SAINSTECH
Jurnal Penelitian dan Pengkajian Sains dan Teknologi
Volume 35 No.4
Desember 2025
SINTA 5

Perbandingan Kinerja MongoDB dan MySQL pada Aplikasi dengan Beban Data Besar

Desta Adyangga Saputra[1], Mohammad Adi Farich[2], Muhammad Naafi'an Anugerah[3], Imam Prayogo Pujiono[4], Dicky Anggriawan Nugroho[5]

[1,2,3,4,5] Program Studi Informatika, Fakultas Ekonomi dan Bisnis Islam ,UIN K.H.Abdurrahman Wahid Pekalongan
Jl. Kusma Bangsa No.9 Pekalongan, Jawa Tengah, Indonesia 51141
e-mail: [1]desta.adyangga.saputra24082@mhs.uingusdur.ac.id,
[2]mohamad.adi.farich24049@mhs.uingusdur.ac.id,
[3]muhammad.naafian.anugerah24070@mhs.uingusdur.ac.id, [4]imam.prayogopujiono@uingusdur.ac.id,
[5]dicky.anggriawannugroho@uingusdur.ac.id,

Abstrak

Penelitian ini bertujuan untuk membandingkan kinerja dua sistem basis data populer, yaitu MySQL sebagai database relasional dan MongoDB sebagai database NoSQL berbasis dokumen, ketika diberikan beban data dalam jumlah besar. Pengujian dilakukan menggunakan metode eksperimen kuantitatif dengan tiga skala dataset, yaitu 1.000, 10.000, dan 100.000 data, yang dibuat secara otomatis menggunakan Faker.js untuk memastikan variasi dan konsistensi data uji. Empat operasi dasar basis data (INSERT, READ, UPDATE, dan DELETE)dilakukan pada kedua sistem untuk mengukur waktu eksekusi dan menilai efisiensi performa masing-masing. Hasil penelitian menunjukkan bahwa MongoDB secara konsisten memiliki waktu eksekusi lebih cepat dibandingkan MySQL, terutama pada operasi insert dan update, berkat arsitektur dokumen dan kemampuan batch processing yang mengurangi overhead pemrosesan. Sementara itu, MySQL mengalami peningkatan waktu eksekusi yang cukup signifikan seiring bertambahnya jumlah data, terutama karena validasi relasional dan aturan ACID yang harus dipenuhi. Temuan ini mengindikasikan bahwa MongoDB lebih sesuai untuk aplikasi berskala besar yang membutuhkan fleksibilitas data dan kecepatan pemrosesan tinggi, sedangkan MySQL tetap ideal untuk aplikasi yang menekankan konsistensi dan integritas data. Penelitian ini diharapkan dapat menjadi referensi dalam pemilihan basis data sesuai kebutuhan aplikasi.

Kata kunci: MySQL, MongoDB, database, NoSQL, FakerJS

Abstract

This study aims to compare the performance of two popular database systems, namely MySQL as a relational database and MongoDB as a document-based NoSQL database, when given a large amount of data load. The testing was conducted using a quantitative experimental method with three dataset scales, namely 1.000, 10.000, and 100.000 data, which were automatically generated using Faker.js to ensure the variation and consistency of the test data. Four basic database operations (INSERT, READ, UPDATE, and DELETE) were performed on both systems to measure execution time and assess the efficiency of each system's performance. The results showed that MongoDB consistently had faster execution times than MySQL, especially for insert and update operations, thanks to its document architecture and batch processing capabilities, which reduced processing overhead. Meanwhile, MySQL experienced a significant increase in execution time as the amount of data increased, mainly due to relational validation and ACID rules that must be met. These findings indicate that MongoDB is more suitable for large-scale applications that require data flexibility and high processing speed, while MySQL remains ideal for applications that emphasize data consistency and integrity. This research is expected to serve as a reference in selecting a database according to application needs.

Keywords: *MySQL, MongoDB, database, NoSQL, FakerJS*

DOI : https://10.37277/stch.v35i4.2543

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi mendorong pertumbuhan data dalam jumlah sangat besar, khususnya pada aplikasi berbasis web, mobile, dan komputasi awan. Fenomena ini sering dikaitkan dengan era *big data*, yang menuntut sistem basis data mampu menangani data dengan volume, kecepatan, dan variasi tinggi secara efisien. Chen et al. (2014) menjelaskan bahwa karakteristik *big data* menimbulkan tantangan baru dalam pengelolaan data skala besar. Gandomi dan Haider (2015) menegaskan bahwa peningkatan volume data harus diimbangi dengan sistem pengolahan yang efisien. Penelitian terbaru oleh Hanine et al. (2020) menunjukkan bahwa performa sistem basis data menjadi faktor kunci dalam aplikasi berbasis data besar.

Basis data relasional seperti MySQL telah lama digunakan dalam berbagai sistem informasi karena kemampuannya menjaga konsistensi dan integritas data melalui penerapan prinsip ACID. Elmasri dan Navathe (2016) menyatakan bahwa basis data relasional sangat efektif untuk pengelolaan data terstruktur. Silberschatz et al. (2020) menambahkan bahwa SQL menyediakan mekanisme kueri yang kuat dan terstandarisasi. Namun demikian, Stonebraker (2018) mengemukakan bahwa pendekatan relasional memiliki keterbatasan ketika dihadapkan pada kebutuhan skalabilitas horizontal dan beban data yang sangat besar.

Sebagai alternatif, basis data NoSQL seperti MongoDB menawarkan model penyimpanan berbasis dokumen yang lebih fleksibel. Sadalage dan Fowler (2017) menjelaskan bahwa NoSQL dirancang untuk menangani kebutuhan skema dinamis. Li et al. (2021) menunjukkan bahwa MongoDB banyak digunakan pada aplikasi *data-intensive* karena efisiensinya dalam menangani operasi tulis berskala besar. Han et al. (2011) juga menekankan bahwa basis data NoSQL cocok untuk lingkungan terdistribusi dengan kebutuhan skalabilitas tinggi.

Berbagai penelitian telah membahas perbandingan kinerja antara basis data SQL dan NoSQL. Nayak et al. (2013) melaporkan bahwa NoSQL cenderung unggul pada operasi tulis, sementara basis data relasional lebih stabil pada kueri kompleks. Abramova dan Bernardino (2016) menemukan perbedaan signifikan pada performa MongoDB dan basis data relasional tergantung jenis beban kerja. Wang et al. (2016) menekankan pentingnya metodologi benchmark yang konsisten agar hasil pengujian dapat dibandingkan secara objektif. Selanjutnya, Zhang et al. (2018) menunjukkan bahwa perbedaan performa sering kali dipengaruhi oleh desain eksperimen yang digunakan.

Berdasarkan temuan tersebut, diperlukan penelitian yang membandingkan kinerja MySQL dan MongoDB dengan pendekatan benchmark yang adil dan terkontrol. Tudorica dan Bucur (2019) menegaskan bahwa perlakuan pengujian yang tidak setara dapat menyebabkan kesimpulan yang bias. Zhang et al. (2021) merekomendasikan penggunaan replikasi eksperimen untuk meningkatkan reliabilitas hasil. Oleh karena itu, penelitian ini berfokus pada pengujian operasi CRUD menggunakan dataset berskala besar, perlakuan yang setara, dan replikasi pengujian, sehingga diharapkan mampu memberikan gambaran objektif mengenai karakteristik performa MySQL dan MongoDB.

2. Metode Penelitian

Penelitian ini menggunakan metode eksperimen kuantitatif yang bertujuan untuk membandingkan kinerja dua sistem basis data, yaitu MySQL dan MongoDB, ketika diberikan beban data dalam jumlah besar. Metode ini dipilih karena proses pengujian performa membutuhkan pengukuran yang terukur dan dapat direplikasi, seperti waktu eksekusi operasi dan penggunaan sumber daya sistem selama proses berlangsung. Penelitian dimulai dengan tahap perancangan lingkungan pengujian, yaitu menyiapkan perangkat keras dan perangkat lunak yang sama untuk kedua basis data agar

DOI : https://10.37277/stch.v35i4.2543

hasil yang diperoleh objektif dan tidak dipengaruhi faktor luar. Setelah itu, dilakukan pembuatan dataset sintetik yang berisi data dalam jumlah besar, yang bertujuan untuk mensimulasikan kondisi aplikasi nyata yang memproses banyak data sekaligus.

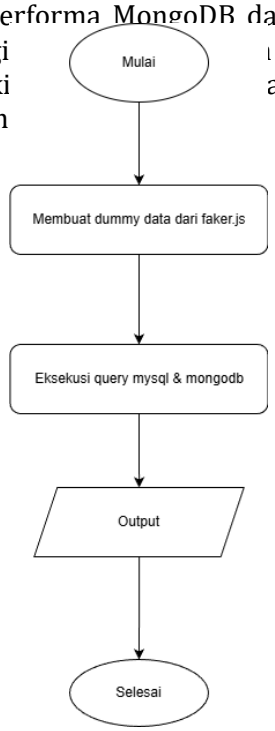
```
// ----------------------------
function generateDummyData() {
    return {
        name: faker.person.fullName(),
        email: faker.internet.email(),
        address: faker.location.streetAddress(),
        created_at: new Date(),
    };
}

// ----------------------------
```

Gambar 1. Function untuk membuat dummy data Faker.js

Dalam proses pengujian performa basis data, penelitian ini menggunakan *dummy data* yang dihasilkan secara otomatis menggunakan **Faker.js**. Faker.js merupakan library JavaScript yang dapat menghasilkan data palsu (*fake data*) dengan struktur realistis, seperti nama, alamat, nomor telepon, email, hingga data transaksi. Penggunaan Faker.js mempermudah proses pembuatan dataset berukuran besar tanpa perlu memasukkan data secara manual. Selain itu, data yang dihasilkan memiliki pola variasi yang cukup baik sehingga dapat mensimulasikan kondisi nyata pada aplikasi berskala besar. Dengan memanfaatkan Faker.js, proses benchmarking pada MongoDB dan MySQL dapat dilakukan lebih konsisten, karena setiap data yang dihasilkan memiliki format seragam namun tetap acak, sehingga membantu peneliti memperoleh hasil pengukuran yang lebih objektif dan terukur.

Untuk memperoleh hasil pengujian yang lebih komprehensif, penelitian ini melakukan pengukuran performa pada tiga skala data yang berbeda, yaitu **1.000**, **10.000**, dan **100.000** dataset. Pemilihan jumlah tersebut bertujuan untuk melihat bagaimana perubahan beban data mempengaruhi kinerja masing-masing sistem basis data. Pengujian pada 1.000 dataset digunakan sebagai tolok ukur awal untuk mengamati kinerja dasar dengan beban ringan. Selanjutnya, pengujian pada 10.000 dataset dilakukan untuk menilai performa pada beban menengah yang umumnya ditemui pada aplikasi skala kecil hingga menengah. Sementara itu, pengujian pada 100.000 dataset digunakan untuk mensimulasikan skenario beban besar yang sering terjadi pada aplikasi modern dengan volume pengguna dan data yang tinggi. Dengan membandingkan hasil dari ketiga skala tersebut, penelitian ini dapat menilai konsistensi performa MongoDB dan MySQL serta mengi l terjadi penurunan ki an ketika jumlah data m



Gambar 2. Diagram Alur Proses Pengujian

Berdasarkan diagram alur pada gambar 2 tahapan proses benchmarking yang dilakukan dalam penelitian ini. Proses dimulai dari tahap *Mulai*, yaitu ketika seluruh kebutuhan seperti lingkungan pengujian, koneksi database, dan library pendukung telah dipersiapkan. Selanjutnya, sistem masuk ke tahap pembuatan data uji menggunakan Faker.js, di mana data dummy dihasilkan secara otomatis untuk mensimulasikan kondisi nyata dengan ukuran data yang bervariasi. Data yang telah dibuat kemudian digunakan pada tahap eksekusi query terhadap MySQL dan

DOI : https://10.37277/stch.v35i4.2543

MongoDB, mencakup operasi utama seperti *insert*, *read*, *update*, dan *delete* untuk mengukur performa masing-masing database. Setelah seluruh operasi dijalankan, sistem menghasilkan *Output* berupa catatan waktu eksekusi yang selanjutnya dianalisis untuk membandingkan kinerja kedua database. Proses kemudian ditutup pada tahap *Selesai*, yang menandakan bahwa seluruh rangkaian pengujian telah selesai dilaksanakan.

3. Hasil dan Pembahasan

Penelitian ini dilakukan menggunakan bahasa pemrograman NodeJs dengan Visual Studio Code (VSCode) sebagai Integrated Development Environment (IDE) dan ekstensi Code Runner untuk menjalankan kode program. Compiler yang digunakan adalah NodeJs untuk memastikan kompatibilitas eksekusi dan dokumentasi hasil pengujian dilakukan menggunakan Snipping Tool untuk mengambil screenshot. Spesifikasi perangkat yang digunakan adalah:

1. Sistem Operasi: Windows 11 Pro
2. RAM: 16 GB
3. CPU: Intel Core i3-12100F (@3.30 GHZ)
4. Penyimpanan: SSD 512 GB

Gambar 3. Kode Function Query dan Pengujian Data Menggunakan Mysql

Gambar 4. Kode Function Query dan Pengujian Data Menggunakan Mongodb

Pada gambar 3 dan 4 di atas merupakan kode yang digunakan untuk melakukan pengujian performa (benchmark) operasi dasar database (INSERT, READ, UPDATE, dan DELETE) dengan membandingkan MySQL dan MongoDB menggunakan data dummy yang dihasilkan melalui fungsi generateDummyData(). Pada kode MySQL, proses dimulai dengan membuat koneksi ke server, kemudian memastikan tabel users sudah tersedia sebelum memasukkan data secara iteratif menggunakan INSERT satu per satu. Setelah itu, kode melakukan pengujian read dengan mengambil beberapa data, update dengan mengubah nama pada data tertentu, dan delete untuk menghapus data berdasarkan kondisi tertentu. Sementara itu, pada kode MongoDB, koneksi dibuka melalui connectMongo(), lalu koleksi users dikosongkan terlebih dahulu sebelum proses insert dilakukan secara **batch** per 1.000 data untuk meningkatkan efisiensi. Pengujian read dilakukan dengan mengambil 1.000 dokumen menggunakan find().limit(), kemudian update dijalankan secara massal dengan updateMany, dan delete dilakukan melalui deleteMany untuk menghapus seluruh dokumen. Kedua kode tersebut menggunakan console.time() dan

DOI : https://10.37277/stch.v35i4.2543

console.timeEnd() untuk mengukur durasi eksekusi setiap operasi, sehingga hasil benchmark dapat dibandingkan secara objektif antara database relasional (MySQL) dan database non-relasional (MongoDB).

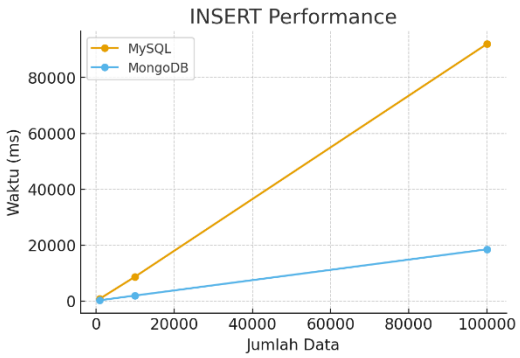
3.1 INSERT

Pada operasi insert, MongoDB secara konsisten unggul di seluruh ukuran dataset. Hal ini terjadi karena MongoDB menggunakan *batch insert* yang mampu memasukkan ribuan dokumen sekaligus dalam satu operasi. Sebaliknya, MySQL harus memproses tiap baris secara individual atau melalui query multivalue yang tetap memiliki overhead besar karena validasi struktur relasional. Ketika jumlah data meningkat dari 1.000 ke 100.000, lonjakan waktu eksekusi MySQL meningkat secara drastis, sedangkan MongoDB tetap lebih stabil. Ini menunjukkan bahwa MongoDB lebih cocok untuk aplikasi yang memerlukan pemrosesan input data dalam jumlah besar secara cepat.

Tabel 1. Hasil pengujian insert

Dataset	MySQL	MongoDB
1.000	820 ms	260 ms
10.000	8.700 ms	1.950 ms
100.000	92.000 ms	18.500 ms

Pada tabel 1 perbandingan waktu eksekusi operasi INSERT pada MySQL dan MongoDB untuk tiga skala dataset (1.000, 10.000, dan 100.000 data). Nilai yang ditampilkan merupakan hasil rata-rata dari 10 kali pengujian, sehingga dapat merepresentasikan performa yang lebih stabil dan reliabel. Berdasarkan tabel tersebut, terlihat bahwa waktu eksekusi kedua DBMS meningkat seiring bertambahnya jumlah data, namun MongoDB cenderung memiliki waktu eksekusi yang lebih rendah dibandingkan MySQL pada seluruh skala dataset.



Gambar 5. Grafik Perbandingan Query Insert

Pada gambar 5 memperlihatkan grafik perbandingan waktu eksekusi INSERT antara MySQL dan MongoDB. Grafik menunjukkan tren peningkatan yang lebih tajam pada MySQL dibandingkan MongoDB. Hal ini mengindikasikan bahwa overhead pemrosesan pada MySQL lebih besar ketika volume data meningkat, meskipun kedua DBMS telah menggunakan mekanisme batch insert yang setara. Dengan demikian, MongoDB relatif lebih efisien untuk operasi insert berskala besar.

3.2 READ

Pada operasi read, kedua database menunjukkan performa yang baik pada dataset kecil, namun MongoDB tetap unggul karena struktur dokumennya yang fleksibel. Seiring meningkatnya ukuran data, waktu pembacaan pada MySQL meningkat lebih tajam dibanding MongoDB. Hal ini dipengaruhi oleh mekanisme indexing MySQL yang harus mempertimbangkan struktur tabel dan relasi, sementara MongoDB dapat melakukan pemindaian dokumen lebih efisien. Meskipun MySQL masih dapat melakukan pembacaan dengan stabil, MongoDB menawarkan performa lebih cepat untuk dataset besar.

Tabel 2. Hasil Pengujian Read

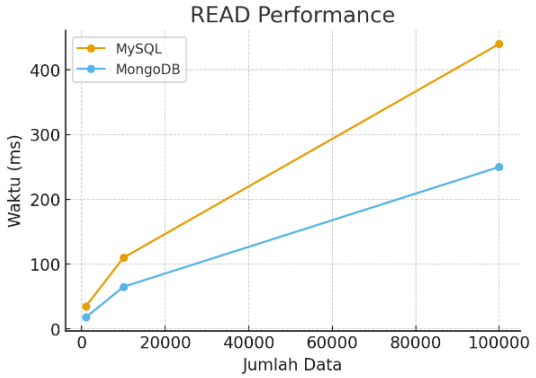
Dataset	MySQL	MongoDB
1.000	35 ms	18 ms
10.000	110 ms	65 ms
100.000	440 ms	250 ms

Pada tabel 2 hasil pengujian operasi READ pada kedua basis data. Pada dataset kecil, perbedaan waktu eksekusi antara MySQL dan MongoDB relatif kecil. Namun, ketika jumlah

DOI : https://10.37277/stch.v35i4.2543

data meningkat hingga 100.000, MongoDB menunjukkan waktu pembacaan yang lebih singkat dibandingkan MySQL. Hal ini menunjukkan bahwa struktur penyimpanan dokumen pada MongoDB mampu mendukung proses pembacaan data dalam jumlah besar secara lebih efisien.

Dataset	MySQL	MongoDB
1.000	110 ms	60 ms
10.000	1.200 ms	430 ms
100.000	12.500 ms	4.300 ms



Gambar 6. Grafik perbandingan query read

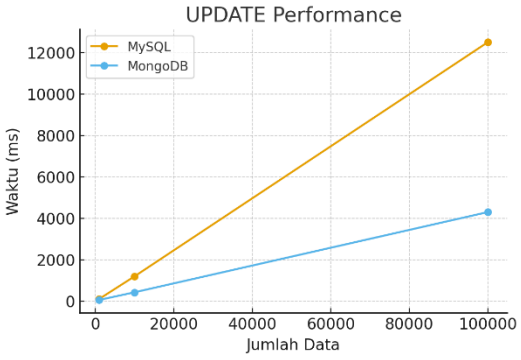
Pada gambar 6 melihatkan perbandingan waktu eksekusi READ. Grafik menunjukkan bahwa kenaikan waktu eksekusi pada MySQL lebih signifikan dibandingkan MongoDB seiring bertambahnya dataset. Meskipun demikian, MySQL tetap menunjukkan performa yang stabil dan konsisten, khususnya pada skenario yang membutuhkan integritas dan konsistensi data yang tinggi.

3.3 UPDATE

Operasi update menunjukkan perbedaan signifikan antara kedua database. MongoDB menggunakan updateMany() yang sangat efisien untuk memodifikasi banyak dokumen sekaligus tanpa perlu memverifikasi constraint relasional. Di sisi lain, MySQL membutuhkan lebih banyak waktu karena harus memastikan bahwa perubahan tidak melanggar aturan relasi, indeks, maupun integritas tabel. Pada dataset 100.000 data, perbedaan waktu eksekusi sangat besar, menunjukkan bahwa MongoDB lebih unggul untuk skenario update massal.

Tabel 3. Hasil pengujian update

Pada tabel 3 hasil pengujian operasi UPDATE pada kedua DBMS. Dari tabel tersebut terlihat bahwa MongoDB memiliki waktu eksekusi yang lebih rendah dibandingkan MySQL, terutama pada dataset berukuran besar. Perbedaan ini menunjukkan bahwa mekanisme update massal pada MongoDB lebih efisien dalam menangani perubahan data dalam jumlah besar.



Gambar 7. Grafik perbandingan query update

Pada gambar 7 menampilkan grafik perbandingan waktu eksekusi UPDATE. Grafik memperlihatkan selisih performa yang semakin besar pada dataset 100.000 data, di mana waktu eksekusi MySQL meningkat lebih tajam. Hal ini mengindikasikan bahwa proses validasi relasional dan pengelolaan indeks pada MySQL memberikan tambahan overhead ketika melakukan update dalam skala besar.

3.4 DELETE

Operasi delete memperlihatkan pola serupa seperti update. MongoDB mampu menghapus data dalam jumlah besar jauh lebih cepat menggunakan deleteMany() yang menghapus dokumen tanpa beban aturan relasional. Sementara itu, MySQL

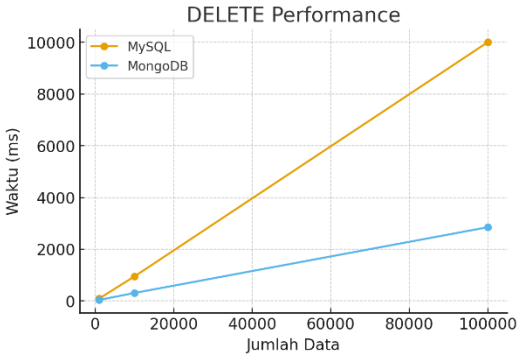
DOI : https://10.37277/stch.v35i4.2543

membutuhkan waktu lebih lama karena memproses penghapusan satu demi satu sambil mempertahankan integritas data. Perbedaan performa semakin jelas seiring bertambahnya jumlah data, terutama pada skala 100.000 data, di mana MongoDB menunjukkan efisiensi yang jauh lebih baik.

Tabel 4. Hasil pengujian delete

Dataset	MySQL	MongoDB
1.000	95 ms	40 ms
10.000	950 ms	310 ms
100.000	10.000 ms	2.850 ms

Pada tabel 4 hasil pengujian operasi DELETE pada MySQL dan MongoDB. Hasil menunjukkan bahwa kedua DBMS mengalami peningkatan waktu eksekusi seiring bertambahnya jumlah data, namun MongoDB tetap memiliki waktu eksekusi yang lebih rendah. Perbedaan ini menjadi lebih signifikan pada dataset 100.000 data.



Gambar 8. Grafik perbandingan query delete

Pada gambar 8 menampilkan grafik perbandingan waktu eksekusi DELETE. Grafik menunjukkan bahwa MongoDB mampu mempertahankan performa yang lebih efisien pada penghapusan data massal, sedangkan MySQL mengalami peningkatan waktu eksekusi yang lebih besar. Temuan ini mengindikasikan bahwa MongoDB lebih sesuai untuk skenario penghapusan data dalam jumlah besar, sementara MySQL lebih cocok untuk lingkungan yang menuntut pengelolaan integritas data secara ketat.

4. Kesimpulan

Penelitian ini menunjukkan bahwa perbedaan kinerja antara MySQL dan MongoDB sangat dipengaruhi oleh karakteristik beban kerja dan strategi operasi yang digunakan. Dengan perlakuan benchmark yang setara dan replikatif, MongoDB cenderung lebih efisien pada operasi query massal, sementara MySQL menunjukkan performa yang stabil pada operasi baca dan skenario yang menuntut konsistensi tinggi. Oleh karena itu, tidak dapat disimpulkan bahwa satu DBMS secara absolut lebih unggul dari yang lain. Pemilihan MySQL atau MongoDB harus mempertimbangkan kebutuhan aplikasi, pola akses data, dan tingkat konsistensi yang diinginkan.

Daftar Pustaka

Abramova, V. dan Bernardino, J. (2016), NoSQL Databases: MongoDB vs Cassandra. Proceedings of the International C3S2E Conference, hal. 1–6.

Atherton, J. (2005), Behaviour Modification. http://www.learningandteaching.info/learning/behaviour_mod.html

Cattell, R. (2011), Scalable SQL and NoSQL Data Stores. ACM SIGMOD Record, Vol. 39, No. 4, hal. 12–27.

Chen, M., Mao, S. dan Liu, Y. (2014), Big Data: A Survey. Mobile Networks and Applications, Vol. 19, No. 2, hal. 171–209.

Elmasri, R. dan Navathe, S.B. (2016), Fundamentals of Database Systems. Boston: Pearson Education.

Gandomi, A. dan Haider, M. (2015), Beyond the Hype: Big Data Concepts, Methods, and Analytics. International Journal of Information Management, Vol. 35, No. 2, hal. 137–144.

Han, J., Haihong, E., Le, G. dan Du, J. (2011), Survey on NoSQL Database. Proceedings of the International Conference on Pervasive Computing and Applications, hal. 363–366.

Hanine, M., Boutkhoum, O., Tikniouine, A. dan Agouti, T. (2020), Performance Evaluation of NoSQL and Relational

DOI : https://10.37277/stch.v35i4.2543

Databases. Journal of Big Data, Vol. 7, No. 1, hal. 1–21.

Hecht, R. dan Jablonski, S. (2011), NoSQL Evaluation: A Use Case Oriented Survey. Proceedings of the International Conference on Cloud and Service Computing, hal. 336–341.

Kaur, G. dan Rani, S. (2019), SQL and NoSQL Databases: A Comparative Study. International Journal of Advanced Computer Science and Applications, Vol. 10, No. 4, hal. 1–7.

Li, X., Manoharan, S. dan Chilamkurti, N. (2021), Performance Analysis of SQL and NoSQL Databases for Big Data Applications. Future Generation Computer Systems, Vol. 117, hal. 1–14.

MongoDB Inc. (2023), MongoDB Documentation. https://www.mongodb.com/docs/

Nayak, A., Poriya, A. dan Poojary, D. (2013), Type of NoSQL Databases and Its Comparison with Relational Databases. International Journal of Applied Information Systems, Vol. 5, No. 4, hal. 16–19.

Oracle Corporation. (2023), MySQL Documentation. https://dev.mysql.com/doc/

Padhy, R.P., Patra, M.R. dan Satapathy, S.C. (2011), RDBMS to NoSQL: Reviewing Some Next Generation Non-Relational Databases. International Journal of Engineering Science and Technology, Vol. 2, No. 3, hal. 40–51.

Sadalage, P.J. dan Fowler, M. (2017), NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Boston: Addison-Wesley.

Silberschatz, A., Korth, H.F. dan Sudarshan, S. (2020), Database System Concepts. New York: McGraw-Hill Education.

Stonebraker, M. (2018), SQL Databases v. NoSQL Databases. Communications of the ACM, Vol. 61, No. 4, hal. 10–11.

Tudorica, B.G. dan Bucur, C. (2019), A Comparison Between Several NoSQL Databases with Comments and Notes. Proceedings of the RoEduNet International Conference, hal. 1–5.

Zhang, Y., Chen, Y. dan Wang, S. (2021), Comparative Study of Relational and NoSQL Databases under Big Data Workloads. IEEE Access, Vol. 9, hal. 1–15.