Performance Analysis of Graph Database and Relational Database

Agustian Prakarsya[1], RZ.Abdul Aziz[2], Ari Kurniawan[3]
ITBis lembah dempo[1], IIB darmajaya Bandar lampung[2],
Kepahiang Street - Pagar Alam No.33, Kuripan Babas, Pagar Alam Utara, Kota Pagar Alam, Sumatera Selatan 31516[1,]
Z.A. Pagar Alam Street, No.93 Labuhan Ratu, Bandar Lampung
Lampung, Indonesia 35141[2]Kota Bandar lampung, 35142, Indonesia[3]
Agustian.prakarsya@gmail.com[1], rz_aziz@darmajaya.ac.id[2] ari.kurniawan@ubl.ac.id

Abstract: There were many database methods for managing data. They were widespread industries that had a huge success. The method used in this study was the relational database method and the graph database. The relational database method considered as the standard for over thirty years. The graph database consisted of several components e.g., nodes, edges, and properties that represented the data storage. The graph database provided index-free adjacency, which meant that each element contained a direct pointer, which was adjacent to the elements and no longer required an index lookup. These databases can been more scaled to the larger datasets as they generally did not require a "joint" operation. Therefore, they were more suited to manage the fluctuating ad-hoc data with changing schemas

Keywords: Relational Databases, Database Graph, Nodes, Edges, and Properties

1. INTRODUCTION

The database was not only a collection of data divided and connected logically and but also a description of the data designed to meet the information needs of an organization (Connolly and Begg, 2010). There were various types of databases designed for specific purposes e.g., relational databases used for transaction data [1], and graph databases used for graphical data with complex relationships e.g., social networks [2]. Graph databases and relational databases were popular databases recently and had many users for babies. The relational databases were outdated and people were looking for new databases for facilitating their work.

The objective of this study was determining whether a graphical database was a better alternative to relational databases or each type of database that had its own advantages and disadvantages, especially in the ability to create, read, update, and delete (CRUD).

2. LITERATURE RIVIEW

2.1 Analysis

Analysis was a problem-solving technique used to break down a system into component parts with the aim of identifying how well these component parts worked and interacted to achieve their goals. The main purpose of the analysis was determining the details about what will been done and what will be proposed.

2.2 SQL vs NoSQL

SQL

SQL (Structure Query Language) was the language for managing databases. It was very easy to read because the fourth spoken language – the syntax has already used words that were easy to read by humans – for example SELECT, FROM, WHERE etc. In SQL, there were two types of language e.g., DDL (Data Definition Language) and DML (Data Manipulation Language). The two types of language had different functions. DDL used to create table structures, while DML used for data in tables that created. SQL commonly used in relational data models. It meant that the tables interconnected to produce the desired information.

NoSQL

NoSQL was not a language. NoSQL was a tool for storing data and retrieving data performed by our database. NoSQL did not require a relational data model and SQL language to do this. NoSQL used metadata in our database and used an index of that data. NoSQL had someone's empathy:

- Table oriented.
- Graphics oriented.
- Document oriented database.
- Key value storage, for example: Memcache and Redis

If we compare SQL with NoSQL, each of the advantages and disadvantages of each

2.3 Graph Database

Graph database was the database model that used the concept of graph theory in which the data was stored in the form of vertices / points / vertices and segments / lines / edges where the vertices and edges reflected real-world entities and relations or referred to the other entities (Angles & Gutierrez 2008; Robinson, et al., 2015; Shimpi, 2013). NOSQL was a family of databases, and graph databases were part of the NOSQL database. NOSQL database was functioned to create a very large data where nothing was not closely related. In NOSQL, large volumes of data were stored efficiently.

Neo4j was currently the most popular database management chart on the market (Van Bruggen, 2014) firstly released in 2007. Neo4j was an open source graphical database implemented by Java and developed by Neo technology (Vicknair et al., 2010). A disk-based, intact, transactional Java-based persistence engine stored structured data in graphs rather than in tabular form. Neo4 consisted of two parts, client, and server.

2.4 Relational Databases

Relational database was a collection of data items organized as a set of formally described tables where the data were accessed or rearranged in many ways without having to reorganize database tables. (Bhugul, 2015).

3. METHOD

3.1 Research Framework

The stage of this section was to establish a framework to run experiments on the aspects that were used as a comparison reference for the two databases e.g., the DML (Data Manipulation Language) and CQL (Cypher Query Language) aspects e.g., the CRUD (Create, Read, Update, Delete) operation. To determine the various objectives, the framework was divided into three phases e.g., preparation, measurement, and evaluation.

3.2 Dataset

In this study, two types of datasets were used e.g., Neo4j for the graph database and MySQL for the relational database. The dataset used in this study was the public data of the Indonesian region. The dataset contained 34 nodes e.g., 34 provinces consisting of around 91162 regions including district, sub-district, and sub-district / village data in accordance with Permendagri No 137 of 2017. The dataset used as an experiment was the smallest amount of area data to the largest data set, which seen below in figure 1.

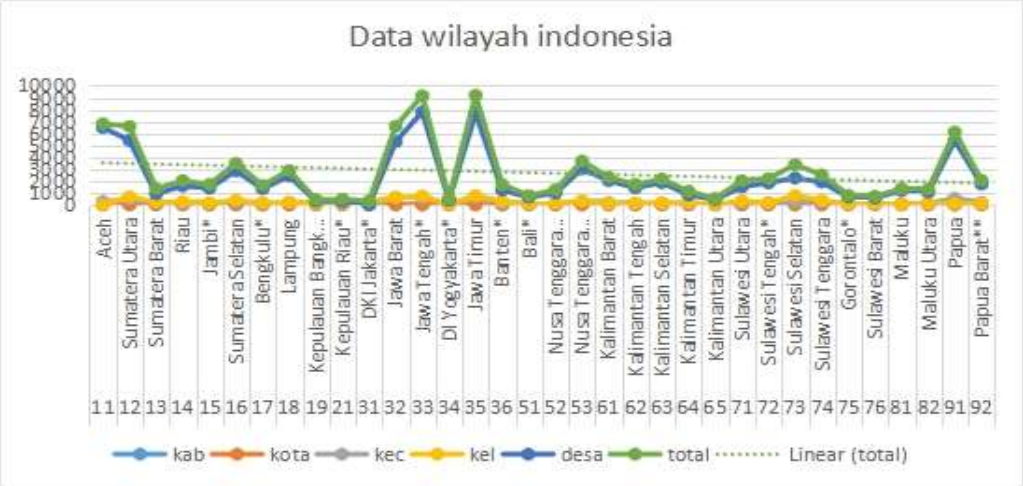

Figure 1. Indonesian Provincial Data

3.3 Relational Database Dml Syntax and SQL Syntax for Graph Databases.

A. The following was the syntax for DML commands in a Relational database

1. Insert

```
INSERT INTO nama tabel VALUES('nilai masukan', 'nilai masukan', 'dst');
```

2. Update

```
UPDATE nama table SET nama column = value where [condition]
```

3. Delete

```
DROP TABLE nama tabel;
```

4. Select

```
SELECT * FROM nama_tabel
```

B. The following is the syntax for CQL commands in the Graph database

1. Insert

```
CREATE (n:nama_node {nama: "orang"})
```

2. Update

```
MATCH (nama_node) where nama_node.nama="orang"
SET nama_node.nama="orang2."
RETURN nama_node
```

3. Delete

```
Match (n:nama_node) DELETE n
```

4. Select

```
MATCH (n:nama_node) RETURN n
```

3.3 Evaluation

Calculation of the average time used for all data from the database. In calculating the average time, there was a function "data. Length" which functioned to retrieve information on the amount of data in the dataset [4]. Data length seen in table 1.

Table 1. Data Length

syntax	MySQL	neo4j
create	15.4694	104.173
read	0.6083	1.180
update	152.0995	1067.271
delete	12.6294	0.738

Meanwhile, "total Time" was the value of the overall time variable obtained from Equation (1), seen in table 2.

Table 2. Date Now

syntax	time Start	time End	total Time
create	11:31	13:55	1.06
read	13:08	14:06	1.13
update MySQL	14:00	18:30	1.35
update Neo4j	12/1/2019 6:30	12/30/2019 14:45	87629.89
delete	20:30	22:15	1.78

4. RESULT AND DISCUSSION

According to the results of the storage test (CRUD) of the dataset into the database, the average time obtained according to Table 3. and Figure 2.

Table 3. Average. Time

Database	create	read	update	delete
MySQL	14.598	0.536	112.320	7.090
Neo4j	98.302	1.040	410.532	0.414

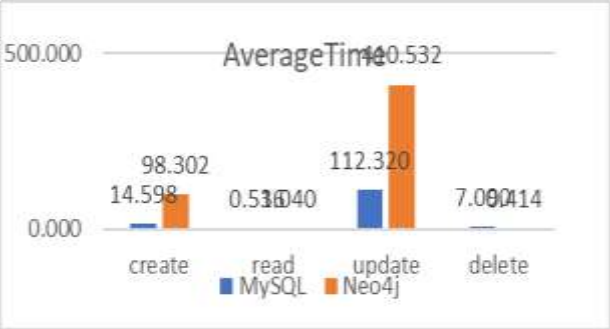

Figure 2. Graphic Average. Time

5. CONCLUSION

According to Performance Analysis, testing of relational database (MySQL) and graph database (Neo4j), there had their respective advantages. Relational database (MySQL) had better performance and speed in creating syntax, with an average time of 14,598 seconds and a read syntax of 0.536 seconds, syntax updates with an average time of 112,320 seconds while the graph database (Neo4j) excels at syntax delete 0.414.

Further research should been carried out by conducting further performance testing on other operations e.g.:

- Throughput. The overall ability of the computer to process data, which was a combination of IO speed, CPU speed, parallel capability and efficiency of the operating system and system software.
- Resources (Resources). Hardware and software, including memory, disk speed, cache controller and so on.
- Memory. The total amount of memory required to complete execution. Create, read, update, delete this value was fetched after the end of execution.
- Please submit your paper in PDF form or Word.

REFERENCES

[1]. Chickerur, S., Goudar, A., & Kinnerkar, A. (2015). Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications. 2015 8th International Conference on Advanced Software Engineering & Its Applications (ASEA), 41–47. http://doi.org/10.1109/ASEA.2015.19

[2]. Kaliyar, R. (2015). Graph Databases: A Survey. International Conference on Computing, Communication and Automation (ICCCA2015),

[3]. Isjhar Kautsar. 2015. Analisis Performansi Metode Graph Decomposition Index pada Graph Database.

[4]. Setialana, Pradana, Teguh Bharata Adji, and Igi Ardiyanto, 2017. "Perbandingan Performa Relational, Document-Oriented dan Graph Database Pada Struktur Data Directed Acyclic Graph." Jurnal Buana