

ANALISIS PERBANDINGAN PERFORMA WAKTU RESPONS KUERI ANTARA MySQL PHP 7.4.2, PostgreSQL, dan MongoDB

COMPARISONAL ANALYSIS OF PERFORMANCE QUESTION RESPONSE TIME BETWEEN MySQL PHP 7.4.2, PostgreSQL, and MongoDB

Rivaldo Hadi Winata¹⁾, Yudha Yunanto Putra²⁾

E-mail : ¹⁾18082010026@student.upnjatim.ac.id , ²⁾18082010032@student.upnjatim.ac.id

^{1,2} Program Studi Sistem Informasi, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur

Abstrak

Di era ini, kebutuhan akan sebuah informasi menjadi hal yang sangat penting bagi perusahaan, organisasi bahkan sampai instansi pemerintahan. Sehingga penggunaan basis data pada sebuah sistem menjadi hal yang tidak terhindarkan. Pemilihan basis data menjadi penting karena penggunaan basis data pada sebuah sistem dituntut tidak hanya menyimpan data dan informasi saja namun juga harus responsif baik dalam mengolah data yang diminta oleh banyak pengguna yang akan berdampak positif terhadap instansi atau organisasi. Pada penelitian ini akan dilakukan pengujian performa waktu respon dari ketiga database yang diujikan yaitu MySQL, PostgreSQL yang merupakan *database* relasional dan MongoDB yang merupakan *database* non-relasional (NoSQL). Pengujian akan dilakukan menggunakan 4 fungsi *query* yaitu *select*, *insert*, *update*, dan *delete* dengan berbagai jumlah *record* data yang berbeda-beda (250, 2500, 5000, 15000, 25000). Adapun data yang digunakan dalam pengujian ini adalah data *Android Permission* yang dimana data tersebut berisi informasi seputar aplikasi android yang terdiri dari 6 kolom dan 29999 baris data. Penelitian ini bertujuan memberikan hasil analisis perbandingan waktu respon setiap *query* diantara ketiga database yang diujikan sehingga, dapat diketahui *database* mana yang performanya paling unggul. Hasilnya terbukti bahwa *database* PostgreSQL paling unggul karena memiliki waktu respon atas setiap *query* yang diberikan hampir pada semua pengujian dimana PostgreSQL waktu responnya tidak pernah lebih dari 150ms selama pengujian dalam penelitian ini.

Kata kunci: *mysql, mongodb, postgresql, perbandingan.*

Abstract

In this era, the need for information is very important for companies, organizations and even government agencies. so that the use of databases on a system is unavoidable. The selection of a database is important because the use of a database in a system is required not only to store data and information but also to be responsive both in processing the data requested by many users which will have a positive impact on the agency or organization. In this study, we will test the response time performance of the three databases tested, i.e MySQL, PostgreSQL which is a relational database and MongoDB which is a non-relational database (NoSQL). The test will be carried out using 4 query functions, which is select, insert, update, and delete with different numbers of data records (250, 2500, 5000, 15000, 25000). The data used in this test is Android Permission data where the data contains information about the android application which consists of 6 columns and 29999 rows of data. This study aims to provide a comparative analysis of the response time of each query among the three databases tested so that it can be seen which database has the most superior performance. The results prove that the PostgreSQL database is the most superior because it has a response time for any given query in almost all tests where the PostgreSQL response time was never more than 150ms during the tests in this study.

Keyword: *mysql, mongodb, postgresql, comparison.*

1. PENDAHULUAN

Menurut Gizen pada tahun 2017, era globalisasi, setiap harinya teknologi informasi semakin berkembang dengan cepat. Salah satu penyebabnya yaitu, pemakaian internet oleh berbagai kalangan masyarakat yang tentunya membuat berbagai instansi pemerintahan hingga swasta menjadi terpicu untuk melakukan peningkatan digitalisasi fungsi kerja. Segala bentuk digitalisasi pada fungsi kerja usaha saat ini merupakan tuntutan yang harus dilalui agar tidak tertinggal oleh zaman dan tergerus oleh teknologi [1]. Menurut Christiono pada tahun 2020, semakin berkembangnya teknologi tentunya kebutuhan akan penyimpanan akan semakin naik. Teknologi penyimpanan yang kini biasa digunakan yaitu DBMS dan NoSQL [2].

Menurut Tavares pada tahun Dalam sebuah aplikasi yang menerapkan *database* sebagai penyimpanan, semakin banyak data yang disimpan serta semakin variatif data dan tipe datanya maka akan mempengaruhi waktu respon dari *database* tersebut. Berbagai cara telah dilakukan untuk memperbaiki hal ini, mulai dari menambah kemampuan perangkat keras seperti memori dan prosesor [3]. Namun, hal ini akan memakan biaya yang tidak sedikit mengingat harga komponen tersebut tidaklah murah. Teknik lain yang bisa dilakukan yaitu melakukan optimasi pada kueri. Mulai dari teknik sederhana seperti optimasi indeks [3]. Hingga penggunaan algoritma kompleks seperti algoritma genetik dan algoritma memetics [4].

Menurut Praba pada tahun 2020 DBMS (*Database Management System*) adalah perantara antara pemakai dengan basis data dari sistem [5]. DBMS dapat membantu pengolahan dan pemeliharaan data dalam jumlah yang besar [6]. Sehingga dengan adanya DBMS maka kekacauan pada pengolahan data dapat dihindari dan dapat dimanfaatkan sesuai dengan kebutuhan dari pengguna [7]. Bahasa basis data biasanya terdiri atas perintah-perintah yang biasa disebut dengan istilah kueri sehingga perintah tersebut akan diproses oleh DBMS [5]. Perintah tersebut biasanya ditetapkan oleh perusahaan atau pengembang dari DBMS tersebut. Perintah apa yang ingin dilakukan dapat ditentukan sendiri oleh user sesuai dengan kebutuhan [5]. MySQL dan PostgreSQL sendiri merupakan salah dua contoh dari DBMS [8][9].

NoSQL bukanlah sebuah bahasa. Menurut Budiman pada tahun 2021 NoSQL adalah mekanisme yang digunakan untuk mengambil data dan menyimpan data yang dilakukan oleh database dari suatu program yang menerapkan penyimpanan NoSQL [10]. NoSQL tidak membutuhkan relasi dari data dan melakukan kueri dalam melakukan hal tersebut [11]. NoSQL menggunakan *key-value* pada database dari suatu program dan memanfaatkan key dari data tersebut untuk mengidentifikasi suatu data [12]. Sehingga, NoSQL menjadi berbeda sekali jika dibandingkan dengan DBMS. MongoDB merupakan salah satu contoh dari NoSQL [13].

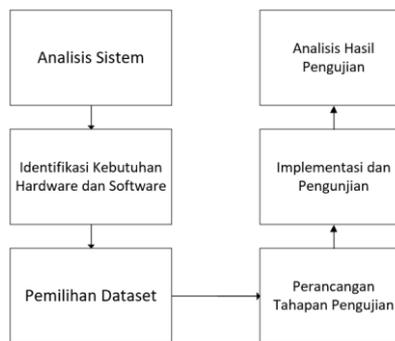
Adanya berbagai teknologi dalam ketiga basis data tersebut menimbulkan berbagai pertanyaan baru, manakah yang memiliki responsivitas yang lebih baik dalam melakukan pemrosesan data [14]. Hal ini menjadi penting karena seiring dengan meningkatnya kebutuhan tidak hanya akan informasinya saja namun juga kecepatan dalam mengakses informasi tersebut [15]. sehingga pemilihan basis data untuk membangun sebuah sistem juga menjadi hal yang krusial terutama di era *Big Data* saat ini [16]. Oleh karena itu, penulis akan melakukan pengujian terhadap 3 database yaitu MySQL, PostgreSQL, dan MongoDB dimana yang akan diuji adalah kinerja waktu respon kueri dari masing masing basis data yang terdiri dari berbagai model pengujian untuk menghasilkan perbandingan yang akurat [17].

Tujuan dilakukannya penelitian ini untuk membandingkan waktu respon pada masing-masing database dengan perintah kueri yang telah disesuaikan pada masing-masing database. kemudian akan dilakukan perbandingan waktu respon dari tiap-tiap database

sehingga dapat diketahui mana yang paling efektif dan efisien dalam mengeksekusi kueri. Pengujian akan dilakukan dengan berbagai skenario yang ada seperti eksekusi fungsi *insert*, *update*, *delete*, *select*, dengan berbagai jumlah data yang berbeda di setiap fungsi. Manfaat dari penelitian untuk mengetahui performa antara basis data sehingga dapat diketahui manakah yang paling efisien dan efektif dalam mengeksekusi kueri sehingga kedepannya dapat dijadikan sebagai acuan dalam pemilihan database mana yang paling cocok dan tepat sesuai kebutuhan sistem.

2. METODOLOGI

Banyak sekali perbedaan yang dimiliki antara DBMS dan NoSQL. Untuk mengetahui perbedaannya pada saat eksekusi khususnya respon waktunya maka akan dilakukan pengujian. Pengujian dilakukan dengan melakukan 4 kueri atau perintah dasar yaitu *INSERT (create)*, *SELECT (read)*, *UPDATE (update)*, *DELETE (delete)*. Dalam melaksanakan penelitian ini dilakukan 6 tahapan antara lain: analisis sistem, identifikasi hardware dan software, penentuan dataset, perancangan tahapan pengujian, Implementasi dan pengujian, serta analisis hasil pengujian yang digambarkan pada gambar berikut:



Gambar 1. Tahapan Penelitian

2.1 Analisis Sistem

Pada tahapan berikut dilakukan analisis terhadap berbagai lingkup sistem untuk pengujian yang akan dilakukan. Untuk mengukur kecepatan respon dari setiap database ini diperlukan berbagai software, hardware serta dataset untuk melakukan pengujian. sehingga harapannya proses pengujian bisa berjalan lancar dan menghasilkan hasil penelitian yang akurat.

2.2 Identifikasi Kebutuhan Hardware dan Software

Dalam tahap ini dilakukan identifikasi perangkat lunak dan perangkat keras yang dimana akan menjadi media pengujian dan beberapa perangkat lunak akan menjadi *database* objek pada pengujian ini.

Tabel 1. Spesifikasi Software yang Digunakan

No.	Software	Versi
1.	Microsoft Windows	10/64 bit OS
2.	MySQL Xampp	PHP 7.4.2
3.	PostgreSQL	9.6
4.	MongoDB	4.4
5.	MongoDB Compass	4.4

Tabel 2. Spesifikasi Hardware yang Digunakan

No.	Hardware	Spesifikasi
1.	Memory	16 GB DDR4

2.	Prosesor	Intel Core i5-10400F (4.30 GHz) DDR4
3.	Graphic	Nvidia GTX 1660 Super GDDR6
4.	Storage	SSD NVME PCIE 512GB

2.3 Penentuan Dataset

Dataset pada studi kasus ini akan menggunakan sebuah dataset yang berisi informasi seputar aplikasi android seperti nama aplikasi, kategori, *rating*, harga, dan sebagainya yang terdiri dari 6 kolom dan 29999 baris data. pada pengujian kali ini tidak menggunakan data yang memiliki relasi dikarenakan database MongoDB merupakan database NoSQL yang tidak mendukung sistem relasi antar tabel. sehingga untuk hasil yang lebih akurat maka hanya digunakan 1 tabel saja pada ketiga database tanpa menggunakan relasi antar tabel pada database MySQL dan PostgreSQL.

2.4 Perancangan Tahapan Pengujian

Pada tahap ini akan menjelaskan tahap-tahap dilakukannya pengujian pada 3 database. dimulai dari pengujian fungsi *select*, *insert*, *update* dan *delete*. Sebelum melakukan pengujian dilakukan tahap instalasi ketiga database pada komputer yang menjadi tempat pengujian. Selanjutnya ketiga database akan dihubungkan ke sebuah aplikasi bernama *DataGrip* yang merupakan aplikasi untuk melakukan manajemen berbagai database hanya dalam satu aplikasi sehingga memudahkan pada saat melakukan pengujian. Selanjutnya setelah semua database terhubung ke *Datagrip* maka dapat dilakukan pengujian.

2.5 Menguji Kueri

Pada penelitian ini dilakukan pengujian kueri yaitu *INSERT (create)*, *SELECT (read)*, *UPDATE (update)*, *DELETE (delete)*. Pengujian kueri dilakukan pada 5 limit data yaitu 250, 2500, 5000, 15000, dan yang terakhir yaitu 25000. Kueri pertama yang digunakan pada penelitian ini yaitu *SELECT* seperti yang terlampir pada Table 1.

Tabel 3. Kueri *SELECT* yang digunakan untuk pengujian.

No.	Database	Kueri
1.	MySQL	SELECT * from android_permission limit 25000;
2.	MongoDB	db.android_permission.find().limit(25000)
3.	PostgreSQL	select * from "Android_Permission" limit 25000;

Kueri kedua yang dilakukan untuk pengujian yaitu kueri *INSERT*. Untuk kueri *INSERT* dilakukan looping untuk melakukan *INSERT* sebanyak 250, 2500, 5000, 15000, dan yang terakhir yaitu 25000 data. Untuk detail dari kueri yang digunakan untuk pengujian terlampir pada Tabel 4.

Kueri ketiga yang dilakukan untuk pengujian yaitu kueri *UPDATE*. Untuk kueri *UPDATE* dilakukan pada data sebanyak 250, 2500, 5000, 15000, dan yang terakhir yaitu 25000 data. Untuk detail dari kueri yang digunakan untuk pengujian terlampir pada Tabel 5.

Kueri keempat yang dilakukan untuk pengujian yaitu kueri *DELETE*. Untuk kueri *DELETE* dilakukan pada data sebanyak 250, 2500, 5000, 15000, dan yang terakhir yaitu 25000 data. Untuk detail dari kueri yang digunakan untuk pengujian terlampir pada Tabel 6.

Tabel 4. Kueri INSERT yang digunakan untuk pengujian.

No.	Database	Kueri
1.	MySQL	<pre> DELIMITER \$\$ CREATE PROCEDURE insertData() BEGIN DECLARE counter INT DEFAULT 1; WHILE counter <= 250 DO INSERT INTO android_permission(App, Package, Category, Description, Rating, Price) VALUES ('a', 'a', 'a', 'a', 0, 0); SET counter = counter + 1; END WHILE; END \$\$ DELIMITER ; call insertData(); </pre>
2.	MongoDB	<pre> for(var x=1; x<=25000; x++){ db.android_permission.insertOne({ App: "a", Category: "a", Description: "aaa", Package: "aaa", Price:5, Rating: 0}) } do \$\$ begin for r in 1..25000 loop insert into "Android_Permission"("App","Package","Category","Description","Rating","Price") values ('a', 'a', 'a', 'a', 0, 0); end loop; end; \$\$; </pre>
3.	PostgreSQL	<pre> do \$\$ begin for r in 1..25000 loop insert into "Android_Permission"("App","Package","Category","Description","Rating","Price") values ('a', 'a', 'a', 'a', 0, 0); end loop; end; \$\$; </pre>

Tabel 5. Kueri UPDATE yang digunakan untuk pengujian.

No.	Database	Kueri
1.	MySQL	<pre> UPDATE android_permission SET App = 'b', Package = 'b', Category = 'b', Description = 'b', Rating = 9, Price = 9 WHERE App='a'; </pre>
2.	MongoDB	<pre> db.android_permission.updateMany({ App: "a" }, {\$set: { App: "b", Category: "b", Description: "bbbb", Package: "bbbb", Price:9, Rating: 9 }}) </pre>
3.	PostgreSQL	<pre> UPDATE "Android_Permission" set "App" = 'b', "Category" = 'b', "Description" = 'bbbb', </pre>

```
"Package" = 'bbbb',           "Rating" = 9,
"Price" = 9
WHERE "App" = 'a';
```

Table 6. Kueri DELETE yang digunakan untuk pengujian.

No.	Database	Kueri
1.	MySQL	DELETE from android_permission where App = 'b';
2.	MongoDB	db.android_permission.deleteMany({ App: "b" })
3.	PostgreSQL	DELETE FROM "Android_Permission" where "App" = 'b';

3. HASIL DAN PEMBAHASAN

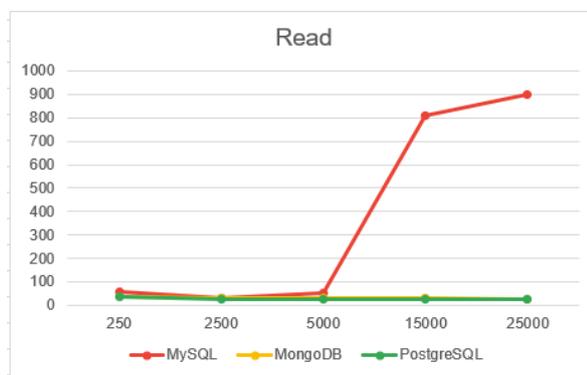
Berdasarkan kueri yang telah disampaikan sebelumnya, berikut akan disampaikan hasil pengujian yang telah dilakukan. Data disajikan dalam bentuk tabel waktu respon kueri serta didukung dengan grafik untuk memudahkan dalam memahami dan mengambil kesimpulan nantinya.

3.1 SELECT

Berikut merupakan hasil pengujian yang telah dilakukan sebelumnya pada kueri *SELECT*. Untuk hasil *response time* dengan satuan *milisecond* (ms) akan dijabarkan dalam sebuah tabel dan grafik perbandingan dari MongoDB, MySQL, dan PostgreSQL.

Tabel 7. Hasil Pengujian SELECT Data

No.	Database	250	2500	5000	15000	25000
1.	MySQL	58ms	31ms	51ms	810ms	900ms
2.	MongoDB	35ms	31ms	29ms	32ms	27ms
3.	PostgreSQL	36ms	25ms	25ms	24ms	26ms



2.

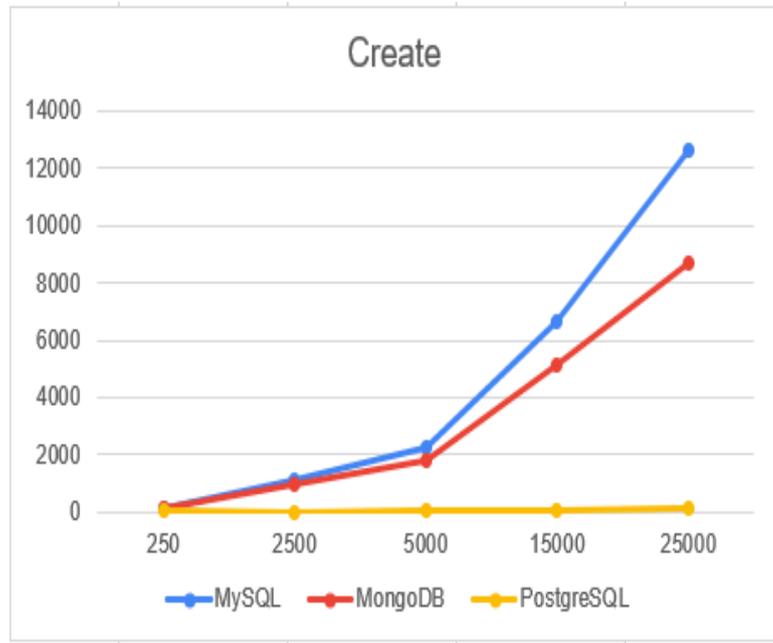
Gambar 2. Grafik Hasil Pengujian SELECT Data

3.2 INSERT

Berikut merupakan hasil pengujian yang telah dilakukan sebelumnya pada kueri *INSERT*. Untuk hasil *response time* dengan satuan *milisecond* (ms) akan dijabarkan dalam sebuah tabel dan grafik perbandingan dari MongoDB, MySQL, dan PostgreSQL.

Tabel 8. Hasil Pengujian *INSERT* Data

No.	Database	250	2500	5000	15000	25000
1.	MySQL	161ms	1s 153ms	2s 250ms	6s 622ms	12s 644ms
2.	MongoDB	146ms	970ms	1s 789ms	5s 161ms	8s 663ms
3.	PostgreSQL	27ms	26ms	62ms	91ms	103ms



3.

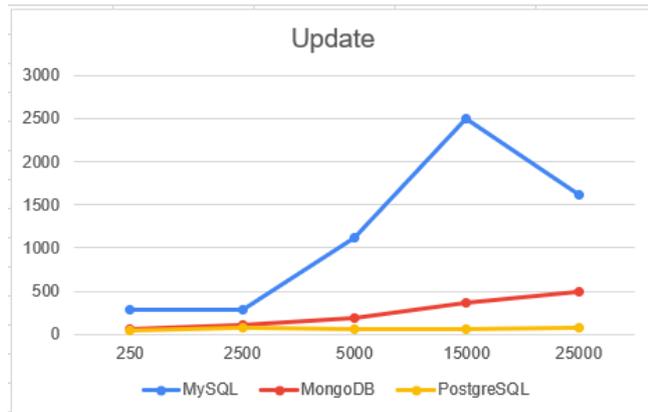
Gambar 3. Grafik Hasil Pengujian *INSERT* Data

3.3. *UPDATE*

Berikut merupakan hasil pengujian yang telah dilakukan sebelumnya pada kueri *UPDATE*. Untuk hasil *response time* dengan satuan *milisecond* (ms) akan dijabarkan dalam sebuah tabel dan grafik perbandingan dari MongoDB, MySQL, dan PostgreSQL.

Tabel 9. Hasil Pengujian *UPDATE* Data

No.	Database	250	2500	5000	15000	25000
1.	MySQL	286ms	277ms	1s 125ms	2s 493ms	1s 621ms
2.	MongoDB	54ms	107ms	184ms	368ms	501ms
3.	PostgreSQL	44ms	70ms	52ms	66ms	74ms



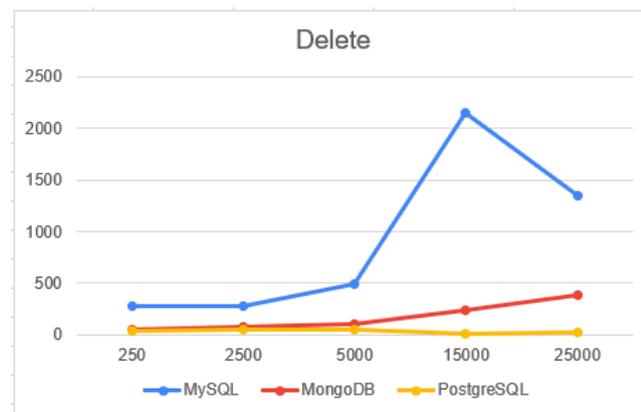
4. Gambar 4. Grafik Hasil Pengujian UPDATE Data

3.4. DELETE

Berikut merupakan hasil pengujian yang telah dilakukan sebelumnya pada kueri *DELETE*. Untuk hasil *response time* dengan satuan *milisecond* (ms) akan dijabarkan dalam sebuah tabel dan grafik perbandingan dari MongoDB, MySQL, dan PostgreSQL.

Tabel 10. Hasil Pengujian DELETE Data

No.	Database	250	2500	5000	15000	25000
1.	MySQL	283ms	282ms	496ms	2s 156ms	1s 346ms
2.	MongoDB	50ms	76ms	106ms	239ms	382ms
3.	PostgreSQL	43ms	46ms	49ms	13ms	20ms



5. Gambar 5. Grafik Hasil Pengujian DELETE Data

Berdasarkan seluruh pengujian yang telah ditampilkan dalam menganalisis perbandingan kinerja kueri *response time* dari MongoDB, MySQL, dan PostgreSQL telah dipetakan hasil pengujian agar lebih mudah dalam memahami hasil penelitian sebagai berikut:

Tabel 11. Hasil Pengujian Total Keseluruhan Response Time

No.	Pengujian	MySQL	MongoDB	PostgreSQL
1.	SELECT	1s 850ms	154ms	136ms
2.	INSERT	22s 830ms	16s 729ms	309ms

3.	UPDATE	5s 802ms	1s 214ms	306ms
4.	DELETE	4s 563ms	853ms	171ms

Berdasarkan tabel hasil pengujian total keseluruhan *response time* maka dapat disimpulkan bahwa *response time* tercepat diungguli oleh PostgreSQL sebagai urutan pertama pada 5 pengujian kueri yaitu *INSERT (create)*, *SELECT (read)*, *UPDATE (update)*, *DELETE (delete)*. Lalu urutan kedua diperoleh oleh MongoDB dan yang terakhir yaitu urutan ketiga diperoleh oleh MySQL pada 5 pengujian kueri yaitu *INSERT (create)*, *SELECT (read)*, *UPDATE (update)*, *DELETE (delete)*.

4. KESIMPULAN DAN SARAN

Dari penelitian ini dapat disimpulkan bahwa basis data DBMS PostgreSQL yang paling unggul kemudian diikuti oleh MongoDB yang merupakan basis data NoSQL dapat dilihat dari kecepatan respon dari setiap *query* yang diujikan, namun jika dibandingkan dengan basis data MySQL yang cenderung lemah pada waktu respon dalam memproses *query* yang memiliki selisih waktu yang sangat signifikan bahkan jika hanya dibandingkan dengan MongoDB. Penelitian ini membuktikan bahwa tidak semua *database* relasional (DBMS) memiliki performa yang lebih buruk dari *database* NoSQL karena PostgreSQL yang merupakan *database* relasional masih mampu bersaing dengan MongoDB yang merupakan *database* NoSQL. Saran untuk penelitian selanjutnya dapat melakukan penelitian yang sama dengan menggunakan *database* yang berbeda dan menggunakan fungsi yang lebih kompleks.

5. DAFTAR RUJUKAN

- [1] G. Kiraz and C. Toğay, "IoT Data Storage: Relational & Non-Relational Database Management Systems Performance Comparison," *34. TBD Natl. Informatics Symp.*, pp. 48–52, 2017.
- [2] K. Christiono1 and Hendi Sama2, "Studi Komparasi Database Management System Antara Maria Db Dan Postgresql Terhadap Efisiensi Penggunaan Sumber Daya Komputer," *Univ. Int. Batam*, vol. 1, no. 1, pp. 573–579, 2020.
- [3] M. S. O.M.I Tavares., Rangkoly S.M., Sarah B. D., Utami Ema., Mustafa, "Analisis Perbandingan Performansi Waktu Respons Kueri," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 303–313, 2020.
- [4] R. Klewka, W. Truskowski, and M. Skublewska-Paszowska, "Comparison of MySQL, MSSQL, PostgreSQL, Oracle databases performance, including virtualization Porównanie wydajności baz danych MySQL, MSSQL, PostgreSQL oraz Oracle z uwzględnieniem wirtualizacji," vol. 16, no. June, pp. 279–284, 2020.
- [5] M. S. Ardian Dwi Praba, "Studi Perbandingan Performansi Antara Mysql Dan Postgresql," *J. Khatulistiwa Inform.*, vol. VIII, no. 2, pp. 1–6, 2020.
- [6] "Database Management System – School of Information Systems." <https://sis.binus.ac.id/2017/12/18/database-management-system/> (accessed Nov. 11, 2021).
- [7] M. Radoev, "A Comparison between Characteristics of NoSQL Databases and Traditional Databases," *Comput. Sci. Inf. Technol.*, vol. 5, no. 5, pp. 149–153, 2017, doi: 10.13189/csit.2017.050501.
- [8] N. Ferdiansyah, D. A. Rahayu, and R. Permala, "Comparison of Postgresql, Mariadb and Mongoddb Capabilities in Processing Lapan Satellite Ais Data," pp. 231–238, 2019, doi: 10.30536/p.siptekgan.2019.v23.23.
- [9] B. M. Klimek and M. Skublewska-paszowska, "Comparison of the performance of relational databases PostgreSQL and MySQL for desktop application

- Porównanie wydajności relacyjnych baz danych PostgreSQL oraz MySQL dla aplikacji desktopowej,” vol. 18, no. January, pp. 61–66, 2021.
- [10] S. Budiman, F. Fadhila, and V. A. Saputro, “Perbandingan Performa SQL dan NoSQL Dengan PHP,” vol. 6, no. 1, pp. 38–42, 2021.
- [11] “SQL vs NoSQL – School of Information Systems.” <https://sis.binus.ac.id/2016/08/23/sql-vs-nosql/> (accessed Nov. 11, 2021).
- [12] V. Ardiyansyah, S. Budiman, and F. Fadhila, “A Analisis Performa Kecepatan MySQL dan NoSQL Pada Sistem Operasi Windows dan Linux,” *Jnanaloka*, pp. 21–26, 2021, doi: 10.36802/jnanaloka.2021.v2-no1-21-26.
- [13] L. G. Wiseso, M. Imrona, and A. Alamsyah, “Analisis Performa Neo4j, MongoDB, dan PostgreSQL sebagai Database Manajemen Big Data Pemilu 2019,” vol. 7, no. 3, pp. 9690–9697, 2020.
- [14] R. Poljak, P. Poscic, and D. Jaksic, “Comparative analysis of the selected relational database management systems,” *2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 - Proc.*, pp. 1496–1500, 2017, doi: 10.23919/MIPRO.2017.7973658.
- [15] J. Kurniasih, E. Utami, and S. Raharjo, “Heuristics and Metaheuristics Approach for Query Optimization Using Genetics and Memetics Algorithm,” *2019 1st Int. Conf. Cybern. Intell. Syst. ICORIS 2019*, vol. 1, no. August, pp. 168–172, 2019, doi: 10.1109/ICORIS.2019.8874909.
- [16] R. Wodyk and M. Skublewska-Paszkowska, “Performance comparison of relational databases SQL Server, MySQL and PostgreSQL using a web application and the Laravel framework,” *J. Comput. Sci. Inst.*, vol. 17, no. October, pp. 358–364, 2020, doi: 10.35784/jcsi.2279.
- [17] R. Pamungkas, “Optimalisasi Query Dalam Basis Data My Sql Menggunakan Index,” *Res. Comput. Inf. Syst. Technol. Manag.*, vol. 1, no. 1, p. 27, 2018, doi: 10.25273/research.v1i1.2453.