PERBANDINGAN PERFORMA RESPON WAKTU KUERI MYSQL, POSTGRESQL, DAN MONGODB

COMPARISON OF QUESTION TIME RESPONSE PERFORMANCE MYSQL, POSTGRESQL, AND MONGODB

Yudha Yunanto Putra[1)], Oktania Purwaningrum[2)], Rivaldo Hadi Winata[3)]
E-mail : [1)]18082010032@student.upnjatim.ac.id , [2)]18082010029@student.upnjatim.ac.id, [2)]18082010026@student.upnjatim.ac.id

[1,2,3]Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional "Veteran" Jawa Timur

Abstrak

Teknologi telah mempengaruhi digitalisasi pada fungsi pekerjaan pada berbagai instansi. Pekerjaan dalam setiap fungsi kerja pada instansi dapat secara mudah dilakukan dengan adanya teknologi. Penggunaan teknologi berupa aplikasi membutuhkan sistem *database* untuk menyimpan data-data dan informasi pada suatu instansi. DBMS pada dasarnya terbagi menjadi 2 jenis yaitu DBMS relasional dan non-relasional. Contoh dari DBMS relasional adalah MySQL dan PostgreSQL, sedangkan DBMS non-relasional adalah MongoDB. Penelitian ini mengukur waktu yang diperlukan setiap DBMS dalam mengeksekusi *query*. Penelitian ini melakukan eksekusi pada berbagai skenario *query*, yaitu *INSERT (create), SELECT (read), UPDATE (update), DELETE (delete), SUM*, dan *COUNT*. *Dataset* yang digunakan dalam penelitian ini berasal dari *open* data website kaggle yaitu data tentang permainan catur yang terdiri dari 16 kolom dan 20.058 baris data. Hasil dari penelitian ini yaitu PostgreSQL yang paling cepat waktu respon eksekusi *query* dibanding DBMS lain.

Kata kunci: *Perbandingan, DBMS, MySQL, PostgreSQL, MongoDB*

Abstract

In this age of technology, it affects the digitization of work functions in various agencies. Work in every work function at the agency can be easily done with the presence of technology. The use of technology in the form of applications requires a database system to store data and information in an agency. DBMS is basically divided into 2 types, namely relational and non-relational DBMS. Examples of relational DBMS are MySQL and PostgreSQL, while non-relational DBMS are MongoDB. This study measures the time required for each DBMS to execute queries. This research executes on various query scenarios, namely INSERT (create), SELECT (read), UPDATE (update), DELETE (delete), SUM, and COUNT. The dataset used in this study comes from the open data website kaggle, namely data about chess games consisting of 16 columns and 20,058 rows of data. The result of this research is that PostgreSQL has the fastest query execution response time compared to other DBMS.

Keyword: *Comparison, DBMS, MySQL, PostgreSQL, MongoDB*

1. PENDAHULUAN

Banyaknya data yang disimpan pada *database* mempengaruhi kecepatan penggunaan aplikasi yang akan diakses [1]. Hal ini dapat memberi pengaruh buruk terhadap kinerja dari instansi [2]. Untuk meningkatkan kinerja dari database dapat dilakukan berbagai cara seperti: menambah kemampuan *hardware* yang digunakan pada sistem seperti memori dan *processor*, optimalisasi *query* yang digunakan, dan cara-cara lainnya [3]. Pemilihan DBMS yang digunakan juga bisa mempengaruhi performa pengolahan data [4]. DBMS atau *Database Management System* merupakan sistem yang

digunakan mengelola suatu *database* secara efisien dan efektif serta dapat menjalankan operasi terhadap data yang diminta oleh banyak pengguna pada suatu sistem [5].

DBMS pada dasarnya terbagi menjadi 2 jenis yaitu DBMS relasional dan non-relasional. DBMS relasional menyimpan data berupa baris dan kolom dalam sebuah tabel dengan konsistensi data yang tinggi [6]. DBMS relasional memiliki konsep *data relations* pada setiap tabel yang memiliki keteraturan struktur, penggunaan yang lebih mudah, dan tingkat presisi yang tinggi [7]. Konsep *data relations* ini dapat mengurangi faktor ambigu pada data [8]. DBMS relasional yang paling sering digunakan contohnya MySQL dan PostgreSQL .

Sedangkan DBMS non-relasional atau NoSQL, tidak bergantung tabel dan skema baku. Baris dan kolom data dapat ditambahkan kapan saja tanpa proses eksklusif [9]. Kumpulan data dapat berubah setelah pengaturan sistem, sehingga memerlukan sistem penyimpanan data fleksibel [10]. Ada empat format penyimpanan berbeda pada NoSQL yaitu *key-value*, *columns*, *document-based*, dan *graff-based* [11]. Pada *Not Only* SQL (NoSQL) dapat menyimpan data dalam berbagai jenis secara efektif. Mulai dari data terstruktur, semi terstruktur, dan tidak terstruktur [12]. Contoh dari NoSQL adalah MongoDB.

Dari berbagai jenis DBMS tersebut, maka harus memperhatikan jenis DBMS mana yang cocok digunakan pada sebuah instansi dan DBMS mana yang paling menampilkan performa yang baik untuk memproses data-data yang dibutuhkan instansi [13]. Terlebih lagi jika jumlah datanya ribuan atau disebut *Big Data* [14]. Pemilihan DBMS ini menjadi hal yang penting karena tidak hanya performa yang baik tetapi juga kecepatan DBMS dalam memproses data [15].

Sehingga tujuan penelitian ini adalah untuk membanding performa DBMS untuk memproses *query* pada *big data*. Penelitian ini mengukur waktu yang diperlukan setiap DBMS dalam mengeksekusi *query*. Hasil dari penelitian ini dapat menunjukkan jenis DBMS mana yang paling cepat dan paling lambat untuk mengeksekusi suatu *query*. Penelitian ini menggunakan DBMS relasional yaitu MySQL dan PostgreSQL, dan untuk DBMS non-relasional yaitu MongoDB. Penelitian ini melakukan eksekusi pada berbagai skenario *query*, yaitu *INSERT (create), SELECT (read), UPDATE (update), DELETE (delete), SUM*, dan *COUNT*. *Query* tersebut dieksekusi dengan berbagai jumlah data yang berbeda. Hasil penelitian ini diharapkan bisa dijadikan acuan untuk suatu instansi dalam memilih DBMS yang cocok dan sesuai dengan kebutuhan dari sistem yang ada.

2. METODOLOGI

Pada dasarnya, ada banyak perbedaan antara *database* relasional MySQL, PostgreSQL dan database non-relasional NoSQL MongoDB. Pengujian dilakukan untuk membantu menemukan perbedaan kinerja pemrosesan data dari kedua *database* [1]. Pada penelitian ini dilakukan pengujian respon waktu dengan berbagai skenario untuk mengetahui respon waktu dari setiap *database*. Setelah melakukan pengujian maka respon waktu akan dicatat, direkap, kemudian dibandingkan. Berikut merupakan tahapan penelitian yang dilakukan.

Gambar 1. Tahapan Penelitian

2.1 Identifikasi Kebutuhan Perangkat

Tahap ini melakukan identifikasi perangkat lunak dan perangkat keras yang dibutuhkan untuk melakukan pengujian sesuai dengan spesifikasi yang dibutuhkan dari MySQL, MongoDB, PostgreeSQL. Tabel 1 merupakan spesifikasi dari *testbed* yang digunakan untuk pengujian.

Tabel 1. Spesifikasi *Software* yang Digunakan

No.	*Software*	Versi
1.	Microsoft Windows	10/64 bit OS
2.	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

2.2 Persiapan Dataset

Dataset penelitian ini kami ambil dari sebuah website *open data* yaitu kaggle.com. Dataset dari penelitian ini akan menggunakan dataset yang berisi informasi tentang permainan catur yang terdiri dari *black_id, black_rating, created_at, id, increment_code, last_move_at, moves, opening_eco, opening_name, opening_ply,rated, turns, victory_status, white_id, white_rating, winner* yang terdiri dari 16 kolom dan 20.058 baris data. Pengujian pada penelitian ini tidak memanfaatkan data yang memiliki relasi dikarenakan MongoDB merupakan *database* NoSQL sehingga tidak memiliki relasi. Sehingga untuk menyamakan pengujian serta untuk mencapai hasil yang lebih akurat dan relevan maka penelitian ini hanya menggunakan 1 tabel saja untuk ketiga *database.*

2.3 Perancangan Pengujian

Tahap ini akan menjelaskan langkah pengujian pada setiap *database*. Sebelum melakukan pengujian dilakukan tahap instalasi ketiga database pada *testbed* yang menjadi tempat pengujian. Selanjutnya ketiga database akan dihubungkan ke *DataGrip* yang merupakan sebuah aplikasi untuk memanajemen *database.* Maka, dengan adanya *DataGrip* pengujian akan lebih mudah dan ringkas. Selanjutnya setelah semua database terhubung ke *Datagrip* maka dapat dilakukan pengujian dimulai dari pengujian fungsi *select, insert, update* dan *delete.*

Pengujian kueri dilakukan pada 5 limit data yang berbeda yaitu 250, 2500, 5000, 15000, dan yang terakhir yaitu 20000. Pada penelitian ini dilakukan pengujian kueri yaitu *INSERT (create), SELECT (read), UPDATE (update), DELETE (delete), SUM,* dan *COUNT.* 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 games LIMIT 20000;`
2.	MongoDB	`db.games.find().limit(20000);`
3.	PostgreSQL	`SELECT * FROM games LIMIT 20000;`

Kueri kedua yang dieksekusi untuk diuji adalah kueri *INSERT*. Untuk kueri *INSERT* dilakukan menggunakan perulangan karena pada MongoDB *INSERT* pada lebih dari satu baris harus menggunakan perulangan sehingga MySQL dan PostgreeSQL menyesuaikan dengan menggunakan perulangan. Kueri ketiga yang dieksekusi untuk diuji adalah kueri *UPDATE*. Rincian kedua kueri yang digunakan untuk pengujian terlampir pada Tabel 4.

Tabel 4. Kueri *INSERT* yang digunakan untuk pengujian.

No	Database	Kueri Kedua *(INSERT)*	Kueri Ketiga *(UPDATE)*

1	MySQL	`DELIMITER $$`
		`CREATE PROCEDURE insertData()`
		`BEGIN`
		`DECLARE counter INT DEFAULT 1;`
		`WHILE counter <= 250 DO`
		`INSERT INTO games(`
		`id,`
		`rated,`
		`created_at,`
		`last_move_at,`
		`turns,`
		`victory_status,`
		`winner,`
		`increment_code,`
		`white_id,`
		`white_rating,`
		`black_id,`
		`black_rating,`
		`moves,`
		`opening_eco,`
		`opening_name,`
		`opening_ply)`
		`VALUES('2', 'a', 123, 123, 123, '5', 'abc', 'abc', '0', 0, 'a', 2, 'abc', 'asas', 'ajdaj', 1);`
		`SET counter = counter + 1;`
		`END WHILE;`
		`END$$`
		`DELIMITER ;`
		`call insertData();`
			`UPDATE games set`
			`id='2',`
			`rated= 'sdsda',`
			`created_at = 1235,`
			`last_move_at= 1235,`
			`turns= 1235,`
			`victory_status= 'dsds5',`
			`winner= 'abdsdc',`
			`increment_code= 'abdsdsc',`
			`white_id= 'vsvs0',`
			`white_rating= 11,`
			`black_id = 'dsdsa',`
			`black_rating= 21,`
			`moves= 'abcsfsf',`
			`opening_eco= 'afsfssas',`
			`opening_name= 'ajdfsfsaj',`
			`opening_ply= 12`
			`WHERE id = '2';`

2	MongoDB	`for(var x=1; x<=20000; x++){`
		`db.games.insertOne(`
		`{ black_id: "a",`
		`black_rating: 2,`
		`created_at: "123",`
		`id:"2",`
		`increment_code: "abc",`
		`last_move_at: "asas",`
		`moves: "abc",`
		`opening_eco: "asas",`
		`opening_name: "ajdaj",`
		`opening_ply: 1,`
		`rated: "a",`
		`turns: 123,`
		`victory_status:"5",`
		`white_id: "0",`
		`white_rating: 0,`
		`winner: "abc"}`
		`)`
		`}`
			`db.games.updateMany({id: "a"},`
			`{$set: {`
			`black_id: "a",`
			`black_rating: 2,`
			`created_at: "123",`
			`id:"2",`
			`increment_code: "abc",`
			`last_move_at: "asas",`
			`moves: "abc",`
			`opening_eco: "asas",`
			`opening_name: "ajdaj",`
			`opening_ply: 1,`
			`rated: "a",`
			`turns: 123,`
			`victory_status:"5",`
			`white_id: "0",`
			`white_rating: 0,`
			`winner: "abc"`
			`}})`

3	PostgreSQL	`do $$`
		`begin`
		`for r in 1..20000 loop`
		`insert into "games"(`
		`"id",`
		`"rated",`
		`"created_at",`
		`"last_move_at",`
		`"turns",`
		`"victory_status",`
		`"winner",`
		`"increment_code",`
		`"white_id",`
		`"white_rating",`
		`"black_id",`
		`"black_rating",`
		`"moves",`
		`"opening_eco",`
		`"opening_name",`
		`"opening_ply"`
			`UPDATE "games" set`
			`"id"='2',`
			`"rated"= 'asas',`
			`"created_at" = 123,`
			`"last_move_at"= 123,`
			`"turns"= 123,`
			`"victory_status"= '5',`
			`"winner"= 'abc',`
			`"increment_code"= 'abc',`
			`"white_id"= '0',`
			`"white_rating"= 0,`
			`"black_id" = 'a',`
			`"black_rating"= 2,`
			`"moves"= 'abc',`
			`"opening_eco"= 'asas',`
			`"opening_name"= 'ajdaj',`

No	Database	Kueri Kedua *(INSERT)*	Kueri Ketiga *(UPDATE)*
		) values ('2', 'a', 123, 123, 123, '5', 'abc', 'abc', '0', 0, 'a', 2, 'abc', 'asas', 'ajdaj', 1); end loop; end; $$;	"opening_ply"= 1 WHERE "id" = '2';

Kueri keempat yang dieksekusi untuk diuji adalah kueri *DELETE*. Rincian kueri yang digunakan untuk pengujian terlampir pada Tabel 5.

Tabel 5. Kueri *DELETE* yang digunakan untuk pengujian.

No.	Database	Kueri
1.	MySQL	`DELETE FROM games where id = '2';`
2.	MongoDB	`db.games.deleteMany( { id: "a"} )`
3.	PostgreSQL	`DELETE FROM "games" where "id" = '2'`

Kueri kelima yang dieksekusi untuk diuji adalah kueri *SELECT SUM()*. Pengujian kueri *SELECT SUM()* tidak diulang seperti kueri yang lain dikarenakan dalam melakukan *SELECT SUM()* tidak dapat dilimit. Sehingga untuk *SELECT SUM()* akan dilakukan pada satu kolom yaitu winner. Rincian kueri yang digunakan untuk pengujian terlampir pada Tabel 6.

Tabel 6. Kueri *SELECT SUM()* yang digunakan untuk pengujian.

No.	Database	Kueri
1.	MySQL	`SELECT winner, SUM(opening_ply) FROM games GROUP BY winner;`
2.	MongoDB	`db.games.aggregate([{ $group: { _id: "$winner", "TotalGiliran": { $sum: "$opening_ply" } } }]);`
3.	PostgreSQL	`SELECT winner, SUM(opening_ply) FROM games GROUP BY winner;`

Kueri keenam yang dieksekusi untuk diuji adalah kueri *SELECT COUNT()*. Pengujian kueri *SELECT COUNT()* tidak diulang seperti kueri yang lain dikarenakan dalam melakukan *SELECT COUNT()* tidak dapat dilimit. Sehingga untuk *SELECT COUNT()* akan dilakukan pada satu kolom yaitu winner. Rincian kueri yang digunakan untuk pengujian terlampir pada Tabel 7.

Tabel 7. Kueri *SELECT COUNT()* yang digunakan untuk pengujian.

No.	Database	Kueri
1.	MySQL	`SELECT winner, COUNT(*) FROM games GROUP BY winner;`
2.	MongoDB	`db.games.aggregate([ {"$group" : {_id:"$winner", count:{$sum:1}}} ]);`

No.	Database	Kueri
3.	PostgreSQL	`SELECT winner, COUNT(*) FROM games GROUP BY winner ;`

3. HASIL DAN PEMBAHASAN

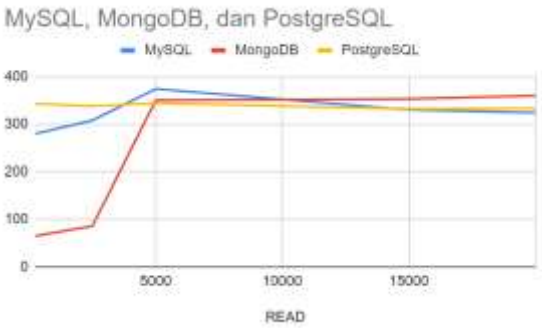
Berdasarkan kueri yang telah ditetapkan pada tahap sebelumnya, berikut dilampirkan hasil pengujian dari *testbed* yang telah ditetapkan. Data performa respon waktu disusun dalam suatu tabel dan juga dibuat grafik dari tabel.

3.1 *SELECT*

Tabel 8 dan gambar 2 adalah hasil pengujian pada kueri *SELECT*. Untuk hasil *response time* menggunakan satuan *milisecond* (ms) dan data dari MongoDB, MySQL, dan PostgreSQL akan dilampirkan dengan 5 limit data yaitu 250, 2500, 5000, 15000, 20000.

Tabel 8. Hasil Pengujian *SELECT* Data

No.	Database	250	2500	5000	15000	20000
1.	MySQL	280ms	308ms	374ms	330ms	324ms
2.	MongoDB	65ms	86ms	350ms	353ms	360ms
3.	PostgreSQL	343ms	338ms	344ms	332ms	333ms



Gambar 2. Grafik Hasil Pengujuan *SELECT* Data

3.2 *INSERT*

Tabel 9 dan gambar 3 adalah hasil pengujian pada kueri *INSERT*. Untuk hasil *response time* menggunakan satuan *milisecond* (ms) dan data dari MongoDB, MySQL, dan PostgreSQL akan dilampirkan dengan 5 limit data yaitu 250, 2500, 5000, 15000, 20000.

Tabel 9. Hasil Pengujian *INSERT* Data

No.	Database	250	2500	5000	15000	20000
1.	MySQL	112ms	155ms	2s 259ms	6s 718ms	8s 915ms
2.	MongoDB	613ms	1s 719ms	3s 622ms	8s 358ms	10s 250ms
3.	PostgreSQL	69ms	21ms	67ms	112ms	132ms

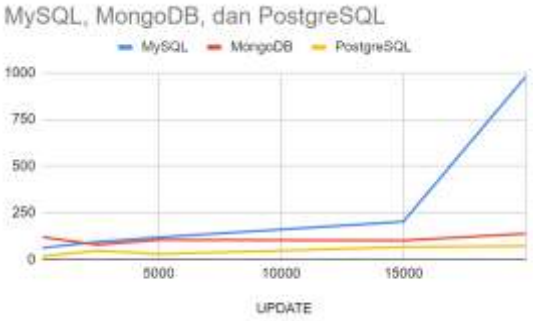
Gambar 3. Grafik Hasil Pengujuan *INSERT* Data

3.3 *UPDATE*

Tabel 10 dan gambar 4merupakan 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 10. Hasil Pengujian *UPDATE* Data

No.	Database	250	2500	5000	15000	20000
1.	MySQL	62ms	94ms	118ms	202ms	980ms
2.	MongoDB	120ms	78ms	105ms	102ms	138ms
3.	PostgreSQL	17ms	47ms	29ms	66ms	72ms



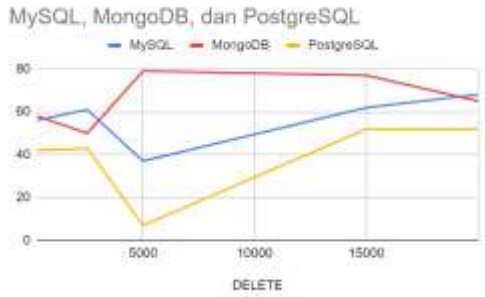
Gambar 4. Grafik Hasil Pengujuan *UPDATE* Data

3.4 *DELETE*

Tabel 11 dan gambar 5 merupakan hasil pengujian yang telah dilakukan sebelumnya pada kueri *DE:ETE*. Untuk hasil *response time* dengan satuan *milisecond* (ms) akan dijabarkan dalam sebuah tabel dan grafik perbandingan dari MongoDB, MySQL, dan PostgreSQL.

Tabel 11. Hasil Pengujian *DELETE* Data

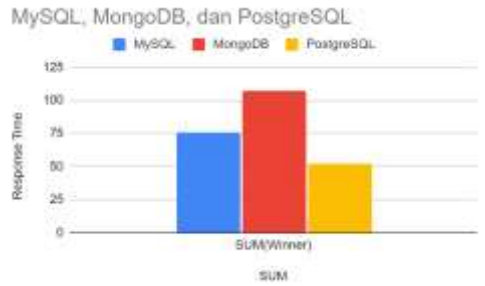
No.	Database	250	2500	5000	15000	20000
1.	MySQL	56ms	61ms	37ms	62ms	68ms
2.	MongoDB	58ms	50ms	79ms	77ms	65ms
3.	PostgreSQL	42ms	43ms	7ms	52ms	52ms



Gambar 5. Grafik Hasil Pengujuan *DELETE* Data

3.5 *SELECT SUM()*

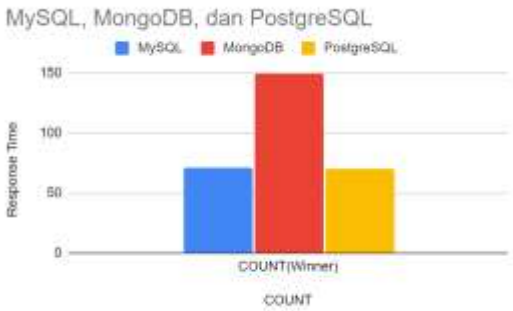
Gambar 6 merupakan hasil pengujian yang telah dilakukan sebelumnya pada kueri *SELECT SUM()*. Untuk hasil *response time* dengan satuan *milisecond* (ms). MySQL memperoleh 76ms, MongoDB 107ms, PostgreSQL 52ms dan akan dilengkapi dengan sebuah grafik perbandingan dari setiap *database*.



Gambar 6. Grafik Hasil Pengujuan *SELECT SUM()* Data

3.6 *SELECT COUNT()*

Gambar 7 merupakan hasil pengujian yang telah dilakukan sebelumnya pada kueri *SELECT COUNT()*. Untuk hasil *response time* dengan satuan *milisecond* (ms). MySQL memperoleh 71ms, MongoDB 149ms, PostgreSQL 70ms dan akan dilengkapi dengan sebuah grafik perbandingan dari setiap *database*.



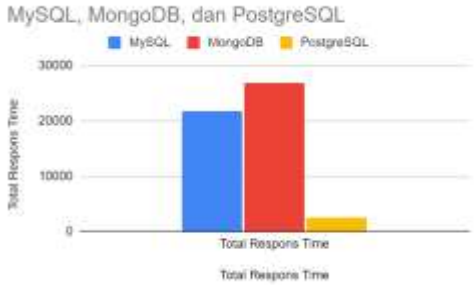
Gambar 7. Grafik Hasil Pengujuan *SELECT COUNT()* Data

Setelah dilakukan pengujian dari setiap kueri pada *testbed*, untuk mempermudah dalam analisa maka *response time* dari perulangan akan dipetakan dengan cara dijumlahkan setiap hasilnya sehingga ditemukanlah hasil akhir dari performa respon waktu *database* seperti telampir pada tabel 11 dan grafik pada gambar 8.

Tabel 12. Hasil Pengujian Total Keseluruhan *Response Time*

No.	Pengujian	MySQL	MongoDB	PostgreSQL
1.	SELECT	1s 616ms	1s 214ms	1s 609ms

No.	Pengujian	MySQL	MongoDB	PostgreSQL
2.	INSERT	18s 159ms	24s 562ms	401ms
3.	UPDATE	1s 456ms	543ms	231ms
4.	DELETE	284ms	329ms	196ms
5.	SELECT SUM()	76ms	107ms	52ms
6.	SELECT COUNT()	71ms	149ms	70ms
	Total *Response Time*	**21s 662ms**	**26s 904ms**	**2s 559ms**



Gambar 8. Grafik Pengujian Total Keseluruhan *Response Time*

Berdasarkan tabel dan grafik hasil pengujian total keseluruhan *response time* maka dapat disimpulkan bahwa *response time* tercepat diungguli oleh PostgreSQL sebagai urutan pertama pada Total *Response Time* sebesar 2s 559ms. Lalu urutan kedua diperoleh oleh MySQL dengan Total *Response Time* sebesar 26s 904ms. Lalu yang terakhir yaitu urutan ketiga diperoleh oleh MongoDB dengan Total *Response Time* sebesar 21s 662ms.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian diatas dapat disimpulkan bahwa basis data PostgreSQL yang paling unggul. Kemudian Diikuti oleh MySQL dan MongoDB. MySQL dan PostgreSQL sendiri merupakan DBMS. Sedangkan MongoDB merupakan NoSQL. Penelitian ini membuktikan bahwa stigma yang mengatakan bahwa performa dari NoSQL lebih cepat dibandingkan dengan DBMS tidak selamanya benar. Terbukti dengan menang telaknya PostgreSQL yang merupakan DBMS terhadap MongoDB pada beberapa skenario pengujian. Akan tetapi, disini MongoDB pada beberapa skenario dapat mengungguli MySQL meskipun pada Total *Response Time* MySQL dapat mengalahkan MongoDB. Semuanya tergantung dari skenario penggunaan seperti apa yang akan dilalui oleh basis data tersebut. Saran untuk penelitian lebih lanjut yaitu dapat menggunakan basis data yang berbeda dan menggunakan kueri yang lebih banyak dan kompleks lagi.

5. DAFTAR RUJUKAN

[1] M. Silalahi, "Perbandingan Performansi Database Mongodb Dan Mysql Dalam Aplikasi File Multimedia Berbasis Web," *Comput. Based Inf. Syst. J.*, vol. 6, no. 1, p. 63, 2018, doi: 10.33884/cbis.v6i1.574.

[2] 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.

[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] J. Y. Seo, D. W. Lee, and H. M. Lee, "Performance comparison of CRUD

operations in IoT based big data computing," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 5, pp. 1765–1770, 2017, doi: 10.18517/ijaseit.7.5.2674.

[5] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Min. Anal.*, vol. 3, no. 2, pp. 85–101, 2020, doi: 10.26599/BDMA.2019.9020015.

[6] R. Panche, B. Ilijoski, and B. Tojtovska, "Comparing Databases for Inserting and Querying," *Fac. Comput. Sci. Eng.*, pp. 7–8, 2019.

[7] B. M. Klimek and M. Skublewska-paszkowska, "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.

[8] C. Martinez-Millana, A. Martinez-Millana, C. Fernandez-Llatas, B. V. Martinez, and V. T. Salcedo, "Comparing data base engines for building big data analytics in obesity detection," *Proc. - IEEE Symp. Comput. Med. Syst.*, vol. 2019-June, pp. 208–211, 2019, doi: 10.1109/CBMS.2019.00050.

[9] 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.

[10] S. Patel, S. Kumar, S. Katiyar, R. Shanmugam, and R. Chaudhary, "EasyChair Preprint MONGODB VS MYSQL: A Comparative Study of MongoDB and MySQL Based on Their Performance," 2020.

[11] M. M. Patil, A. Hanni, C. H. Tejeshwar, and P. Patil, "A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing-Sharding in MongoDB and its advantages," *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, pp. 325–330, 2017, doi: 10.1109/I-SMAC.2017.8058365.

[12] S. Chakraborty, S. Paul, and K. M. Azharul Hasan, "Performance Comparison for Data Retrieval from NoSQL and SQL Databases: A Case Study for COVID-19 Genome Sequence Dataset," *ICREST 2021 - 2nd Int. Conf. Robot. Electr. Signal Process. Tech.*, pp. 324–328, 2021, doi: 10.1109/ICREST51555.2021.9331044.

[13] 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.

[14] M. M. Eyada, W. Saber, M. M. El Genidy, and F. Amer, "Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments," *IEEE Access*, vol. 8, pp. 110656–110668, 2020, doi: 10.1109/ACCESS.2020.3002164.

[15] W. Khan, W. Ahmad, B. Luo, and E. Ahmed, "SQL database with physical database tuning technique and NoSQL graph database comparisons," *Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2019*, no. Itnec, pp. 110–116, 2019, doi: 10.1109/ITNEC.2019.8729264.