

Comparative Study of MySQL and MongoDB Performance on Node.js REST API Using Chinook Database

Studi Perbandingan Performa MySQL dan MongoDB pada REST API Node.js Menggunakan Database Chinook

Iman Setiawan ¹⁾

¹⁾Teknik Informatika, Fakultas Teknik, Universitas Pelita Bangsa

Email: ¹⁾ imansetiawan1998@mhs.pelitabangsa.ac.id

How to Cite :

Setiawan, I. (2025). Comparative Study of MySQL and MongoDB Performance on Node.js REST API Using Chinook Database. Jurnal Media Computer Science, 4(2). Doi: <https://doi.org/10.37676/jmcs.v4i2>

ARTICLE HISTORY

Received [25 Juni 2025]

Revised [10 Juli 2025]

Accepted [11 Juli 2025]

KEYWORDS

MySQL, MongoDB, REST API,
Node.js, Apache JMeter.

This is an open access article under the
[CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



ABSTRAK

Kebutuhan akan sistem manajemen basis data (Database Management System) yang efisien dan optimal semakin penting, terutama dalam pengembangan aplikasi berbasis REST API. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan performa dua jenis DBMS yang umum digunakan, yaitu MySQL (relasional) dan MongoDB (non-relasional), ketika diimplementasikan dalam REST API berbasis Node.js. Pengujian dilakukan di lingkungan server nyata untuk mencerminkan kondisi operasional sesungguhnya, menggunakan alat Apache JMeter dengan empat parameter utama, yaitu processed requests, response time, throughput, dan error rate. Data yang digunakan bersumber dari Chinook Database yang telah diperbanyak hingga mencapai 10.000 entri, dan diuji dengan beban pengguna virtual sebanyak 50 dan 100 pengguna selama 120 detik, dengan skenario yang telah diatur. Hasil penelitian menunjukkan bahwa MySQL unggul pada hampir seluruh parameter, dengan processed requests tertinggi (64.098 permintaan), response time terendah (135 ms), dan throughput terbesar (532 permintaan/detik), serta tanpa error. Sebaliknya, MongoDB mencerminkan karakteristik arsitektur non-relasional yang lebih dinamis, namun menunjukkan performa yang lebih rendah pada skenario pengujian ini. Kesimpulan dari penelitian ini menyatakan bahwa pemilihan DBMS harus disesuaikan dengan kebutuhan aplikasi, di mana MySQL lebih cocok untuk pengolahan data yang terstruktur dan konsisten, sedangkan MongoDB lebih sesuai untuk data semi-terstruktur atau tidak terstruktur yang berkembang secara dinamis.

ABSTRACT

The need for an efficient and optimal Database Management System (DBMS) is increasingly critical, especially in the development of REST API-based applications. This study aims to evaluate and compare the performance of two commonly used types of DBMS MySQL (relational) and MongoDB (non-relational) when implemented in a Node.js-based REST API. Testing was conducted in a real server environment to reflect actual operational conditions, using Apache JMeter as the benchmarking tool and focusing on four main parameters processed requests, response time, throughput, and error rate. The dataset was sourced from the Chinook Database, which was expanded to contain 10,000 records and tested under virtual user loads of 50 and 100 users for 120 seconds, under predefined test scenarios. The results showed that MySQL outperformed

MongoDB in nearly all parameters, with the highest processed requests (64,098 requests), lowest response time (135 ms), and highest throughput (532 requests/second), with no errors. On the other hand, MongoDB reflected the more dynamic characteristics of non-relational architecture but demonstrated lower performance in this test scenario. The conclusion of this study emphasizes that the choice of DBMS should be based on the specific needs of the application—MySQL is more suitable for structured and consistent data processing, while MongoDB is better suited for semi-structured or unstructured data that evolves dynamically.

PENDAHULUAN

Database merupakan fondasi utama dalam sistem informasi modern yang berfungsi untuk menyimpan, mengelola, dan mengakses data secara efisien. Seiring berkembangnya kebutuhan teknologi, jenis basis data pun semakin beragam, salah satunya dibedakan menjadi sistem basis data relasional dan non-relasional. *MySQL* adalah contoh dari basis data relasional yang menyimpan data dalam tabel dengan skema tetap dan menjamin integritas melalui konsep *ACID*. Sementara itu, *MongoDB* merupakan contoh basis data non-relasional yang menyimpan data dalam bentuk dokumen fleksibel berformat *BSON*, dan menawarkan kemampuan skalabilitas horizontal yang baik. Perbedaan arsitektur ini menjadikan masing-masing memiliki keunggulan tersendiri tergantung pada kebutuhan sistem. Dalam praktik pengembangan aplikasi, pemilihan database sering kali tidak hanya didasarkan pada jenis data yang ditangani, tetapi juga pada performa yang dihasilkan ketika menghadapi beban kerja tertentu, terutama dalam lingkungan yang membutuhkan kecepatan dan efisiensi tinggi.

REST API (Representational State Transfer Application Programming Interface) adalah arsitektur web yang pada saat ini menjadi standar komunikasi data antara *client* dan *server* dalam aplikasi berbasis web maupun *mobile*. Dalam pengembangan *REST API*, kinerja *backend* sangat bergantung pada jenis basis data yang digunakan untuk menyimpan dan memproses data. Oleh karena itu, penting untuk mengetahui bagaimana karakteristik performa masing-masing database saat digunakan dalam arsitektur *REST API* berbasis *Node.js*. Faktor seperti kecepatan dalam menangani permintaan, jumlah permintaan yang dapat ditangani secara bersamaan, tingkat kegagalan (*error rate*), serta *throughput* menjadi parameter penting dalam mengukur efektivitas integrasi antara database dan API.

Berdasarkan beberapa penelitian terdahulu yang dilakukan oleh Winata, dkk. (2021), diketahui bahwa mereka membandingkan performa waktu respons kueri dasar seperti *SELECT*, *INSERT*, *UPDATE*, dan *DELETE* antara *MySQL*, *PostgreSQL*, dan *MongoDB* menggunakan *dataset Android Permission* tanpa relasi antar tabel. Pengujian dilakukan pada lima tingkatan data mulai dari 250 hingga 25.000 baris, dan hasilnya menunjukkan bahwa *PostgreSQL* unggul di hampir semua skenario dengan waktu respons yang stabil dan cepat, rata-rata di bawah 150ms.

Sementara itu, penelitian oleh Sinaga, dkk (2023) membandingkan dua database *NoSQL*, yaitu *MongoDB* dan *Elasticsearch*, dan menemukan bahwa *MongoDB* secara konsisten lebih unggul pada seluruh operasi *CRUD*, dengan kecepatan hingga ratusan kali lipat lebih cepat dibanding *Elasticsearch*, terutama dalam proses *create* dan *read*. Selanjutnya, penelitian oleh Andrianto dan Suyatno (2024) membandingkan *MySQL* dan *MongoDB* dalam implementasi *REST API* berbasis *Node.js*, dengan beban pengguna virtual menggunakan *Apache JMeter*. Hasil pengujian menunjukkan bahwa *MySQL* memiliki performa lebih stabil, dengan *response time* lebih rendah, *throughput* lebih tinggi, dan tanpa *error* saat diuji pada kondisi beban tinggi (Andrianto & Suyatno, 2024).

Penelitian ini hadir sebagai upaya untuk memvalidasi dan melengkapi studi yang telah dilakukan oleh Andrianto & Suyatno (2024) yang membahas perbandingan performa *MySQL* dan *MongoDB* pada *REST API*. Berbeda dari penelitian sebelumnya yang menggunakan *Postman* untuk pengujian, studi ini mencoba memberikan pengujian yang lebih mendekati skenario nyata

penggunaan sample database *Chinook* dengan menggunakan *Apache JMeter*, beban *virtual user* hingga 100 pengguna, serta parameter evaluasi tambahan seperti *processed requests*, *response time*, *throughput*, dan *error rate*.

Penulis tidak bermaksud memberikan inovasi besar dalam metodologi, namun ingin ikut serta mengonfirmasi dan menyumbangkan data tambahan dari hasil pengujian dengan pendekatan yang sedikit berbeda. Dengan memanfaatkan struktur data dari *Chinook Database* dan penggunaan *REST API* sederhana berbasis *Node.js*, penelitian ini diharapkan mampu memberikan kontribusi sebagai acuan praktis bagi para pengembang dalam menentukan sistem basis data yang tepat untuk aplikasi berskala ringan hingga menengah.

LANDASAN TEORI

Database Management System

Database Management System (DBMS) merupakan perangkat lunak yang dirancang untuk menyimpan, mengelola, dan mengambil data secara efisien, baik untuk kebutuhan skala kecil maupun besar. DBMS menyediakan seperangkat fungsionalitas seperti manajemen transaksi, kontrol akses, dan optimasi kueri, yang memungkinkan pengguna untuk mengelola data tanpa harus membangun sistem penyimpanan data dari awal. Dengan adanya DBMS, organisasi dapat memusatkan data mereka dalam satu sistem yang terstruktur dan dapat diakses oleh banyak aplikasi secara serempak, sehingga efisiensi dan keamanan data lebih terjamin (Taipalus, 2024).

Terdapat berbagai jenis DBMS, yang secara umum diklasifikasikan menjadi dua kategori utama, yaitu SQL (*Structured Query Language*) dan NoSQL (*Not Only SQL*). SQL digunakan dalam sistem basis data relasional yang mengandalkan tabel terstruktur dan hubungan antar-entitas, cocok untuk aplikasi dengan data yang konsisten dan terstruktur. Sebaliknya, NoSQL digunakan untuk menyimpan data semi-terstruktur atau tidak terstruktur, seperti dokumen JSON atau data graf, dan dirancang agar lebih fleksibel dan skalabel. Pemilihan antara SQL dan NoSQL sangat bergantung pada jenis data serta kebutuhan akses dan skala aplikasi (Taipalus, 2024).

REST API

REST API (*Representational State Transfer Application Programming Interface*) merupakan arsitektur web dalam implementasi API yang berfungsi untuk menghubungkan serta memfasilitasi komunikasi antara satu aplikasi dengan aplikasi lainnya melalui jaringan internet. (Andrianto & Suyatno, 2024). REST API bekerja dengan prinsip HTTP dan menyediakan akses terhadap data melalui metode standar seperti *GET*, *POST*, *PUT*, dan *DELETE*. Pengguna tidak dapat mengakses *database* secara langsung, melainkan melalui *endpoint* yang disediakan, sehingga data tetap aman dan terstruktur. Dengan REST API, pengguna dapat memperoleh informasi yang selalu diperbarui jika terjadi perubahan data pada sistem.

Nodejs

Node.js adalah sebuah platform yang digunakan untuk menjalankan kode *JavaScript* di luar browser, dibangun di atas mesin V8 milik *Google Chrome*. Platform ini sangat populer di kalangan pengembang karena mudah digunakan dan efisien dalam menangani proses *input/output* (I/O), seperti membaca file atau berkomunikasi dengan server, berkat model kerjanya yang berbasis *event-driven* dan *non-blocking*.

JavaScript sendiri merupakan bahasa pemrograman yang hanya berjalan di satu jalur (*single-threaded*) dan tidak bisa langsung mengakses fitur-fitur tingkat rendah dari sistem operasi, seperti membuat thread atau menjalankan perintah sistem. Untuk mengatasi hal tersebut, *Node.js* menggunakan kode tambahan yang ditulis dalam bahasa C/C++ agar tetap bisa menjalankan tugas-tugas tersebut dengan cepat dan efisien (Kyriakou & Tselikas, 2022).

Chinook Database Sample

Chinook Database adalah sebuah basis data sampel yang tersedia untuk umum dan struktur datanya secara luas dipergunakan dalam berbagai penelitian dan praktik pengujian sistem basis data. Basis data ini merepresentasikan sebuah perusahaan yang beroperasi di sektor penjualan media digital, termasuk produk audio dan video. Data di dalamnya sebagian besar berasal dari *iTunes Library*, sementara informasi mengenai pelanggan dan karyawan dibuat secara fiktif, dan data penjualan digenerasi secara acak untuk periode empat tahun. Karena struktur datanya yang lengkap—mencakup artis, album, lagu (*media tracks*), faktur (*invoice*), dan pelanggan Chinook sering dimanfaatkan sebagai studi kasus dalam perancangan *data warehouse* guna mendukung pengambilan keputusan strategis, seperti promosi musik (Putra dkk., 2023).

Apache Jmeter

Apache JMeter merupakan alat yang banyak digunakan untuk analisis dan pengujian kinerja *REST API*. Pemilihan alat ini didasarkan pada sejumlah keunggulan dibandingkan dengan alat lain seperti HammerDB dan SysBench, antara lain antarmuka yang ramah pengguna, kompatibilitas bawaan dengan sistem operasi Windows, dukungan terhadap berbagai basis data seperti MySQL dan Oracle, serta dokumentasi daring yang komprehensif. Kombinasi antara desain berbasis pengguna, kompatibilitas lintas platform, dan kemampuan analisis hasil yang kuat menjadikan Apache JMeter sebagai alat pengujian performa yang ideal dalam konteks penelitian ini (Indrianto, 2023; Tejaya dkk, 2023).

METODE PENELITIAN

Penelitian ini menerapkan pendekatan kuantitatif komparatif deskriptif guna mengevaluasi dan membandingkan kinerja dua sistem manajemen basis data, yaitu *MySQL* sebagai basis data relasional dan *MongoDB* sebagai basis data non-relasional, dalam konteks implementasi *REST API* menggunakan *Node.js*. Metode ini dipilih untuk memperoleh data numerik yang dapat dianalisis secara objektif, dengan tujuan mengukur efektivitas kedua sistem database berdasarkan parameter teknis tertentu. Berikut adalah alur penelitian yang akan dilakukan:

1. Identifikasi Masalah dan Tujuan Penelitian
2. Pemilihan dan Konversi Dataset
3. Pengembangan *REST API*
4. Deployment ke Server Cloud
5. Desain Skenario dan Eksekusi Pengujian
6. Pengumpulan dan Analisis Data
7. Penarikan Kesimpulan dan Saran

Dataset yang digunakan dalam penelitian ini adalah *Chinook Database*, yaitu dataset simulasi industri musik digital yang umum digunakan dalam pengujian sistem informasi. *Dataset* ini tersedia dalam format SQL dan dirancang khusus untuk sistem basis data relasional seperti *MySQL*. Oleh karena itu, untuk dapat digunakan pada sistem basis data non-relasional seperti *MongoDB*, dataset ini perlu melalui proses konversi struktur data. Proses konversi dilakukan secara langsung dengan prinsip kesetaraan, di mana setiap tabel pada *MySQL* diubah menjadi *collection* dalam *MongoDB*, dan setiap baris data diubah menjadi *document*. Tidak dilakukan normalisasi ulang maupun penerapan fitur khusus *MongoDB* seperti *embedding* atau *referencing*, sehingga struktur data pada kedua sistem tetap seragam.

Pendekatan ini bertujuan agar hasil pengujian performa dapat dibandingkan secara adil dan setara tanpa adanya bias desain skema. Selain itu, jumlah data dalam database juga diperluas

secara terkontrol hingga mencapai sekitar 10.000 entri dengan menduplikasi dan menyebarkan data sesuai struktur aslinya. Penambahan data ini bertujuan untuk menciptakan skenario uji performa yang lebih realistis dan menantang, serta mendekati kondisi sistem dalam skala operasional sesungguhnya.

Aplikasi REST API dibangun menggunakan runtime environment Node.js dengan framework Express.js. Sistem REST API ini dirancang untuk membuat operasi dasar CRUD (Create, Read, Update, Delete) dan juga operasi cukup kompleks seperti agregasi serta analitik untuk pengujian, sesuai dengan kebutuhan aplikasi web modern. Untuk setiap jenis database, digunakan endpoint yang identik agar proses pengujian dapat dilakukan secara simetris dan obyektif.

Pengujian performa dilakukan menggunakan Apache Jmeter, dilakukan dalam dua skenario beban pengguna, yaitu dengan 50 dan 100 virtual users, yang masing-masing dijalankan selama 120 detik dan diulang sebanyak 3 kali untuk memastikan konsistensi dan reliabilitas hasil. Konfigurasi JMeter mencakup pengaturan thread group, ramp-up time, loop count tak terbatas, dan test duration 2 menit. Empat parameter utama diukur dalam pengujian ini, yaitu:

1. *Processed Requests*, jumlah total permintaan yang berhasil diproses server selama waktu pengujian.
2. *Response Tim*, waktu rata-rata yang dibutuhkan untuk memberikan respons terhadap setiap permintaan.
3. *Throughput*, jumlah permintaan yang berhasil diproses per detik (request per second).
4. *Error Rate*, persentase permintaan yang gagal diproses, yang mengindikasikan stabilitas dan reliabilitas sistem.

Untuk mensimulasikan kondisi penggunaan nyata, pengujian dilakukan dalam lingkungan cloud menggunakan Google Cloud Platform dengan spesifikasi server yang memadai, serta klien pengujian dijalankan secara lokal menggunakan perangkat dengan performa tinggi. Rincian spesifikasi perangkat server dan client yang digunakan dalam pengujian ditampilkan pada tabel berikut:

Tabel 1. Keterangan Server

Spesifikasi Server	
Platform Server	Google Cloud Platform
Jenis VM	Compute Engine
Processor	4 vCPU
Memory	16 GB
Storage	40 GB Persistent SSD
Network	Google Cloud High-Speed Network
Ubuntu Server	24.04
Node.js	22.15.0
Express.js	5.1.0
MySQL	8.4.5
MongoDB	8.0.6

Tabel 2. Keterangan Client

Spesifikasi Server	
Processor	AMD Ryzen 5 (6-Core)
Memory	16GB
Storage	512GB NVME
Memory	16GB
Windows	11 Pro 23H2
Apache JMeter	5.6.3

Untuk memastikan pengujian dilakukan secara adil dan mencerminkan skenario dunia nyata, REST API yang dibangun terdiri dari sepuluh *endpoint* utama yang mewakili berbagai jenis operasi, mulai dari operasi dasar hingga agregasi dan penghapusan data. *Endpoint* tersebut dirancang untuk memproses data yang cukup kompleks serta menyimulasikan kebutuhan umum dalam aplikasi bisnis digital, seperti *dashboard* keuangan, riwayat transaksi, statistik genre musik, hingga manajemen pengguna. Tabel berikut merangkum daftar *endpoint*, metode *HTTP* yang digunakan, skenario masing-masing *endpoint*, serta tabel yang terlibat dari sisi *database*:

Tabel 3. Skenario Pengujian

No	Endpoint	Method	Skenario	Table Involved
1	/total_sales	GET	Melihat total pendapatan untuk KPI dashboard	Invoice
2	/invoice_details/:id	GET	Melihat detail faktur beserta informasi lagu (menggunakan join)	Invoice, InvoiceLine, Track, Album
3	/top_tracks	GET	Menampilkan Track teratas berdasarkan jumlah penjualan	Tracks Album, Track, InvoiceLine
4	/customer/:id/invoices	GET	Pelanggan melihat riwayat faktur mereka	Customer, Invoice
5	/employee_customers/:id	GET	Staf dukungan mengambil data pelanggan yang ditugaskan kepada mereka	Employee, Customer
6	/playlist/:id/tracks	GET	Memuat isi playlist	Playlist, PlaylistTrack, Track
7	/genre_stats	GET	Analitik detail Genre dengan durasi dan harga rata-rata	Genre, Track
8	/add_invoice	POST	Faktur baru ditambahkan saat proses checkout	Invoice, InvoiceLine
9	/update_customer/:id	PUT	Pelanggan memperbarui data pribadi mereka	Customer
10	/delete_playlist/:id	DELETE	Pengguna menghapus playlist lagu	Playlist, PlaylistTrack

Hasil pengujian dianalisis menggunakan metode deskriptif kuantitatif dengan menghitung nilai rata-rata dari setiap parameter pada masing-masing skenario. Analisis dilakukan secara komparatif untuk mengetahui keunggulan relatif dari *MySQL* maupun *MongoDB*, serta mengidentifikasi karakteristik performa masing-masing dalam konteks penggunaan *REST API*. Dengan pendekatan ini, penelitian diharapkan dapat memberikan gambaran menyeluruh dan mendalam mengenai efektivitas kedua sistem basis data dalam implementasi *REST API*, serta menjadi referensi praktis dan ilmiah dalam pengembangan sistem *backend* modern.

Data hasil pengujian dicatat dalam bentuk tabel agar hasil uji coba dapat disajikan secara terstruktur dan sistematis. Penyajian ini memudahkan proses analisis dan perbandingan performa antara dua sistem basis data, yaitu *MySQL* dan *MongoDB*, pada berbagai kondisi beban pengguna. Selanjutnya, data tersebut divisualisasikan dalam bentuk diagram batang guna memberikan gambaran yang lebih jelas dan intuitif terhadap perbedaan performa yang dihasilkan oleh masing-masing sistem.

HASIL DAN PEMBAHASAN

Hasil

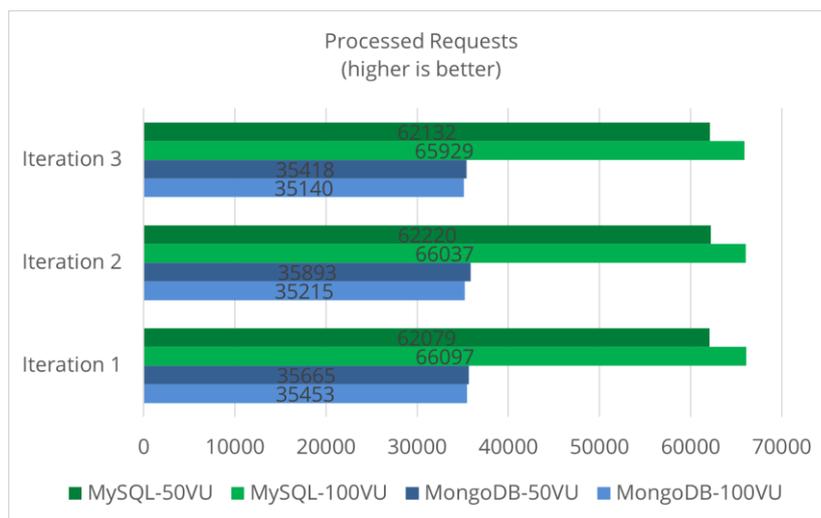
Pengujian performa dilakukan untuk membandingkan dua jenis sistem basis data yang digunakan dalam REST API. Evaluasi dilakukan menggunakan alat bantu pengujian yang telah disesuaikan, dengan berbagai skenario pengguna virtual. Tujuannya adalah untuk mendapatkan gambaran menyeluruh mengenai kinerja masing-masing sistem. Hasil pengujian yang ditampilkan pada setiap iterasi merupakan nilai rata-rata dari masing-masing parameter pengujian yang telah dihitung secara otomatis oleh Apache JMeter, dan disajikan secara lengkap pada tabel berikut.

Tabel 4. Hasil Pengujian

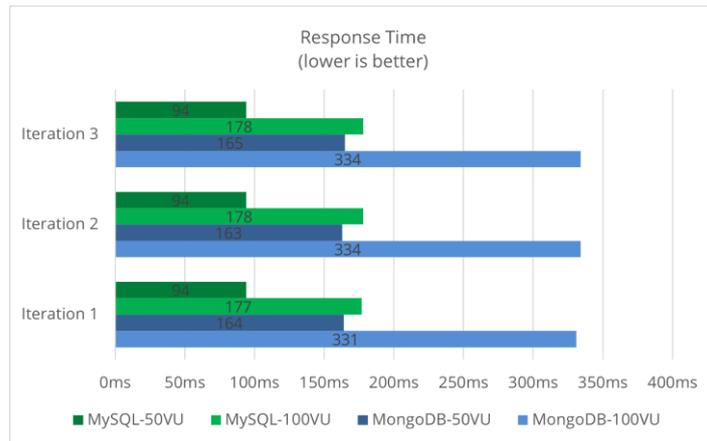
Database	Virtual Users	No	Processed Requests	Response Time (ms)	Thoughtput (per second)	Error Rate (%)
MySQL	50	1	62079	94	516,86	0
		2	62220	94	517,18	0
		3	62227	94	517,57	0
	100	1	66097	177	549,76	0
		2	66037	178	548,76	0
		3	65929	178	547,77	0
MongoDB	50	1	35665	164	296,56	0,02
		2	35893	163	298,53	0,02
		3	35418	165	294,54	0,03
	100	1	35453	331	294,43	0,08
		2	35215	334	292,41	0,06
		3	35140	334	291,88	0,09

Visualisasi Data

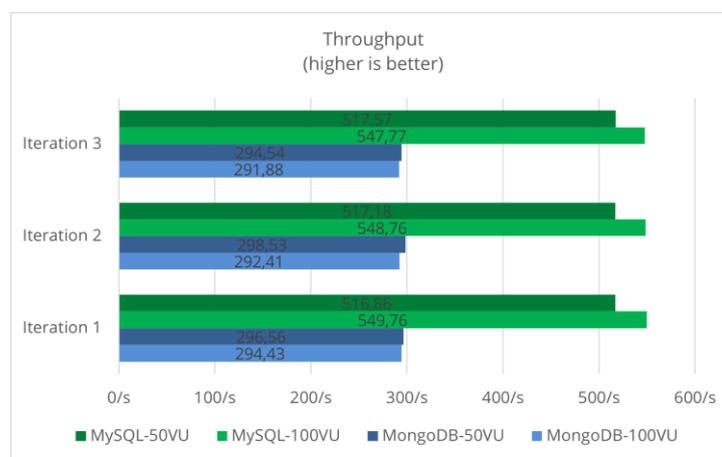
Untuk memberikan gambaran yang lebih jelas mengenai performa sistem yang diuji, melalui pendekatan visual ini, diharapkan hasil analisis menjadi lebih mudah dipahami dan dapat memberikan gambaran yang lebih menyeluruh terkait kekuatan dan kelemahan dari masing-masing sistem yang diuji, data hasil pengujian divisualisasikan dalam bentuk grafik sebagai berikut.



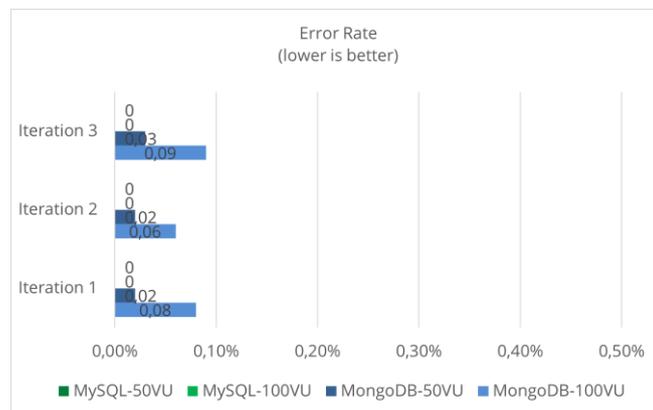
Gambar 1. Grafik Parameter Processed Requests



Gambar 2. Grafik Parameter Response Time



Gambar 3. Grafik Parameter Throughput



Gambar 4. Grafik Parameter Error Rate

Pembahasan

Hasil pengujian menunjukkan bahwa *MySQL* memiliki nilai *Processed Requests* yang lebih tinggi dibandingkan *MongoDB* pada kedua skenario pengujian, baik dengan 50 maupun 100 *virtual users*. *MySQL* mampu menangani lebih dari 62.000 permintaan dalam waktu 2 menit saat diuji dengan 50 pengguna, dan meningkat hingga lebih dari 66.000 permintaan pada 100 pengguna. Sebaliknya, *MongoDB* hanya mampu memproses sekitar 35.000 permintaan pada masing-masing skenario. Perbedaan signifikan ini menunjukkan bahwa arsitektur relasional *MySQL* lebih efisien dalam

menangani beban transaksi simultan pada struktur data yang kompleks, terutama dalam sistem yang membutuhkan konsistensi tinggi dan eksekusi *query* berlapis.

Pada parameter *Response Time*, sistem berbasis *MySQL* juga menunjukkan performa yang lebih unggul. Rata-rata waktu tanggapan *MySQL* berada di kisaran 94 milidetik pada 50 pengguna dan meningkat menjadi sekitar 178ms pada 100 pengguna. Sementara itu, *MongoDB* mencatat waktu tanggapan yang lebih lambat, yaitu 164–165ms pada 50 pengguna, dan meningkat drastis hingga 334ms pada 100 pengguna. Waktu respons yang lebih cepat pada *MySQL* menunjukkan bahwa sistem ini lebih responsif dan stabil saat menghadapi beban permintaan tinggi, yang sangat penting dalam menjaga kenyamanan pengguna dalam aplikasi nyata.

Dari sisi *Throughput*, yaitu jumlah permintaan yang dapat diproses per detik, *MySQL* kembali unggul dengan capaian di atas 500 *request per second* pada semua skenario. *MongoDB*, di sisi lain, hanya mampu memproses sekitar 290 *request per second* meskipun beban pengguna meningkat. Hal ini menunjukkan bahwa sistem berbasis *MySQL* memiliki kemampuan pengolahan yang lebih efisien dalam satuan waktu yang sama. *Throughput* yang tinggi sangat krusial dalam konteks sistem yang melayani banyak pengguna secara bersamaan, seperti pada aplikasi *e-commerce*, layanan musik *streaming*, atau sistem pelaporan *real-time*.

Pada parameter terakhir, yaitu *Error Rate*, *MySQL* berhasil mempertahankan performa tanpa mencatatkan satu pun kesalahan atau kegagalan proses (0%). Sebaliknya, *MongoDB* mencatat *error rate* yang bervariasi dari 0,02% hingga 0,09%, terutama saat diuji pada skenario 100 pengguna. Meskipun persentasenya kecil, kemunculan *error* dalam sistem produksi dapat memengaruhi keandalan dan pengalaman pengguna secara keseluruhan. Hasil ini mengindikasikan bahwa *MySQL* tidak hanya unggul dalam hal kecepatan dan kapasitas, tetapi juga dalam hal stabilitas dan keandalan sistem saat beroperasi di bawah tekanan.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil pengujian performa terhadap implementasi REST API menggunakan dua sistem basis data yang berbeda arsitektur, yaitu *MySQL* (relasional) dan *MongoDB* (non-relasional), diperoleh bahwa *MySQL* menunjukkan performa yang lebih unggul. Hal ini terlihat dari jumlah permintaan yang berhasil diproses, waktu respon yang lebih cepat, *throughput* yang lebih tinggi, serta kestabilan sistem dengan *error rate* 0%. Performa *MongoDB* dalam pengujian ini tidak optimal karena struktur data yang digunakan merupakan hasil konversi dari basis data relasional, tanpa memanfaatkan fitur khas *MongoDB* seperti skema dinamis, *embedding*, dan *referencing*.

Temuan ini mengindikasikan bahwa pemilihan sistem basis data perlu disesuaikan dengan karakteristik dan kebutuhan spesifik dari aplikasi yang dikembangkan, struktur data, dan desain arsitektur backend. *MySQL* lebih sesuai digunakan dalam sistem dengan struktur data yang kompleks, relasional, dan memerlukan konsistensi transaksi yang tinggi. Sementara itu, *MongoDB* tetap relevan untuk skenario data tidak terstruktur yang memerlukan fleksibilitas dan skalabilitas, asalkan desain databasenya dioptimalkan. Penelitian ini diharapkan dapat menjadi referensi dalam pengambilan keputusan teknis terkait pemilihan basis data untuk pengembangan sistem berbasis REST API di masa mendatang.

Saran

Berdasarkan hasil pengujian, disarankan agar pengembang sistem yang ingin menggunakan *MongoDB* melakukan perancangan skema data yang benar-benar sesuai dengan karakteristik *NoSQL*, bukan sekadar melakukan konversi langsung dari SQL. Penggunaan fitur-fitur khas *MongoDB* seperti *embedding*, *referencing*, serta pemodelan dokumen secara menyeluruh perlu diterapkan agar performa *database* dapat dioptimalkan. Selain itu, dalam perbandingan performa yang adil di masa depan, penting untuk memastikan bahwa struktur dan kompleksitas data yang digunakan pada masing-masing sistem *database* benar-benar setara secara fungsional dan konteks penggunaannya.

Untuk pengembangan penelitian selanjutnya, disarankan dilakukan pengujian dengan skala data yang lebih besar, misalnya ratusan ribu hingga jutaan entri, agar dapat mengevaluasi ketahanan dan skalabilitas sistem secara lebih komprehensif. Selain itu, dapat ditambahkan parameter pengujian lain seperti *CPU usage*, *memory consumption*, dan *connection latency* agar gambaran performa sistem menjadi lebih jelas. Pendekatan ini akan memperkaya wawasan terkait pemilihan arsitektur basis data yang tepat sesuai kebutuhan spesifik dari sistem backend yang dikembangkan.

DAFTAR PUSTAKA

- Andrianto, L. D., & Suyatno, D. F. (2024). Analisis performa load testing antara MySQL dan NoSQL MongoDB pada REST API Node.js menggunakan Postman. *Journal of Emerging Information Systems and Business Intelligence (JEISBI)*, 5(1), 18–26.
- Budiman, S., Saputro, V. A., & Fadhila, F. (2021). Analisis performa kecepatan MySQL dan NoSQL pada sistem operasi Windows dan Linux. *JNANALOKA*, 2(1), 21–26.
- Haryadi, H. L., Sujjada, A., & Simatupang, D. S. (2023). Perbandingan REST API menggunakan Node.js dan PHP pada aplikasi pemilihan umum. *Jurnal Riset Sistem Informasi dan Teknik Informatika (JURASIK)*, 8(2), 460–468.
- Indrianto, I. (2023). Performance testing on web information system using Apache JMeter and BlazeMeter. *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, 7(2), 138–149.
- Kyriakou, K.-I. D., & Tselikas, N. D. (2022). Complementing JavaScript in high-performance Node.js and web applications with Rust and WebAssembly. *Electronics*, 11(3217). <https://doi.org/10.3390/electronics11193217>
- Mutmainnah, M. A., Musyriyah, & Zulkarnaim, N. (2022). Perbandingan relational database dan non-relational database dalam pengembangan smart tourism. *Jurnal Teknik Informatika dan Sistem Informasi (JUTISI)*, 8(1), 150.
- Pratama, D. R., Irmawati, B., & Robbani, R. (2023). Pengembangan REST API SIABANG (Sistem Administrasi Pembangunan) menggunakan Java. *JBegaTI*, 4(1).
- Pujas, D., Amar, M. S., Amar, M. I., Arif, M. E., & Mustamin, M. M. (2024). Pengukuran kinerja database SQL dan NoSQL pada aplikasi e-commerce. *Jurnal Fokus Elektroda: Energi Listrik, Telekomunikasi, Komputer, Elektronika dan Kendali*, 9(1), 12–15.
- Putra, G. N. D. W., & Pramatha, C. (2023). Desain dan implementasi data warehouse penjualan pada Chinook sample database. *Jurnal Elektronik Ilmu Komputer Udayana (JELIKU)*, 12(2), 323–334.
- Putra, Y. Y., Purwaningrum, O., & Winata, R. H. (2022). Perbandingan performa respon waktu kueri MySQL, PostgreSQL, dan MongoDB. *Jurnal Sistem Informasi dan Bisnis Cerdas (SIBC)*, 15(1), 39–48.
- Setyawan, A. B., Kautsar, I. A., & Azizah, N. L. (2022). Query response time comparison SQL and NoSQL for contact tracing application: Perbandingan waktu respons kueri SQL dan tanpa SQL untuk aplikasi pelacakan kontak. *Procedia of Engineering and Life Science*, 2(2).
- Sinaga, T. L., Charibaldi, N., & Cahyana, N. H. (2023). Perbandingan waktu respon aplikasi database NoSQL Elasticsearch dan MongoDB pada pengujian operasi CRUD. *Jurnal Informatika Sunan Kalijaga (JISKA)*, 8(1), 22–36.
- Taipalus, T. (2024). Vector database management systems: Fundamental concepts, use-cases, and current challenges. *Cognitive Systems Research*, 85, 101216.
- Tejaya, W., Rahman, S., & Munir, A. (2023). Pengujian website Invitees menggunakan metode load testing dengan Apache JMeter. *KHARISMA Tech*, 18(1), 99–112.