optimization of transaction database

by Hani Mumtahana

Submission date: 20-Dec-2022 10:15AM (UTC+0700)
Submission ID: 1984825961
File name: 11528-Article_Text-6741-1-15-20220708.docx (226.99K)
Word count: 3306
Character count: 17555

Optimization of Transaction Database Design with MySQL and MongoDB

Submitted : xxxxx | **Accepted** : xxxx | **Published** : xxxxx

Abstract: Database is one of the most important parts in running an application. Databases can be used to store data. However, in running an application, the selection of an appropriate database needs to be considered, so that the resulting information can be in accordance with user needs. In developing applications, more people use RDBMS design which has a very structured nature, but technological developments have introduced NoSQL as a database development method. In this research, MySQL and MongoDB databases will be tested in transaction processing. The tests carried out include comparisons of database designs, comparisons of defining table structures, comparisons of running time in the insert, update, delete, search processes. The test results show that MongoDB has a simpler table description structure than MySQL. The results of the running time test show that MongoDB has a faster running time difference than MySQL. In the insert process there is a time difference of 0.005625 sec, update 0.001688 sec, delete 0.00075 sec and search 0.006875 sec.

Keywords: database, MySQL, MongoDB, NoSQL, RDBMS

INTRODUCTION

Database is a collection of data that is integrated and processed to produce information that is useful for its users (Nahrun et al., 2017). Utilization of databases in application development is a very important part. Without a database an application will not be able to produce the information needed by the user.

The development of Information technology and Information Systems requires every company or organization that uses database services to be able to follow market trends and competition. In addition, the very high investment value for technology-based projects is also a consideration for management (*Masa Depan Teknologi Basis Data | Hypernet*, 2018). Database development began with the Integrated Data Store generation in 1960. In 1970, it was introduced to the relational database generation and became the most widely used database model by application developers to date (Heal, 2016). The generation of relational databases (RDB) developed in tandem with the use of the SQL database language (Structured Query Language).

The growth in the data needs of every company or organization in presenting information is getting higher, giving rise to the term 'Big Data'. 'Big Data' technology is able to provide large volumes of data, high speed, real-time and large data storage capacity (Ramzan et al., 2019). As data needs grow, database users are starting to turn to unstructured data processing. In addition, the emergence of the need for 'cloud storage' which presents data in real time is no longer able to be handled by the RDB generation or currently better known as SQL. Big Data has structured, semi-structured and unstructured data patterns so that it is able to process and store data up to 1TB (Terabyte) in size, while RDB is only capable of storing data in the Gigabyte range.

NoSQL (non-relational database) began to emerge as the right choice as an alternative to unstructured data storage (Moniruzzaman & Hossain, 2013). Currently, web application developers prefer NoSQL, because NoSQL is made to support modern applications with a more specific data model and has a flexible schema.

In making the database design, several stages will be passed so that it can produce a database implementation that is able to produce valid information. Several stages that must be carried out are analysis of data requirements, conceptual database design, transformation of database design on the DBMS and query execution. In this study, a comparison of database design using MySQL and MonggoDB will be carried out.

The need for data transactions in the current digital era is a very crucial thing to prove the performance of a business/industry. Service and delivery of information that is fast, efficient and precise is a high selling point. The trials carried out (Bhaswara et al., 2017)(Halimi et al., 2021)(Deari et al., 2018)(Győrödi et al., 2022) in conducting CRUD testing, NoSQL with MongoDB has better performance compared to RDBMS databases.

In this study, a comparison of the MySQL and MongoDB database design stages will be carried out in a sales transaction. The purpose of this study is to compare the design stages in database creation, so that conclusions can be drawn about the use of the right database used in the process and scale of transactions that will be implemented by a business/industry.

Commented [MA1]: why compare only MongoDB and MySQL is not included Oracle, Ms. Acess, SQL Server, Postgree SQL, SQL Lite, Teradata and more

yes in only MySQL and MongoDB (RDBMS vs NoSQl model database)

LITERATURE REVIEW

Database

In its development, the database is one of the most important parts in processing data into information. Database is defined as a collection of records that are integrated with each other (Connolly & Begg, 2015)(Mumtahana, 2021)(Raut Professor, 2017). Database is a procedure for processing data into information that can be utilized by users according to their needs. The information is obtained from the relationships that occur between data in a particular community (Susilo, 2017).

There are several purposes of using the database, including:

1. Facilitate data storage even on data that has a very large value.
2. Reduce data redundancy
3. Maintain data accuracy
4. Accommodate data so that the availability of data needed by users can be available
5. Maintain data security
6. Provide user sharing
7. Provide complete data

Database Management System is an application software that is used to manipulate data, data management and access control in a database (Susena et al., 2015)(Connolly & Begg, 2015). There are several DBMS that are widely used, including Oracle, MySQL, Ms. Acess, SQL Server, Postgree SQL, SQL Lite, Teradata and more.

Commented [MA2]: MongoDB need to compare with this

No, this is only teory of database

Database System is a collection of databases that have mutual relationships and several applications that are integrated with each other so as to produce information (Putra et al., 2020). Several components of the Database System are (1) Hardware, (2) Operating System, (3) Database, (4) DBMS, (5) User and (6) Application.

The database is used to generate information from a collection of related data by giving the Query command. Queries are used to manipulate data so that the information can be presented in real time. In addition, databases are needed by various groups, for example, businesses, academics, organizations, governments, companies and even small agencies/organizations that want to present information in real time.

Relationship Database Management System (RDBMS)

Relationship Database Management System (RDBMS) which uses the principle of relations between tables to generate information. RDBMS is a database that is based on the concept of relationship which has a basic mathematical theory and relationship theory (Ramzan et al., 2019). Every existing data is stored in a table, while to produce a good data integration, there is a relationship that occurs between tables in a particular community. Meanwhile, to perform data manipulation, RDBMS uses SQL which is written in a structured manner .

Not Only Structure Query Language (NoSQL)

Not Only SQL or often known as NoSQL is a database design that was created with the aim of modeling special data that is more flexible and schematic and does not use SQL to manipulate data (Moniruzzaman & Hossain, 2013)(Palanisamy & Suvithavani, 2020). NoSQL databases do not apply the ACID (atomicity, consistency, Isolation, durability) rules that are applied to the RDBMS (Nance et al., 2013).

METHOD

In this study, a comparison of MySQL and MongoDB database designs will be carried out in the case of transactions. The designs that will be made are the database structure design, table structure design and database performance in the CRUD (Create, Read, Update, Delete) process. The steps in this research are as follows:

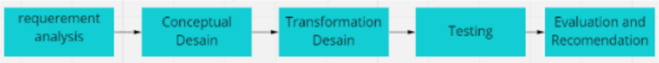


Fig. 1. Research Steps

Figure 1 shows the steps taken in this research. The first step of the research is to analyze the data and information needs of a transaction. In the next step, create a conceptual database design in the form of a table/document structure design and a relation/join structure between tables. After the conceptual design is made, the next step is to transform MySQL and NoSQL. The next stage is to test the CRUD process (Create, Read, Update, Delete) so that it will produce recommendations for the appropriate database to use.

RESULT

In this study, a sample of data/information needs from sales transactions at a basic food store will be carried out. This research has several results, namely:

1. Analysis of sales transaction data needs

Processing data into information in the sales transaction process results in an analysis of data and information needs in the form of:

a. Information needs as follows:
   1) Data of goods to be sold
   2) Transaction data that occurs in the sales process
b. Data requirements are as follows:
   1) Goods
   2) Transaction
   3) Detail_transaction

2. Table/Document Structure Design

In this section, a conceptual table/document structure design will be made on MySQL and MongoDB. In MySQL the data/information needs can be stored or manipulated in a clear table structure of data types, character length and selection of key attributes. However, MongoDB stores data/information needs in the form of a more flexible document. The description of the needs of the table structure in MySQL is as follows:

a. Produc

Table 1. structure produc

Kode_barang	Char	6	Primary Key
Nama_barang	Varchar	30	-
Harga_barang	Money	-	-
Qty	Int	-	-

In Table 1 is a description of the structure of information needs about produc.

b. Transaction

Table 2. structure transaction

no_transaksi	Char	6	Primary Key
no_detail_tr	Int	-	Foreign_key
Total_pembayaran	Money	-	-

In table 2 is a description of structure of information needs about transaction.

c. Detail_transaction

Table 3. structure detail_transaction

no_detail_tr	Int	-	Primary_key
Kode_barang	Char	6	Foreign Key
Jum_beli	Int	-	-
Total	Money	-	-

In table 2 is a description of structure of information needs about detail_transaction.

Meanwhile, data/information storage with MongoDB uses the JSON concept, where data/information is stored in the form of documents that are more dynamic and flexible in the concept of connecting one data to another.

```
db.barang.insertone ({
_kode_barang :"brg001",
nama_barang :"kerupuk",
qty : 24,
harga : 5000
}
```

Fig. 2. Conceptual design of a document in MongoDB.

Figure 2 is a conceptual design of a document in MongoDb. mongoDb has a more flexible design independent of defining data types and required character lengths. This makes data storage on MongoDb adaptable to information needs.

3. Relationship/Join

The relation in MySQL and MongoDB conceptually has the same understanding. Where the relationship is to connect data dependencies from one table/document with other tables/documents. The following describes the relationship of the three tables in MySQL.

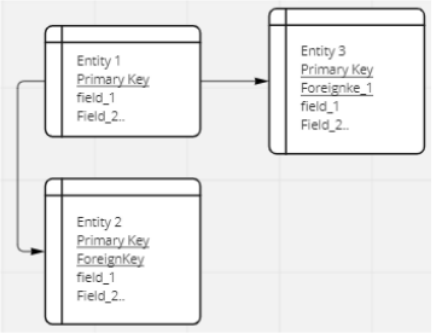
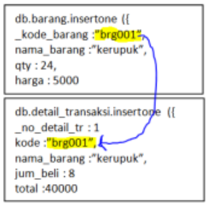


Fig 3. Relational design in RDBMS

The relationship design in Figure 3. above illustrates the relationship of 3 tables that are mutually dependent on the delivery of information. Each table has one attribute that is selected to be the primary key attribute (Primary Key / PK). In the Items table, the primary key is item_code. From the picture above, the item_code which becomes a PK in the goods table appears as a Foreign Key (FK) in the transaction_detail table. This shows that there is a need for dependency on the PK attribute to become the FK attribute from a table that is related to each other. While the concept of relations in MongoDB is explained in the image below:



```
db.barang.insertone ({
_kode_barang :"brg001",
nama_barang :"kerupuk",
qty : 24,
harga : 5000
```

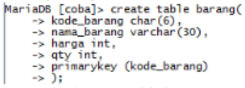
```
db.detail_transaksi.insertone ({
_no_detail_tr : 1
kode :"brg001",
nama_barang :"kerupuk",
jum_beli : 8
total :40000
```

Fig. 4. JOIN concept in MongoDB

The form of a relationship in MongoDB can be shown by the presence of the same data in a transaction document that will be combined with other documents. The term relation between tables in MongoDB is often referred to as a join collection by using a function in MongoDB so that it is able to generate the data/information needed. In making the join concept in MongoDb, it is not required to describe the column name/attribute FK must be the same as the PK attribute in the master table. In the example of the join concept in Figure 4 there is a PK, namely _kode_barang, but in the transaction_details table the code is written, but with the same details as the data in the previous table. This proves that the dynamic nature of the MongoDB database is more flexible than the RDBMS.

4. Implementation MySQL

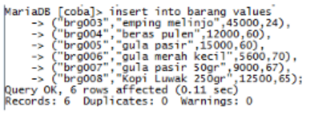
This section will explain the implementation of the database design results using MySQL and MongoDB. The implementations applied are: Create, Insert, and select.



```
MariaDB [coba]> create table barang(
    -> kode_barang char(6),
    -> nama_barang varchar(30),
    -> harga int,
    -> qty int,
    -> primarykey (kode_barang)
    -> );
```

Fig 5. create table barang

Figure 5 is the result of implementing the goods create table. In the goods table there are 4 columns that have data type criteria and character length according to information needs.



```
MariaDB [coba]> insert into barang values
    -> ("brg003","emping melinjo",45000,24),
    -> ("brg004","beras pulen",12000,60),
    -> ("brg005","gula pasir",15000,60),
    -> ("brg006","gula merah kecil",5600,70),
    -> ("brg007","gula pasir 50gr",9000,67),
    -> ("brg008","Kopi Luwak 250gr",12500,65);
Query OK, 6 rows affected (0.11 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

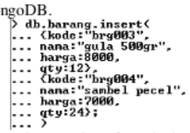
Fig. 6. Insert Items table data

After creating the table structure, the next step is to add data/records to the table with the insert command. In Figure 6, insert data in the goods table with a total of 6 records. It can be seen that the running time required for MySQL to insert data is 6 records with a time of 0.11sec.

From the two experiments, it can be concluded that the RDBMS query using the MySQL DBMS is a structured query. Each query that is written must be adapted to the structure of the SQL language.

5. Implementation MongoDB

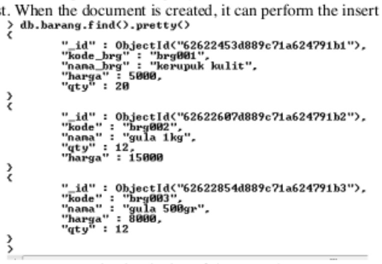
In this section, we will test the implementation of the database design results using MongoDB. The following will be implemented on MongoDB.



```
> db.barang.insert(
... {kode:"brg003",
... nama:"gula 500gr",
... harga:8000,
... qty:12},
... {kode:"brg004",
... nama:"sambel pecel",
... harga:7000,
... qty:24};
... )
```

Fig.7 Implementation of goods documents

In MongoDb to create, inserting a document is technically more dynamic and flexible than RDBMS. Figure 7 is an implementation for creating goods documents. From the picture it can be seen, the description to create is not as complicated as MySQL. Item documents are described without having to define their structure first. When the document is created, it can perform the insert stage.



```
> db.barang.find().pretty()
{
        "_id" : ObjectId("62622453d889c71a624791b1"),
        "kode_brg" : "brg001",
        "nama_brg" : "kerupuk kulit",
        "harga" : 5000,
        "qty" : 20
}
{
        "_id" : ObjectId("62622607d889c71a624791b2"),
        "kode" : "brg002",
        "nama" : "gula 1kg",
        "qty" : 12,
        "harga" : 15000
}
{
        "_id" : ObjectId("62622854d889c71a624791b3"),
        "kode" : "brg003",
        "nama" : "gula 500gr",
        "harga" : 8000,
        "qty" : 12
}
>
```

Fig. 8. Display of data search

Figure 8 shows the view of the data search on MongoDB. From the picture it can be seen, the index for each data is indicated by the _id that is owned by each object. The ObjectId will be used as a pointer when a JOIN occurs.

6. Testing

In research (Bhaswara et al., 2017), they compare the performance, flexibility and scalability of the MongoDB and MySQL DBMS. The test results show that each DBMS has its own advantages. MySQL performance value in operating JOIN is better than MongoDB.

Meanwhile, MongoDB has a good performance value in managing the CRUD process. Table 2. shows the results of performance testing of MySQL and MongoDB.

Table 4. MySQL and MongoDB Performance Test Results

	MySQL				**MongoDB**			
Data	**25**	**50**	**75**	**100**	**25**	**50**	**75**	**100**
Insert	0.045	0.080	0.096	0.112	0.005	0.013	0.027	0.045
Update	0.015	0.037	0.059	0.076	0.002	0.005	0.008	0.014
Delete	0.013	0.025	0.047	0.077	0.003	0.005	0.007	0.009
Search	0.035	0.055	0.077	0.096	0.012	0.022	0.045	0.063

Table 4. shows the difference in run time between MySQL and MongoDB in the insert, update, delete and search processes. The test was carried out on multiples of the first 25 data, 50 data, 75 data and 100 data. The comparison of test results is described in the graphic image below:

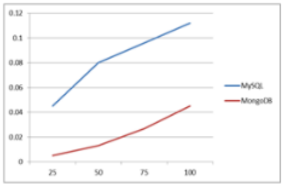


Fig. 9. Comparison of insert times

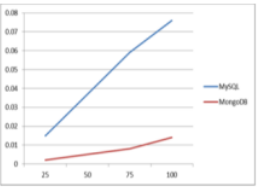


Fig. 10. Comparison of update times

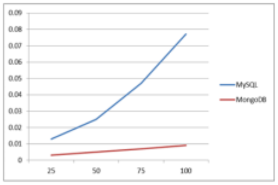


Fig. 11. Comparison of delete times

Figure 9 above is the result of a comparison of the insert test data on MySQL and MongoDB. Figure 10 above is the result of a comparison of the trials on the data update command. Figure 11 above is the result of a comparison of the trials on the delete command.

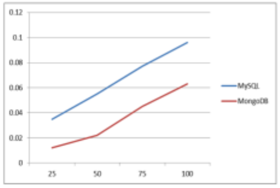


Fig. 12. Comparison of Search times

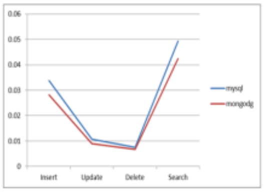


Fig. 13. Performance comparison of MySQL and MongoDB

Figure 12 above illustrates the comparison of the test results on the search data command. Data search is carried out in a simple data search process, data search by applying JOIN, and data search to perform transactions using aggregation functions. From the test results on each of the commands mentioned above, the average running time in each execution results is obtained as shown in Figure 13.

The test results above show that MongoDB has better performance than MySQL. In the application of the insert command between MongoDB and MySQL there is a time difference of 0.005625 sec. the update command has a time difference of 0.001688 sec, delete has a time difference of 0.00075 sec and search has a time difference of 0.006875 sec. From the test results, JOIN MongoDB requires a command that combines several functions, so it requires a longer running time, while in MySQL the JOIN table command is simpler by only providing the functions needed to process data.

DISCUSSIONS

From the results of the trials that have been carried out, it can be formulated several recommendations for selecting a database for processing transaction data. The recommendations given are as follows:

a. In the RDBMS database design, it is necessary to describe the table structure according to the information needs. So that in describing the structure of the database, the RDBMS has a structured and inflexible nature.
b. In NoSQL database and table design is more flexible and there is no need for a description of the table structure as in RDBMS.
c. For transactions that are dynamic and require bid data, a NoSQL database design can be used, while for more complex transaction data storage, an RDBMS database design can be used because with an RDBMS, the processing of transaction data can be carried out in a structured manner and is simpler in using commands.
d. In data processing, NoSQL databases have faster access times than RDBMS databases.

CONCLUSION

From the results of research conducted to assess the optimization of the selection of RDBMS (MySQL) and NoSQL (MongoDB) databases in the transaction process, it shows that MongoDB has a faster running time than MySQL. In addition, in describing the table structure, MongoDB has simpler commands than MySQL. However, in processing transactions that are more complex, MySQL has better performance, because the commands in MongoDB require writing more complex commands by implementing functions in it.

REFERENCES

1. Bhaswara, F. A., Sarno, R., & Sunaryono, D. (2017). Perbandingan Kemampuan Database NoSQL dan SQL dalam Kasus ERP Retail. *Jurnal Teknik ITS*, *6*(2), 510–514. https://doi.org/10.12962/j23373539.v6i2.24031
2. Connolly, T., & Begg, C. (2015). *Database Systems*.
3. Deari, R., Zenuni, X., Ajdari, J., Ismaili, F., & Raufi, B. (2018). Analysis and comparision of document-based databases with relational databases: MongoDB vs MySQL. *2018 International Conference on Information Technologies, InfoTech 2018 - Proceedings*, *September*, 20–21. https://doi.org/10.1109/InfoTech.2018.8510719
4. Győrödi, C. A., Dumse-Burescu, D. V., Zmaranda, D. R., & Győrödi, R. Ş. (2022). A Comparative Study of MongoDB and Document-Based MySQL for Big Data Application Data Management. *Big Data and Cognitive Computing*, *6*(2), 49. https://doi.org/10.3390/bdcc6020049
5. Halimi, A., Sudarmanto, A., Utami, E., & Kusnawi. (2021). ANALISIS PERBANDINGAN KINERJA WAKTU RESPON MYSQL 8.0 DAN NOSQL MONGODB MENGUNAKAN RESTAPI NODEJS PADA STUDI KASUS KELAS ONLINE. *Informatika Wicida*, *10*(1), 26–33.
6. Heal. (2016). *Perkembangan Database – Heal ur program*. Syifasalsabyla.Wordpress.Com.
7. *Masa Depan Teknologi Basis Data | Hypernet*. (2018). Hypernet.Co.Id.
8. Moniruzzaman, A. B. M., & Hossain, S. A. (2013). *NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison*.
9. Mumtahana, H. A. (2021). Basis Data (Teori dan Praktek dengan Query SQL). In *Unipma Pers* (Vol. 1). https://www.academia.edu/8558159/Basis_Data
10. Nahrun, H., Kurniadi, W., & Kasma, S. (2017). OPTIMASI JOIN QUERY PADA BASIS DATA (STUDI KASUS : BASIS DATA SISTEM INFORMASI MANAJEMEN UNIVERSITAS COKROAMINOTO PALOPO). *PROSIDING SEMANTIK 2017*, 209–215.
11. Nance, C., Losser, T., Iype, R., & Harmon, G. (2013). NOSQL VS RDBMS-WHY THERE IS ROOM FOR BOTH. *Proceedings of the Southern Association for Information System Conference*, 111–116.
12. Palanisamy, S., & Suvithavani, P. (2020). A survey on RDBMS and NoSQL Databases MySQL vs MongoDB. *2020 International Conference on Computer Communication and Informatics, ICCCI 2020*. https://doi.org/10.1109/ICCCI48352.2020.9104047
13. Putra, R. A., Alauddin, M. F., Alam, I. N., & Yaqin, M. A. (2020). Pengembangan Arsitektur Data Sistem Informasi Sekolah. *Jurasik (Jurnal Riset Sistem Informasi Dan Teknik Informatika)*, *5*(2), 175. https://doi.org/10.30645/jurasik.v5i2.203
14. Ramzan, S., Bajwa, I. S., Ramzan, B., & Anwar, W. (2019). Intelligent Data Engineering for Migration to NoSQL Based Secure Environments. *IEEE Access*. https://doi.org/10.1109/ACCESS.2019.2916912
15. Raut Professor, A. B. (2017). NOSQL Database and Its Comparison with RDBMS. *International Journal of Computational Intelligence Research*, *13*(7), 1645–1651. http://www.ripublication.com
16. Susena, E., Utami, E., & Sunyoto, A. (2015). PERENCANAAN STRATEGIS SISTEM INFORMASI SMART CAMPUS UNTUK MENINGKATKAN PELAYANAN DI POLITEKNIK INDONUSA SURAKARTA. In *Jurnal Sainstech Politeknik Indonusa Surakarta* (Vol. 1).
17. Susilo, G. (2017). KEAMANAN BASIS DATA PADA SISTEM INFORMASI DI ERA GLOBAL. *JURNAL TRANSFORMASI*, *12*(2).

optimization of transaction database

ORIGINALITY REPORT

16%	15%	5%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

3%

★ www.mdpi.com

Internet Source

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

optimization of transaction database

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8