Warehouse Toolkit, The Definitive Guide to Dimensional Modeling*. 2013.
- [2] V. Rainardi, *Building a data warehouse: With examples in SQL server*. 2008.
- [3] A. Khalaf Hamoud, A. Salah Hashim, and W. Akeel Awadh, "CLINICAL DATA WAREHOUSE A REVIEW," *Iraqi Journal Computing Informatics*, 2018, doi: 10.25195/2017/4424.
- [4] M. P. Ambara, M. Sudarma, and I. N. S. Kumara, "Desain Sistem Semantic Data Warehouse dengan Metode Ontology dan Rule Based untuk Mengolah Data Akademik Universitas XYZ di Bali," *Majalah Ilmiah Teknologi Elektro* 2016, doi: 10.24843/mite.2016.v15i01p02.
- [5] I. M. D. J. Sulastra, M. Sudarma, and I. N. S. Kumara, "PEMODELAN INTEGRASI NEARLY REAL TIME DATA WAREHOUSE DENGAN SERVICE ORIENTED ARCHITECTURE UNTUK MENUNJANG SISTEM INFORMASI RETAIL," *Majalah Ilmiah Teknologi Elektro* 2015, doi: 10.24843/mite.2015.v14i02p03.
- [6] M. R. Pasha, "Data Warehousing and the Unstructured Data," *Bahria Univercity Islam. Campus Gradute Resuslt*, vol. DOI:10.1, 2016.
- [7] S. Bouaziz, A. Nabli, and F. Gargouri, "From traditional data warehouse to real time data warehouse," in *Advances in Intelligent Systems and Computing*, 2017, doi: 10.1007/978-3-319-53480-0_46.
- [8] S. Bouaziz, A. Nabli, and F. Gargouri, "Design a data warehouse schema from document-oriented database," in *Procedia Computer Science*, 2019, doi: 10.1016/j.procs.2019.09.177.
- [9] F. Z. Al Faris, Suharjito, Diana, and A. Nugroho, "Development of Data Warehouse to Improve Services in IT Services Company," in *Proceedings of 2018 International Conference on Information Management and Technology, ICIMTech 2018*, 2018, doi: 10.1109/ICIMTech.2018.8528146.
- [10] H. Chandra, "Analysis of Change Data Capture Method in Heterogeneous Data Sources to Support RTDW," in *2018 4th International Conference on Computer and Information*

- Sciences: Revolutionising Digital Landscape for Sustainable Smart Society, ICCOINS 2018 - Proceedings*, 2018, doi: 10.1109/ICCOINS.2018.8510574.
- [11] I. G. Adnyana, M. Sudarma, and W. G. Ariastina, "Middleware ETL with CDC based on Event Driven Programming," *International Journal Of Engineering And Emerging Technology*, vol. Vol. 3, No, 2018.
- [12] A. Wibowo, "Problems and available solutions on the stage of Extract, Transform, and Loading in near real-time data warehousing (a literature study)," in *2015 International Seminar on Intelligent Technology and Its Applications, ISITIA 2015 - Proceeding*, 2015, doi: 10.1109/ISITIA.2015.7220004.
- [13] A. Sabtu *et al.*, "The challenges of Extract, Transform and Loading (ETL) system implementation for near real-time environment," in *International Conference on Research and Innovation in Information Systems, ICRIS*, 2017, doi: 10.1109/ICRIIS.2017.8002467.
- [14] R. P. Deb Nath, K. Hose, T. B. Pedersen, and O. Romero, "SETL: A programmable semantic extract-transform-load framework for semantic data warehouses," *Information Systems*, 2017, doi: 10.1016/j.is.2017.01.005.
- [15] N. Biswas, A. Sarkar, and K. C. Mondal, "Efficient incremental loading in ETL processing for real-time data integration," *Innovation in System Software Engineering*, 2019, doi: 10.1007/s11334-019-00344-4.
- [16] S. Thulasiram and N. Ramaiah, "Real Time Data Warehouse Updates Through Extraction-Transformation-Loading Process Using Change Data Capture Method," 2020.
- [17] B. Pan, G. Zhang, and X. Qin, "Design and realization of an ETL method in business intelligence project," in *2018 3rd IEEE International Conference on Cloud Computing and Big Data Analysis, ICCCBDA 2018*, 2018, doi: 10.1109/ICCCBDA.2018.8386526.
- [18] Denny, I. P. M. Atmaja, A. Saptawijaya, and S. Aminah, "Implementation of change data capture in ETL process for data warehouse using HDFS and apache spark," in *Proceedings - WBIS 2017: 2017 International Workshop on Big Data and Information Security*, 2018, doi: 10.1109/IWBIS.2017.8275102.
- [19] I. Mekterović and L. Brkić, "Delta view generation for incremental loading of large dimensions in a data warehouse," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings*, 2015, doi: 10.1109/MIPRO.2015.7160496.