International Journal of Cyber and IT Service Management (IJCITSM) p-ISSN: 2797-1325
Vol. 1 No. 2 October 2021 e-ISSN: 2808-554X

Implementation Of Database Auditing By Synchronization DBMS

Muhammad Rehan Anwar[1], Ria Panjaitan[2], Ruli Supriati[3]
National University of Sciences and Technology[1], University of Esa Unggul[2], University of Raharja[3].
Scholars Ave, H-12, Islamabad, Islamabad Capital Territory[1], Jl. Arjuna Utara No.9, Kb. Jeruk, Kota Jakarta Barat[2], Jenderal Sudirman No.40, Cikokol, Kota Tangerang[3].
Pakistan[1], Indonesia[2,3]
e-mail: rehan749@gmail.com[1], ria.panjaitan@esaunggul.ac.id[2] , ruli@raharja.info[3]

Author Notification
October 2021
Final Revised
October 2021
Published
October 2021

To cite this document:
Anwar, M., Panjaitan, R., & Supriati, R. (2021). Implementation Of E-Business Information System In Indonesia : Prospects And Challenges. *International Journal of Cyber and IT Service Management (IJCITSM)*, *1*(2), 197-205. Retrieved from https://iiast-journal.org/ijcitsm/index.php/IJCITSM/article/view/53

DOI:
https://doi.org/10.34306/ijcitsm.v1i1.53

Abstract

Database auditing is a crucial part of database security and governance requirements. Database administrators must be better aware of the procedures used to preserve business data, as well as the standards and regulations that may be applied to the data. Database auditing is a capability for searching the use of database authority and resources at a high level. Every database action that is audited creates an audit trail of the information changes performed when auditing is enabled. Database synchronisation is a type of replication in which every copy of a database is convinced to have the same data. Database synchronisation allows us to do a variety of tasks, one of which is database auditing, which logs every database activity. The audit trails created by data operations allow DBAs (Database Administrators) to do periodic examination of access patterns and data updates in the database management system (Database Management System).

***Keywords** : Database Auditing, Audit Trail, Database Security, Data Synchronization.*

1. Introduction

It is critical for businesses to have the ability to execute analysis and report production from information systems in an effective, efficient, and integrated manner [1]. Many businesses want the analytical process to take as little time as feasible [2]. Since the advent of the digital revolution, information systems have been defined and put into practise. Information security arose in terms of authentication and authorisation when information was used for business processes. Although the notion protects information, it is of no use in investigations. Log files are presented as a way to track database and system access. The log file's primary goal, however, is to restore(recover) transactions. Database auditing is an option for investigating transactions that occur [3]. It is critical to conduct a database audit in order to detect malicious activities, maintain data quality, and optimise system performance [4].

One of the most serious issues in information security is database auditing. Business apps lose track of the company's business operations due to a lack of data audits. Historical or temporal database data is required to develop audits in order to track operations and types of activities through time [3]. Database auditing is a vital part of database security and government regulatory compliance. Database administrators must be more knowledgeable about the methods used to secure corporate data, as well as monitor and guarantee that sufficient data security is in place [5].

Synchronization Database is a component of replication, which is the act of guaranteeing that each copy of the database's data includes the same objects and data. A database's synchronisation function permits data to be updated in real time or on a regular basis if the data changes [6]. This condition may be used to provide audits that records every database activity that is relevant.

Several research on database auditing have been conducted in the past. Some of them include theories that assist the auditing process. The research Auditing Database Design on Historical Data [3] offered many techniques for doing historical data auditing in databases, including audit, line-based audit-based column, audit logs, and semi-structured table auditing. The Three Methodologies Auditing may be done using relational databases initially, and semi-structured auditing with relational database extensions and XML technology (extensions markup language) in databases like IBM DB2 9.5, Oracle 10g, and MS SQL Server. Manytrail audits (audit trails) are created for the environment, database, therefore there are various types in auditing, according to a study titled Teaching Database Security and Auditing [7]. The antrail audit of logon and logoff is the initial category of audit required in most environments auditing, and it captures all unsuccessful login attempts. The database's DCL (Data Control Language) is audited in the second category. Changes to user rights, user logins, and other security characteristics are all part of the DCL. DDL (Data Definition Language) auditing, such as altering database schema or tables, is the third type. DDL instructions are often used in some types of data theft. The auditing of data modifications using DML (Data Manipulation Language) operations is the fourth category. Changes in DML instructions, including old and new values, can be documented via auditing. Auditing modifications to the source of stored procedures and triggers harmful, where code may be readily disguised, is the fifth type. The auditing of database mistakes caused by different factors, such as database assaults by other parties, is the sixth category.

This research uses the relational model to use row-based auditing to audit DML activity on business transactions, taking into account operation status, valid time, and kind of operation.

2. Research Methodology

Synchronization is used to implement database auditing. The Waterfall system development approach is used by DBMS as part of the SDLC (Software Development Life Cycle). The Waterfall approach of system development organises the development step. The following are the steps of research study for Database Auditing Implementation Using DBMS Synchronization:

1. Defining the problem of the designed system.
2. Collecting data related to system design.

3. Collection and mastery of supporting theories for system design.
4. Design Database using theplatform MySQL.
5. Design engine Synchronization Using the Python programming language.
6. System testing to obtain appropriate results.
7. Conclusion.

2.1. Synchronization Overview

The synchronization design implemented in the data exchange process in the MySQL DBMS is shown in Figure 1.

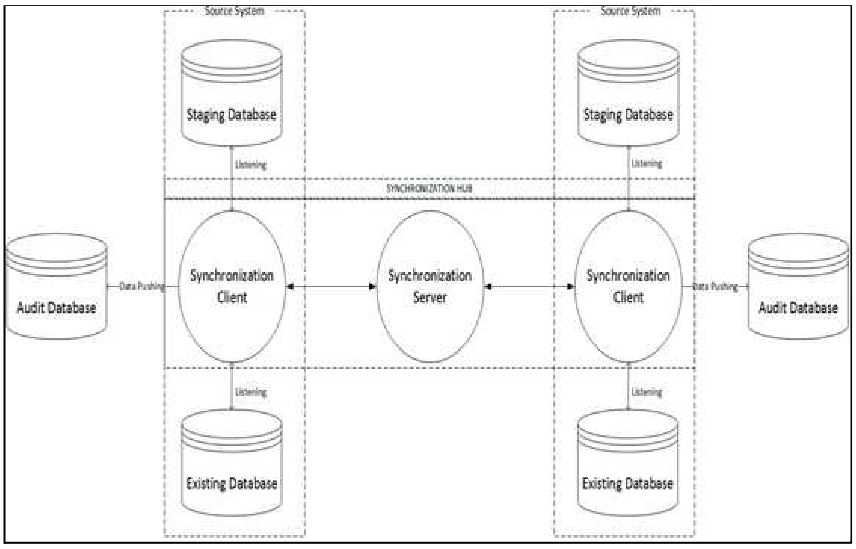


Figure 1. Sync Architecture Schematic

The developed synchronisation engine's architectural schema is real-time. Every time new data or data modifications are added to an existing database, the changes are transmitted to the destination and recorded as a new row in the table auditing that corresponds to the transactions.

The databases that are synced on the source system between the databases that exist and staged at each site must be similar prior to the synchronisation procedure. This is done to maintain the consistency of the synced data. The synchronisation procedure can begin once these requirements are satisfied.

Because the staging database is linked to the current database, synchronisation between the two databases speeds up the process of auditing. The procedure entails auditing on the source system such that the auditing database only receives the outcomes of data changes.

2.2. Process *Auditing*

The process in this study is auditing, which is created by the engine synchronisation, which runs constantly on the source system. The engine starts when a manipulation event happens in an existing database, such as when a user adds new data, edits or deletes data, or modifies or deletes some fields in an existing database.

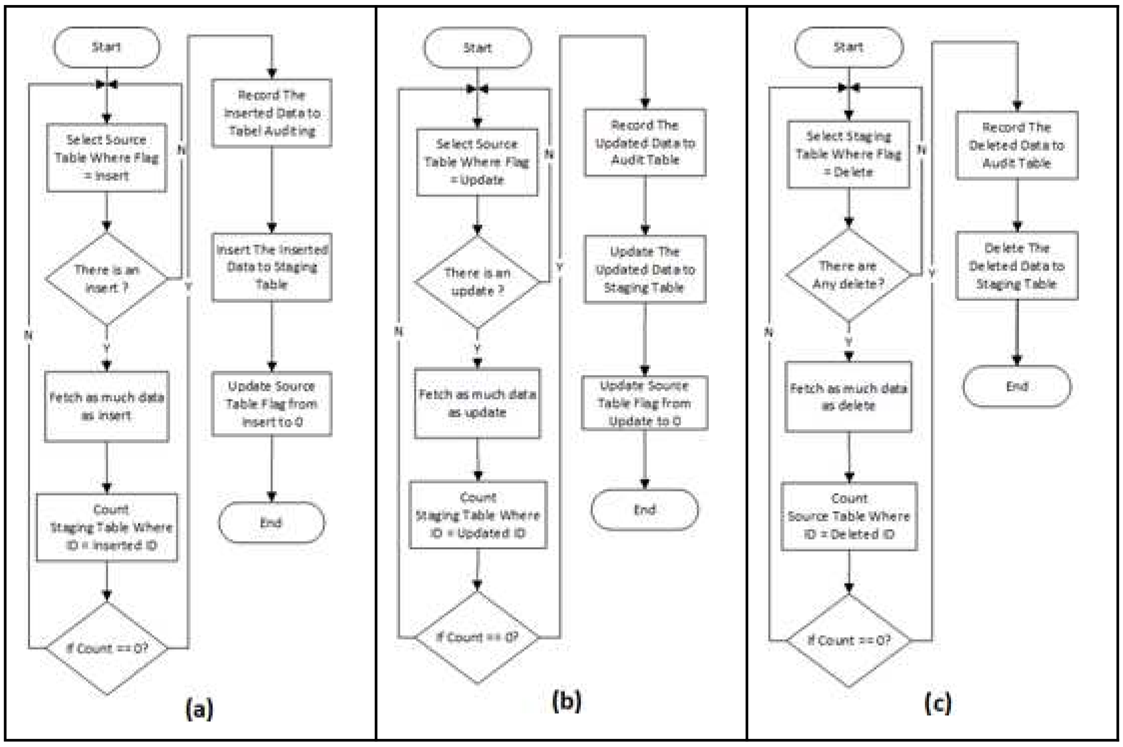

Start
Select Source Table Where Flag = Insert
There is an insert ?
Fetch as much data as insert
Count Staging Table Where ID = Inserted ID
If Count == 0?
Record The Inserted Data to Tabel Auditing
Insert The Inserted Data to Staging Table
Update Source Table Flag from Insert to 0
End
N
Y
(a)
Start
Select Source Table Where Flag = Update
There is an update ?
Fetch as much data as update
Count Staging Table Where ID = Updated ID
If Count == 0?
Record The Updated Data to Audit Table
Update The Updated Data to Staging Table
Update Source Table Flag from Update to 0
End
N
Y
(b)
Start
Select Staging Table Where Flag = Delete
There are Any delete?
Fetch as much data as delete
Count Source Table Where ID = Deleted ID
If Count == 0?
Record The Deleted Data to Audit Table
Delete The Deleted Data to Staging Table
End
N
Y
(c)

Figure 2. Flow *Auditing* (a) *Insert*, (b) *Update*, (c) *Delete*

When an insert event happens, the engine will capture the data that was inserted and save it as a new record in the table in the auditing auditing suitable database. When the engine receives an update event for one or more fields, it records the changes and saves them as a new record in the table in the auditing auditing suitable database. Similarly, when a record is destroyed, the engine synchronisation will capture it and store it as a new record in the relevant auditing table. The Engine will synchronise against three types of data modification events: insert, update, and delete.

Figure 2 shows how the engine's synchronisation mechanism for constructing auditing works. The procedure begins with the selection of a staging table that corresponds to the table that has already undergone DML events in the database. When data changes, the function will fetch as much data as it changes, either inserting, updating, or deleting it. Then the function will conduct the counting on the source system table depending on the ID of the modified data. The modified data is logged to the table auditing if the process counting result is zero.

All DML actions, such as insert, update, and delete, are required for auditing DBMS synchronisation. Auditing DBMS synchronization-based has one flaw: not all database actions, such as DCL and DDL, can be logged. Each table in the source system will be split into three events: insert, update, and delete. The data generated from the synchronisation function of each DML event that happens in the database exists as the input data for table auditing.

3. Literature Review

3.1. Database Auditing

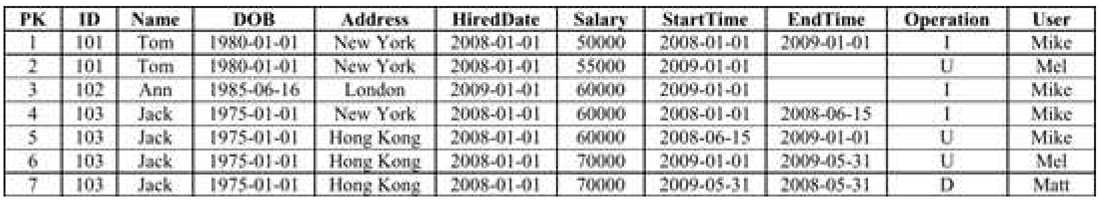
Auditing is a process that involves monitoring and recording activities from a user's database. An audit trail is the output of the auditing process. The audit trail's contents contain records that detail what occurred to the database. Each DBMS has its own restrictions in terms of the number of records or event records it can manage. [8] Database auditing, according to Meg Coffin Murray, may be used to determine who accessed the database, what actions were performed, and what data was modified.

Auditing activities and database access can aid in identifying and resolving database security concerns. Because auditing analyses the record of activities, procedures, and behaviour of organisations or individuals, it plays a critical role in ensuring compliance with the rules [7].

The ability to follow changes in the data trail, what modification actions were performed, and when they occurred using historical data is one of the keys to successful auditing. Historical data may be modelled relationally in databases using a variety of approaches, including distinct tables for historical records, transaction logs, and multidimensional data. Some proposed approaches, such as Row-based auditing or Column-based auditing [3], can be used to keep historical data in auditing.

3.1.1. Auditing With *Row-Based*

To retain historical data, these approaches establish a distinct table in each relational table. The operating table is identical to that of the non-auditing system. For business activities, the operations table merely keeps track of the current value of each item. Static and historical statistics are also included in the table. Data that is static or changes seldom remains unaltered. Only the most recent number will be kept in the operating table for historical data.



PK	ID	Name	DOB	Address	HiredDate	Salary	StartTime	EndTime	Operation	User
1	101	Tom	1980-01-01	New York	2008-01-01	50000	2008-01-01	2009-01-01	I	Mike
2	101	Tom	1980-01-01	New York	2008-01-01	55000	2009-01-01		U	Mel
3	102	Ann	1985-06-16	London	2009-01-01	60000	2009-01-01		I	Mike
4	103	Jack	1975-01-01	New York	2008-01-01	60000	2008-01-01	2008-06-15	I	Mike
5	103	Jack	1975-01-01	Hong Kong	2008-01-01	60000	2008-06-15	2009-01-01	U	Mike
6	103	Jack	1975-01-01	Hong Kong	2008-01-01	70000	2009-01-01	2009-05-31	U	Mel
7	103	Jack	1975-01-01	Hong Kong	2008-01-01	70000	2009-05-31	2008-05-31	D	Matt

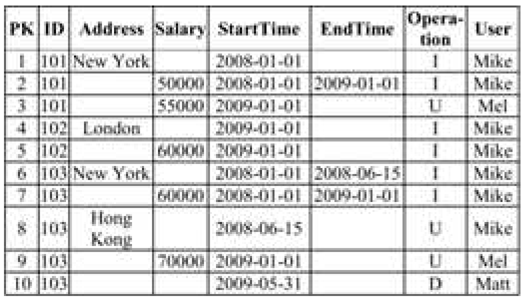
Figure 3. *Row-based Auditing*

As illustrated in Figure 3, the table auditing contains the values for each column of the operational table. Static data is included in table auditing to minimise the operation join query. To retain the age of the data, two timestamps of time are necessary for valid time, namely start time and end time. To decrease comparisons between the same historical data, operation kinds are documented.

3.1.2. Auditing With *Column-Based*

[3] auditing based on the auditing redundancy column. row-based auditing Except for the primary key, such as ID, which is used for reference in the operational table, the data in the historical column of the auditing table only keeps the values that have changed. Because

the time end of each data auditing is unknown, each record in the table auditing column-based may not contain more than one historical data value.



PK	ID	Address	Salary	StartTime	EndTime	Opera-tion	User
1	101	New York		2008-01-01		I	Mike
2	101		50000	2008-01-01	2009-01-01	I	Mike
3	101		55000	2009-01-01		U	Mel
4	102	London		2009-01-01		I	Mike
5	102		60000	2009-01-01		I	Mike
6	103	New York		2008-01-01	2008-06-15	I	Mike
7	103		60000	2008-01-01	2009-01-01	I	Mike
8	103	Hong Kong		2008-06-15		U	Mike
9	103		70000	2009-01-01		U	Mel
10	103			2009-05-31		D	Matt

Figure 4. *Column-based Auditing*

3.2. Synchronization *Database*

Database synchronisation is a procedure that tries to maintain the consistency of data stored on one database server with data stored on another database server. The synchronisation database has a data copy function (copying), which stores data in a table database, either regularly or in real-time. The presence of an asynchronization function database allows data to be updated in real-time or on a regular basis on the database that is being synchronised. In a database management system, this synchronisation function is the foundation for replication (Database Management System) [9].

Synchronization, which is part of a replication database, is a mechanism for distributing and replicating data between databases in order to ensure data consistency [10]. The synchronisation function allows data to be distributed to multiple hosts across a computer network on a regular basis or in real time, depending on the time range. synchronisation Databases may help business applications run smoothly, as well as disseminate data for a variety of objectives, such as enhancing the performance of commercial transactions, decision-making systems, or processing distributed systems over several servers [9].

3.3. SIDEKA (Village and Regional) Village and Regional

In Tangkas Village, Klungkung Regency, the Information System (SIDEKA) is a village data management information system that covers population data management, regional data, and other topics. Data may be accessed by inputting a data master and seeing various data histories.

3.4. Database Management System (DBMS)

A database management system (DBMS) is a computer programme that performs data management tasks such as insertion (insert), alteration / modification (update), deletion (delete), and obtaining data / information (select) for specific purposes. The following are some of the benefits of using a DBMS as a data management system [9] :

1. Practical: DBMS provides a feature of permanent data storage media with a small size but can store a lot of data.

2. Fast: DBMS as a computer application can find and display the required information quickly and accurately.
3. *Up-to-date*: The information available is always changing and accurate at any time.

Homogeneous DBMS and heterogeneous DBMS are two types of database management systems. All locations use the same DBMS product in a homogeneous DBMS system. Meanwhile, heterogeneous DBMS systems are made up of a variety of DBMS products, including non-uniform data models, thus heterogeneous DBMS systems include relational, network, hierarchical, and object-oriented data models [9].

4. Results and Discussion

Each sub-chapter contains an analysis of the results and a discussion of the study, provided in the form of descriptions and explanations.

4.1. Existing System Analysis The existing

SIDEKA is a database auditing solution that was used as a case study (Village and Regional Information System). This system manages two tables: the family table and the population table, both of which are objects in the auditing process.

4.2. Design *Auditing*

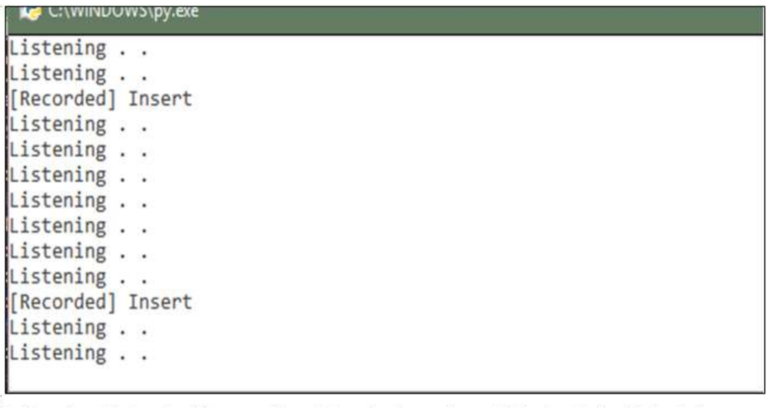
This study employs a model Row-based Auditing based on many types of current auditing databases. To carry out the process auditing, this approach employs a table auditing distinct from the present operating table. Table auditing comprises the value of each column in the existing table, which is displayed historically with the addition of flags, time begin, time end, and st code.

The implementation of the process audit is made easier with this model row-based auditing. The synchronisation engine may simply transfer each value in the record into the table auditing when DML commands such as insert, update, and delete are done on the operations table. At the same time, the prior history's "time_end" column should be updated with the time the operation took place.

4.3. Testing Results

Auditing of tests This is accomplished by changing data in the current SIDEKA (Village and Regional Information System) database, which has an impact on the staging database during the synchronisation process and transmits data changes to the auditing database. The test is carried out by adding data into the family table using the DBMS application client SQLyog to test the process insert.

The results are placed in the table with"id_family"worth "368" in the family row, and the results are inserted in the table with"id_resident"worth "1340" in the second row. When the engine detects a DML process in an existing table, it will synchronise the relevant table in the staging database and store the data changes to the table auditing. Figure 9 depicts the engine recording DML events from the family table process insert.



```
C:\WINDOWS\py.exe
Listening . .
Listening . .
[Recorded] Insert
Listening . .
Listening . .
Listening . .
Listening . .
Listening . .
Listening . .
Listening . .
[Recorded] Insert
Listening . .
Listening . .
```

Figure 9. The *Engine* Works to Capture Changes in the *Insert* Existing Table

Simultaneously, the engine will Me-record DML activity insert that occurs in the family table and the table and save them as residents of row a new in the table auditing, along with some additional attributes such as field "time begin"and"TIME END" to determine the age or effectiveness of a row/data, field "st code" to identify the DML process that occurs from the data/row, and field "flag" to indicate whether the data/row is active / inactive.

The data *inserted* in the existing family table with “id_family” = “368” and in the population table with “id_resident” = “1340”, has been stored as historical data in the table *auditing*. The next test is *the event* DML *update* which will be tested on the same table, namely the family table and the population table.

The results of the update on the family table are displayed in the row with the value of “family_id” of “368,” as well as the results of the update on the population table with the value of “id_resident” of “1340.” The family table is updated by changing the values of “VILLAGE AMBENGAN VILLAGE TANGKAS”, “Y”, and “Y” in many fields, including “address”, “is_raskin”, and “is_jamkesmas”. While the process of updating the population table is accomplished by applying the above to the field "monthly_income" with the value "3000000."

When a process update is performed on an existing table, the engine detects the DML operation and instructs the engine to synchronise the relevant table in the staging database and store the data changes to the table auditing.

The data *updated* in the family table with “id_family” = “368” and in the population table with “id_population” = “1340”, historically stored as a *row* new in the table *auditing*. The data *updated* in the existing table shows that the data has an active value replacing the data recorded from the *insert* in the previous process, which is indicated by thewith *fielda* "flag"value of "1".

Data with "id family" with a value of "368" is deleted in the family table, while data with "id_resident" with a value of "1340" is deleted in the population table. When a DML process happens in the table Existing, the engine's function will recognise the DML process and cause the engine to synchronise the relevant table in the staging database and store the data changes to the table auditing as a new row.

The data *deleted* in the family table with “id_family” = “368” and in the population table with “id_population” = “1340”, historically stored as a *row* new in the table *auditing*. Thedata *deleted* in the existing table shows that the data has been deactivated from the data records *insert* and *update* in the previous process, which is indicated by thewith *fielda* "flag"value of "0".

5. Conclusion

Based on the results of the above-mentioned research, it can be determined that the auditing table should be isolated from the operating table. This is done in order to separate the transaction workload from the analytic workload. When the table is auditing separately from the operational table, the DBMS (engine Database Management System) may conduct auditing queries faster than when the table is auditing as part of the operational system. Furthermore, the DBA (Database Administrator) may more simply administer the DBMS. Preventing engine performance deterioration, reducing data redundancy, saving storage, and simplifying query auditing are all benefits of choosing the proper architecture. The auditing findings can be saved for use by the DBA in analysing access patterns and making changes to data in the database.

References

[1] W. Wisswani, “Implementation of Hybrid Slowly Change Dimensions for Nearly Real Time Data Warehouses,” Lontar Komput., vol. 4, no. 1, 2013.

[2] S. A, M. Sukarsa, and W. B, "Formation of a Data Mart Using the Generalization Method," Lontar Komput., vol. 7, no. 3, 2016.

[3] N. Waraporn, “Database Auditing Design on Historical Data,” Proc. Second Int. sym. net. net. Secur., 2010.

[4] W. Lu and G. Miklau, “Auditing a Database Under Retention Restrictions,” IEEE Int. conf. Data Eng., 2009.

[5] C. Mullins, “Database Auditing Capabilities for Compliance and Security,” The Data Administration Newsletter, 2008. [Online]. Available: http://tdan.com/database-auditing-capabilities-for-compliance-and-security/8135 .

[6] R. Gudakesa, M. Sukarsa, and A. Sasmita, “Two-Ways Database Synchronization In Homogeneous DBMS Using Audit Log Approach,” J. Theor. app. inf. Technol., vol. 65, no. 3, 2014.

[7] L. Yang, “Teaching Database Security and Auditing,” Proc. 40th ACM Tech.Symposium Comput. science. Educ., 2009.

[8] MC Muray, “Database Security: What Students Need to Know,” J. Inf. Technol. Educ. Innov. practice., 2010.

[9] H. Surya, "Design of Two-Way Database Synchronization Application in Homogeneous DBMS With Binary Log Approach," Udayana University, 2014.

[10] MC Mazilu, “Database Replication,” Database System. J., vol. 1, no. 2, 2010.