Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

EVALUASI KINERJA OPERASI INSERT PADA MARIADB DAN MONGODB BERDASARKAN METODE PEMASUKAN DATA DAN UKURAN BATCH

Zulkarnaen Lubis[1*], Taufiq[2]
[1,2]Universitas Malikussaleh, Indonesia
[1*]zulkarnaen.240170101@mhs.unimal.ac.id, [2]taufiq.te@unimal.ac.id

ABSTRACT

This study evaluates the INSERT performance of MariaDB and MongoDB based on data insertion methods and batch sizes. The experiment was conducted using 10,000 records per test scenario, with five repetitions for each configuration. Two insertion methods were compared: one-by-one insertion and batch insertion using batch sizes of 100, 500, and 1,000 records. The experiment produced 40 measurements across both database management systems. The results show that batch insertion was faster than one-by-one insertion in both databases. For one-by-one insertion, MariaDB achieved an average execution time of 0.7820 seconds, compared with 1.5691 seconds for MongoDB. For batch insertion with sizes of 100, 500, and 1,000, MariaDB recorded average execution times of 0.1513, 0.1201, and 0.1155 seconds, respectively, while MongoDB recorded 0.0896, 0.0645, and 0.0656 seconds. Thus, MongoDB achieved lower average execution times for all tested batch sizes, while MariaDB performed better in the one-by-one scenario. These findings indicate that the insertion method strongly affects INSERT performance, whereas increasing the batch size beyond 500 records produced no clear additional improvement under the tested experimental conditions.

Keywords:
MariaDB, MongoDB, INSERT performance, batch insertion, database benchmarking

PENDAHULUAN

Perkembangan teknologi informasi mendorong meningkatnya penggunaan sistem basis data untuk menyimpan dan mengelola data dalam jumlah besar. Basis data memiliki peran penting dalam berbagai aplikasi, mulai dari sistem akademik hingga layanan berbasis web yang membutuhkan proses penyimpanan data secara cepat dan efisien. Seiring dengan meningkatnya kebutuhan tersebut, pemilihan sistem manajemen basis data yang sesuai menjadi hal penting, terutama dalam menentukan kemampuan sistem menangani operasi pemasukan data (*INSERT*).

MariaDB merupakan sistem manajemen basis data relasional yang menggunakan struktur tabel dengan skema yang terdefinisi, sedangkan MongoDB merupakan sistem manajemen basis data berorientasi dokumen yang menyimpan data dalam format dokumen. Keduanya memiliki pendekatan penyimpanan dan pengelolaan data yang berbeda sehingga kinerja pemasukan data yang dihasilkan dapat bervariasi, bergantung pada metode operasi dan konfigurasi pengujian yang digunakan (Ali et al., 2025; Kohi, 2022; Makcha dan Omer 2026 ; Sudarsono et al.2021). Oleh karena itu, perbandingan kinerja kedua sistem diperlukan untuk mengetahui perbedaan waktu eksekusi dalam skenario pemasukan data yang sama.

Salah satu faktor yang dapat memengaruhi kinerja pemasukan data adalah metode yang digunakan. Pemasukan data secara satu per satu (*one-by-one*) menjalankan operasi untuk setiap data, sedangkan pemasukan secara berkelompok (*batch insertion*) mengirimkan sejumlah data dalam satu proses Abdulqadir et al., (2019) ; MariaDB, n.d.-b; MongoDB, n.d.). Pada metode *batch*, ukuran kelompok data dapat memengaruhi waktu yang dibutuhkan untuk menyelesaikan pemasukan seluruh data. Namun, perbedaan karakteristik MariaDB dan MongoDB membuat hasil pengujian perlu diukur secara langsung agar perbandingan didasarkan pada data eksperimen, bukan hanya pada karakteristik masing-masing sistem. Penelitian terdahulu yang membandingkan MariaDB dan MongoDB pada pemasukan data juga menunjukkan hasil yang berbeda arah Kohi, (2022); Sudarsono et al. (2021), dan metode pemasukan tidak diuji sebagai variabel pada keduanya.

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja operasi *INSERT* pada MariaDB dan MongoDB berdasarkan metode pemasukan data dan variasi ukuran *batch*. Pengujian dilakukan terhadap 10.000 data dengan lima kali pengulangan untuk setiap skenario. Metode yang diuji meliputi pemasukan data secara *one-by-one* serta pemasukan data secara *batch* dengan ukuran 100, 500, dan 1.000 data. Kinerja kedua sistem dievaluasi berdasarkan waktu eksekusi yang diukur selama proses pemasukan data. Hasil penelitian diharapkan dapat memberikan gambaran empiris mengenai perbedaan kinerja MariaDB dan MongoDB dalam menangani operasi *INSERT* pada skenario pengujian yang telah ditentukan.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

KAJIAN LITERATUR/TINJAUAN PUSTAKA

Sistem Basis Data Relasional dan MariaDB

Database Management System (DBMS) merupakan perangkat lunak yang digunakan untuk menyimpan, mengelola, dan mengakses data. Sistem basis data dapat dibedakan menjadi sistem relasional dan nonrelasional. Pada sistem relasional, data diorganisasikan ke dalam tabel yang terdiri atas baris dan kolom, sedangkan sistem nonrelasional menyediakan model penyimpanan lain, termasuk model dokumen. Perbedaan model tersebut memengaruhi cara data direpresentasikan dan diakses, sehingga evaluasi kinerja perlu mempertimbangkan jenis operasi dan karakteristik beban kerja yang digunakan.

MariaDB merupakan sistem manajemen basis data relasional yang menggunakan SQL (*Structured Query Language*) untuk mengelola data yang disimpan dalam tabel berskema. Operasi INSERT digunakan untuk menambahkan baris baru ke dalam tabel. Dokumentasi resmi MariaDB menyebutkan bahwa satu pernyataan INSERT dapat memuat lebih dari satu daftar nilai sehingga beberapa baris dimasukkan sekaligus, dan bahwa cara ini merupakan optimasi kecepatan apabila banyak baris dimasukkan (MariaDB, n.d.).

Kinerja pemasukan data pada MariaDB tidak hanya berkaitan dengan jumlah data, tetapi juga dapat dipengaruhi oleh konfigurasi sistem, mesin penyimpanan, indeks, pengelolaan transaksi, dan metode pengiriman data. Sebagai contoh, Liyanage et al., (2026) melaporkan perbedaan kinerja yang berkaitan dengan desain mesin penyimpanan ketika membandingkan InnoDB dan MyRocks pada MariaDB. Oleh karena itu, waktu eksekusi hasil eksperimen perlu ditafsirkan berdasarkan lingkungan dan prosedur pengujian yang digunakan.

Basis Data NoSQL Berorientasi Dokumen dan MongoDB

MongoDB merupakan sistem basis data NoSQL yang menggunakan model dokumen. Data dikelompokkan ke dalam *collection* dan disimpan sebagai dokumen BSON, sehingga dokumen dapat merepresentasikan data dengan struktur yang lebih fleksibel dibandingkan tabel relasional.

MongoDB menyediakan metode untuk memasukkan satu dokumen maupun beberapa dokumen sekaligus. Metode insertMany() memasukkan seluruh dokumen dalam sebuah array ke dalam *collection* dan menyediakan opsi ordered; apabila pemasukan diatur tidak berurutan, dokumen dapat disusun ulang oleh server untuk meningkatkan kinerja. Metode ini juga menerima pengaturan *write concern* (MongoDB, n.d.).

Pemasukan beberapa dokumen dalam satu permintaan berpotensi mengurangi frekuensi komunikasi antara aplikasi dan basis data dibandingkan pengiriman setiap dokumen secara terpisah. Namun, besarnya pengaruh tersebut perlu diukur karena dapat dipengaruhi oleh ukuran data, pengaturan penulisan, dan kondisi lingkungan pengujian.

Operasi INSERT dan Metode Pemasukan Data

Operasi *INSERT* merupakan operasi untuk menambahkan data baru ke dalam basis data. Pada MariaDB, operasi ini menambahkan baris ke dalam tabel, sedangkan pada MongoDB operasi pemasukan menambahkan dokumen ke dalam *collection*. Meskipun tujuannya serupa, kedua sistem menggunakan model penyimpanan dan antarmuka operasi yang berbeda.

Pendekatan pertama adalah metode satu per satu (*one-by-one*), yaitu ketika setiap data dikirimkan melalui operasi terpisah. Pendekatan kedua adalah metode *batch*, yaitu ketika beberapa data dikelompokkan dan dikirimkan dalam satu permintaan atau satu kelompok operasi. Kedua sistem menyediakan mekanisme untuk pendekatan kedua, yaitu daftar nilai majemuk pada *INSERT* di MariaDB (MariaDB, n.d.-b) dan insertMany() di MongoDB (MongoDB, n.d.). Metode *batch* berpotensi mengurangi frekuensi pemanggilan operasi dan biaya komunikasi, tetapi besarnya manfaat bergantung pada cara implementasi, ukuran *batch*, dan karakteristik sistem yang diuji.

Ali dan Arif (2019) membahas penggunaan pendekatan pemasukan data pada basis data SQL dan NoSQL, termasuk *bulk insert* dan *batch insert*. Penelitian tersebut relevan untuk menjelaskan mengapa metode pemasukan perlu diperhitungkan dalam evaluasi kinerja.

Dalam penelitian ini, metode pemasukan data dibedakan menjadi pemasukan satu per satu dan pemasukan *batch* dengan ukuran 100, 500, dan 1.000 data. Variasi ukuran *batch* digunakan untuk mengamati perubahan waktu eksekusi ketika jumlah data per kelompok ditingkatkan. Perbandingan dilakukan pada jumlah data yang sama untuk setiap skenario, sehingga analisis difokuskan pada pengaruh metode pemasukan dan ukuran *batch* pada MariaDB dan MongoDB. Rincian implementasi pada masing-masing sistem dijelaskan pada bagian Metode.

Pengukuran Kinerja Basis Data

Kinerja basis data dapat dievaluasi menggunakan beberapa indikator, antara lain waktu eksekusi, *throughput*, latensi, dan penggunaan sumber daya. Indikator yang dipilih harus sesuai dengan tujuan penelitian dan prosedur pengukuran yang benar-benar dilakukan. Makcha & Omer, (2026), misalnya, menilai waktu pemasukan, latensi kueri, waktu pembaruan dan penghapusan, serta penggunaan CPU dan memori pada MongoDB dan SQL Server dalam lingkungan terkontrol berbasis Docker, sedangkan Mostafa et al., (2022) merancang SciTS untuk mengukur

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

kemampuan ingesti data dan latensi kueri pada basis data deret waktu.

Waktu eksekusi menunjukkan durasi yang diperlukan untuk menyelesaikan suatu operasi. Dalam penelitian ini, waktu eksekusi operasi INSERT diukur dalam detik dan digunakan sebagai indikator utama perbandingan. Nilai yang lebih rendah menunjukkan bahwa operasi selesai dalam waktu yang lebih singkat pada skenario pengujian tersebut. *Throughput* menggambarkan jumlah data yang diproses per satuan waktu; apabila jumlah data dan waktu eksekusi tercatat dengan batas pengukuran yang konsisten, *throughput* dapat dihitung sebagai jumlah data dibagi waktu eksekusi. Metrik ini memberikan perspektif tambahan, tetapi tidak merupakan pengukuran independen karena diturunkan dari waktu eksekusi.

Hasil evaluasi juga bergantung pada kondisi pengujian. Weißbach et al., (2020) serta Weißbach & Springer, (2022) menunjukkan bahwa kinerja basis data berbeda menurut pola akses data dan kombinasi perangkat lunak dalam lingkungan pemrosesan aliran data. Ali et al., (2025) membandingkan PostgreSQL, MongoDB, dan MySQL pada ukuran data 100 hingga 100.000 baris dengan operasi yang dijalankan melalui lapisan API Hasura pada lingkungan cloud; kondisi tersebut membuat hasilnya tidak sepenuhnya terisolasi pada operasi basis data itu sendiri.

Berdasarkan pertimbangan tersebut, penelitian ini menggunakan waktu eksekusi operasi INSERT dalam satuan detik sebagai indikator utama. Setiap skenario diukur lima kali; nilai rata-rata digunakan untuk perbandingan, dan variasi antar-pengulangan digunakan untuk menilai konsistensi pengukuran.

Penelitian Terdahulu dan Posisi Penelitian

Penelitian terdahulu telah membandingkan kinerja basis data relasional dan NoSQL dengan sistem, ukuran data, dan beban kerja yang beragam. Ringkasan penelitian yang paling berkaitan disajikan pada Tabel 1.

Tabel 1. Ringkasan Penelitian Terdahulu yang Berkaitan

Penelitian	Sistemyang Dibandingkan	Operasi dan Skala Data	Temuan yang Dilaporkan
Abdulqadir et al (2019)	MySQL dan MongoDB	Pemasukan (*bulk insert* pada SQL dan *batch insert* pada NoSQL) dan pembaruan; hingga 3.000.000 data	Pemasukan *batch* pada MongoDB lebih cepat daripada *bulk insert* pada MySQL (3.000.000 data: 67,87 detik berbanding 606,53 detik)
Pandey (2020)	MongoDB dan MySQL	Beban kerja YCSB pada lingkungan *cloud*	Perbandingan kinerja kedua sistem menggunakan *benchmark* YCSB
Sudarsono et al. (2021)	MariaDB dan MongoDB	*INSERT, UPDATE*, *DELETE*, *SELECT*; 100 sampai 10.000 data	MongoDB memiliki waktu respons yang jauh lebih baik pada *INSERT*, *UPDATE*, dan *DELETE* untuk satu tabel; *SELECT* dengan kriteria, terutama dua kriteria, lebih baik pada SQL
Kohi (2022)	MariaDB, MongoDB, dan CockroachDB	Kinerja, indeks, transaksi, *trigger*, operasi CRUD, dan data berukuran besar; 1 sampai 1.000.000 data	MariaDB rata-rata lebih cepat daripada MongoDB dan CockroachDB pada pemasukan data 1 sampai 1.000.000 data
Ali et al. (2025)	PostgreSQL, MongoDB, dan MySQL	*Insert*, *select*, *delete*; 100 sampai 100.000 baris	PostgreSQL unggul secara keseluruhan; MongoDB fleksibel tetapi kinerjanya menurun pada volume data yang lebih besar
Liyanage et al. (2026)	MongoDB dan MariaDB (InnoDB dan MyRocks)	YCSB yang diperluas dengan operasi perpanjangan nilai (*extend*); *throughput* dan latensi	Perbedaan kinerja yang signifikan berkaitan dengan desain mesin penyimpanan dan penanganan nilai berukuran bervariasi
Makcha dan Omer (2026)	MongoDB dan SQL Server	Pemasukan data, kueri, pembaruan, penghapusan, CPU, dan memori; 10.000 sampai 5.000.000 rekaman	MongoDB lebih baik pada pemasukan data; SQL Server unggul pada kueri terstruktur berbasis tipe data tertentu

Temuan pada Tabel 1 menunjukkan hasil yang beragam. Abdulqadir et al., (2019), Sudarsono et al. (2021), serta Makcha & Omer, (2026) melaporkan keunggulan MongoDB pada pemasukan data terhadap MySQL, MariaDB, dan SQL Server, sedangkan Kohi, (2022) melaporkan MariaDB secara rata-rata lebih cepat daripada MongoDB, dan Ali et al., (2025) melaporkan bahwa PostgreSQL secara keseluruhan lebih unggul daripada MongoDB dan bahwa kinerja

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

MongoDB menurun pada volume data yang lebih besar. Perbedaan ini menegaskan bahwa hasil perbandingan bergantung pada sistem pembanding, ukuran data, dan lingkungan pengujian, sehingga tidak dapat digeneralisasi ke semua konfigurasi.

Perbandingan MariaDB dan MongoDB bukan hal yang sepenuhnya baru. Sudarsono et al. (2021), membandingkan keduanya pada beberapa operasi basis data, dan Kohi, (2022) mengevaluasinya dalam cakupan yang lebih luas. Berdasarkan telaah penulis, metode pemasukan data (satu per satu dan *batch*) serta ukuran *batch* tidak disebutkan sebagai variabel pengujian pada kedua penelitian tersebut. Posisi penelitian ini adalah fokus eksperimen yang lebih khusus pada aspek tersebut.

Penelitian ini mengevaluasi waktu eksekusi operasi *INSERT* pada MariaDB dan MongoDB dengan membandingkan metode pemasukan satu per satu dan metode *batch* berukuran 100, 500, serta 1.000 data. Setiap skenario menggunakan 10.000 data dan diulang lima kali. Kesimpulan dibatasi pada skenario dan konfigurasi eksperimen tersebut, sehingga tidak dimaksudkan untuk menyatakan bahwa salah satu DBMS selalu lebih unggul pada semua kondisi.

METODE PENELITIAN

Jenis Penelitian

Penelitian ini menggunakan metode eksperimen kuantitatif untuk menganalisis perbandingan kinerja operasi pemasukan data (*INSERT*) pada MariaDB dan MongoDB. Pengujian dilakukan dengan membandingkan dua metode pemasukan, yaitu metode satu per satu (*one-by-one*) dan metode berkelompok (*batch*). Pada metode *batch*, ukuran kelompok yang diuji adalah 100, 500, dan 1.000 data.

Variabel bebas penelitian ini adalah sistem basis data (MariaDB dan MongoDB), metode pemasukan data (*one-by-one* dan *batch*), serta ukuran *batch* (100, 500, dan 1.000 data) yang hanya berlaku pada metode *batch*. Variabel terikatnya adalah waktu eksekusi pemasukan 10.000 data yang dinyatakan dalam detik.

Kinerja dalam penelitian ini diukur berdasarkan waktu eksekusi yang diperlukan untuk memasukkan 10.000 data pada setiap skenario pengujian. Pengukuran dilakukan sebanyak lima kali untuk setiap skenario. Data hasil pengujian dianalisis menggunakan statistik deskriptif berupa rata-rata waktu eksekusi dan simpangan baku (*standard deviation*) untuk membandingkan waktu pemasukan serta variasi hasil pengukuran pada kedua sistem basis data.

Lingkungan Pengujian

Eksperimen dijalankan pada laptop dengan sistem operasi Windows 11, prosesor Intel Core i5-1235U (12 prosesor logis), RAM 8 GB, dan penyimpanan SSD. Kedua sistem basis data dijalankan secara lokal pada laptop yang sama. Bahasa pemrograman yang digunakan adalah Python versi 3.13.2 dengan lingkungan pengembangan PyCharm Community Edition 2024.3.2, dan seluruh pengukuran dilakukan dalam satu sesi pada 25 Oktober 2026.

Sistem basis data yang dibandingkan adalah MariaDB versi 10.4.32 (distribusi mariadb.org yang disertakan dalam paket XAMPP) dan MongoDB versi 8.0.32. MariaDB diakses melalui alamat *127.0.0.1* pada port 3306, sedangkan MongoDB dijalankan sebagai layanan lokal dan diakses melalui alamat *mongodb://localhost:27017*.

Program Python digunakan untuk menjalankan operasi pemasukan dan mencatat waktu eksekusi setiap skenario. Koneksi ke MariaDB menggunakan pustaka *mysql-connector-python* versi 26.7.0, sedangkan koneksi ke MongoDB menggunakan PyMongo versi 4.18.3. Pengujian *batch* MariaDB menggunakan mekanisme *executemany()*, sedangkan pengujian *batch* MongoDB menggunakan *insert_many()*. Pada metode *one-by-one*, setiap data dimasukkan melalui satu pemanggilan *execute()* pada MariaDB dan satu pemanggilan *insert_one()* pada MongoDB.

Pada MariaDB, *autocommit* dinonaktifkan dan satu *commit* dilakukan setelah seluruh 10.000 data dimasukkan; waktu *commit* termasuk dalam pengukuran. Pengaturan server MariaDB yang tercatat saat pengujian antara lain *innodb_flush_log_at_trx_commit* bernilai 1 dan *innodb_buffer_pool_size* sebesar 16 MB. Pada MongoDB, *write concern* dan parameter *ordered* pada *insert_many()* tidak diubah dari nilai bawaan.

Dataset dan Skenario Pengujian

Dataset penelitian berupa data sintetis sebanyak 10.000 data yang dibuat menggunakan program Python. Setiap data memiliki tiga atribut utama, yaitu identitas, nama, dan nilai. Data nama dibentuk dengan pola *Mahasiswa_i*, sedangkan nilai dibentuk dari sisa bagi indeks data dengan 100, dengan *i* adalah nomor urut data. Pada MongoDB, atribut identitas disimpan pada *field _id* dengan nilai berurutan dari 1 sampai 10.000. Pada MariaDB, data disimpan dalam tabel dengan kolom *id* (*INT*, kunci utama), *nama* (*VARCHAR*(50)), dan *nilai* (*INT*) menggunakan mesin penyimpanan InnoDB.

Dataset yang sepadan, baik jumlah maupun isinya, digunakan pada kedua sistem basis data. Pengujian dibagi menjadi empat skenario untuk setiap sistem, sebagaimana dirangkum pada Tabel 2.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

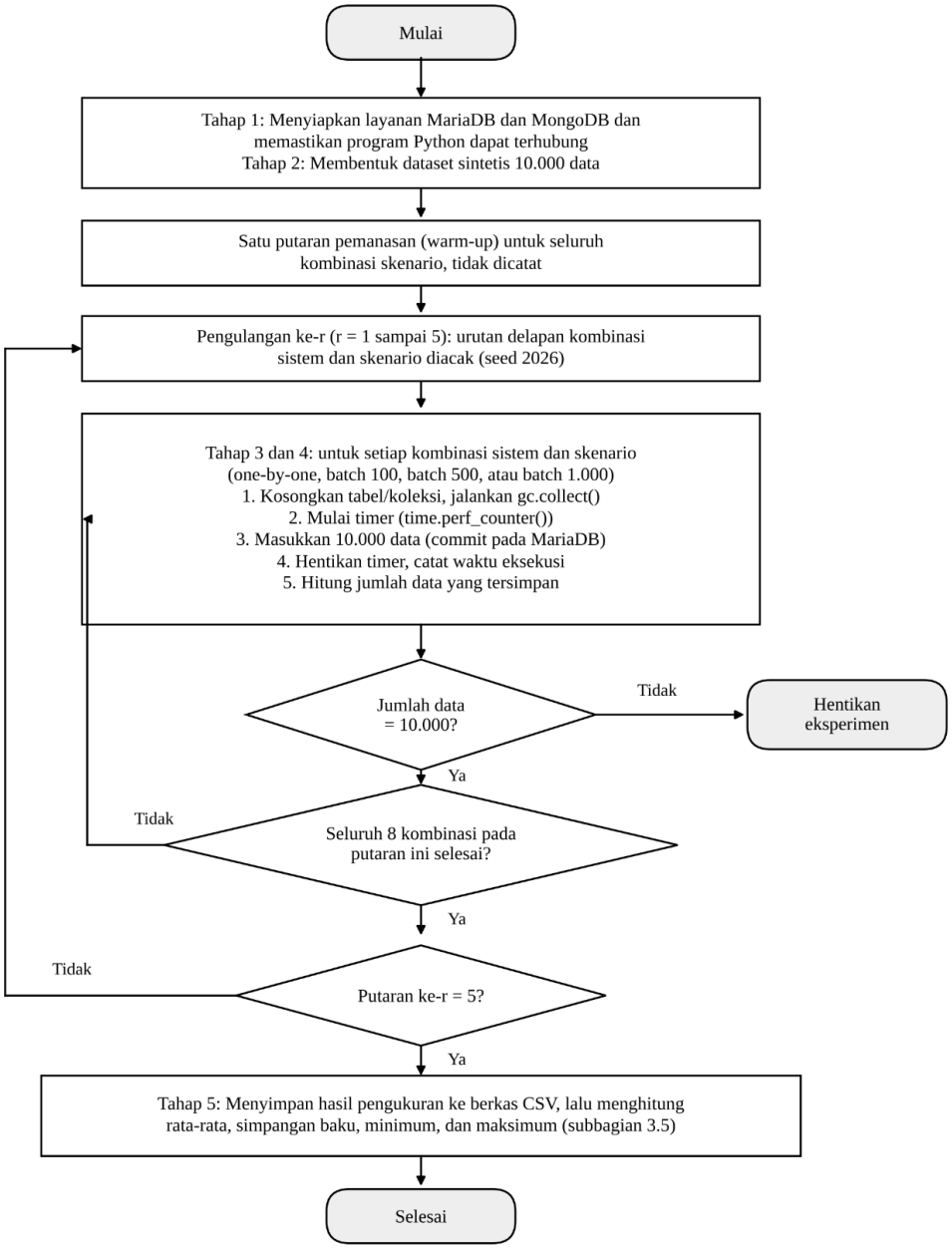
Tabel 2. Skenario Pengujian pada Setiap Sistem Basis Data

Skenario	Metode Pemasukan	Ukuran Kelompok (data)	Jumlah Pemanggilan Operasi
One-by-one	Satu per satu	1	10.000
Batch 100	Berkelompok	100	100
Batch 500	Berkelompok	500	20
Batch 1.000	Berkelompok	1.000	10

Pada metode *batch*, ukuran kelompok menentukan jumlah pemanggilan operasi yang diperlukan untuk memproses seluruh dataset. Setiap skenario pada masing-masing sistem basis data diulang sebanyak lima kali. Oleh karena itu, rancangan pengujian terdiri atas delapan kombinasi skenario, yaitu empat skenario pada MariaDB dan empat skenario pada MongoDB, dengan total 40 pengukuran waktu eksekusi.

Prosedur Pengujian

Pengujian dilaksanakan melalui lima tahap yang diringkas dalam diagram alir pada Gambar 1.



Gambar 1. Diagram Alir Prosedur Pengujian

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

Tahap pertama adalah menyiapkan layanan MariaDB dan MongoDB serta memastikan bahwa program Python dapat terhubung dengan kedua sistem basis data. Tahap kedua adalah membentuk dataset sintetis sebanyak 10.000 data dengan struktur yang telah ditentukan.

Tahap ketiga adalah menjalankan pengujian metode *one-by-one* dan *batch* pada masing-masing sistem basis data. Pada metode *one-by-one*, operasi pemasukan dilakukan secara individual untuk setiap data. Pada metode *batch*, data dibagi menjadi kelompok sesuai ukuran 100, 500, atau 1.000 data, dan proses dilanjutkan sampai seluruh 10.000 data selesai diproses.

Tahap keempat adalah mencatat waktu eksekusi setiap pengulangan menggunakan fungsi *time.perf_counter()* pada Python. Pengukuran dimulai tepat sebelum pemanggilan operasi pemasukan yang pertama dan diakhiri setelah pemanggilan yang terakhir. Pada MariaDB, *commit* termasuk dalam pengukuran, dan pada metode *batch* pemecahan data menjadi kelompok berada di dalam pengukuran pada kedua sistem. Pembentukan dataset, pembukaan koneksi, dan pengosongan data tidak dimasukkan ke dalam pengukuran. Sebelum setiap pengukuran, tabel MariaDB dikosongkan dengan *TRUNCATE TABLE*, koleksi MongoDB dikosongkan dengan *delete_many({})*, dan pengumpulan sampah memori Python (*gc.collect()*) dijalankan. Setelah pengukuran waktu berakhir, jumlah baris atau dokumen dihitung untuk memastikan bahwa seluruh 10.000 data telah tersimpan. Pemeriksaan ini tidak menambah durasi yang dicatat, dan eksperimen dihentikan apabila jumlahnya tidak sesuai. Sebelum pengukuran dimulai, satu putaran pemanasan (*warm-up*) untuk seluruh kombinasi skenario dijalankan tanpa dicatat. Pada setiap putaran pengulangan, urutan kedelapan kombinasi skenario diacak menggunakan *seed* 2026.

Tahap kelima adalah menyimpan hasil pengukuran ke dalam berkas CSV. Hasil dari berbagai skenario kemudian digabungkan untuk memudahkan pengolahan dan perbandingan waktu eksekusi.

Teknik Analisis Data

Analisis dilakukan menggunakan statistik deskriptif dengan dua ukuran utama, yaitu rata-rata waktu eksekusi dan simpangan baku sampel, yang dilengkapi nilai minimum dan maksimum dari lima pengulangan. Seluruhnya dihitung menggunakan modul *statistics* dan fungsi bawaan Python (*mean*, *stdev*, *min*, dan *max*). Rata-rata digunakan untuk menggambarkan waktu eksekusi pada setiap skenario, sedangkan simpangan baku serta nilai minimum dan maksimum digunakan untuk mengetahui tingkat variasi waktu eksekusi di antara lima pengulangan.
Rata-rata waktu eksekusi dihitung dengan rumus berikut.

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

dengan $\overline{x}$ adalah rata-rata waktu eksekusi, x_i adalah waktu eksekusi pada pengulangan ke-i, dan n adalah jumlah pengulangan, yaitu lima.
Simpangan baku sampel dihitung dengan rumus berikut.

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \tag{2}$$

dengan s adalah simpangan baku sampel, sedangkan x_i, $\overline{x}$, dan n sebagaimana didefinisikan pada (1).

Nilai rata-rata dan simpangan baku dihitung untuk setiap kombinasi sistem basis data dan metode pemasukan. Hasilnya digunakan untuk membandingkan waktu eksekusi MariaDB dan MongoDB pada metode *one-by-one* serta pada setiap ukuran *batch* yang diuji. Skenario dengan rata-rata waktu eksekusi lebih rendah diidentifikasi sebagai skenario yang menyelesaikan pemasukan lebih cepat dalam lingkungan eksperimen ini, sedangkan simpangan baku digunakan sebagai informasi pendamping untuk menilai variasi hasil antarpengulangan. Selisih antarskenario dinyatakan sebagai rasio dan persentase penurunan rata-rata waktu eksekusi, dan dua skenario dinilai tidak dapat dibedakan apabila rentang minimum sampai maksimum kelima pengulangannya saling tumpang tindih.

Batasan Penelitian

Analisis dibatasi pada waktu eksekusi operasi pemasukan data dan tidak digunakan untuk menyimpulkan perbedaan penggunaan CPU, memori, atau metrik kinerja lain yang tidak diukur dalam eksperimen. Hasil penelitian merepresentasikan kinerja kedua sistem berdasarkan implementasi, pustaka, dan lingkungan pengujian yang digunakan, bukan ukuran mutlak kinerja seluruh konfigurasi MariaDB dan MongoDB.

HASIL DAN PEMBAHASAN

Hasil Pengukuran Waktu Eksekusi

Seluruh 40 pengukuran berhasil dilaksanakan, dan pada setiap pengukuran jumlah data yang tersimpan sama dengan 10.000. Rata-rata waktu eksekusi, simpangan baku sampel, serta nilai minimum dan maksimum dari lima pengulangan disajikan pada Tabel 3 dan divisualisasikan pada Gambar 1, sedangkan hasil setiap pengulangan disajikan pada Lampiran (Tabel L1).

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

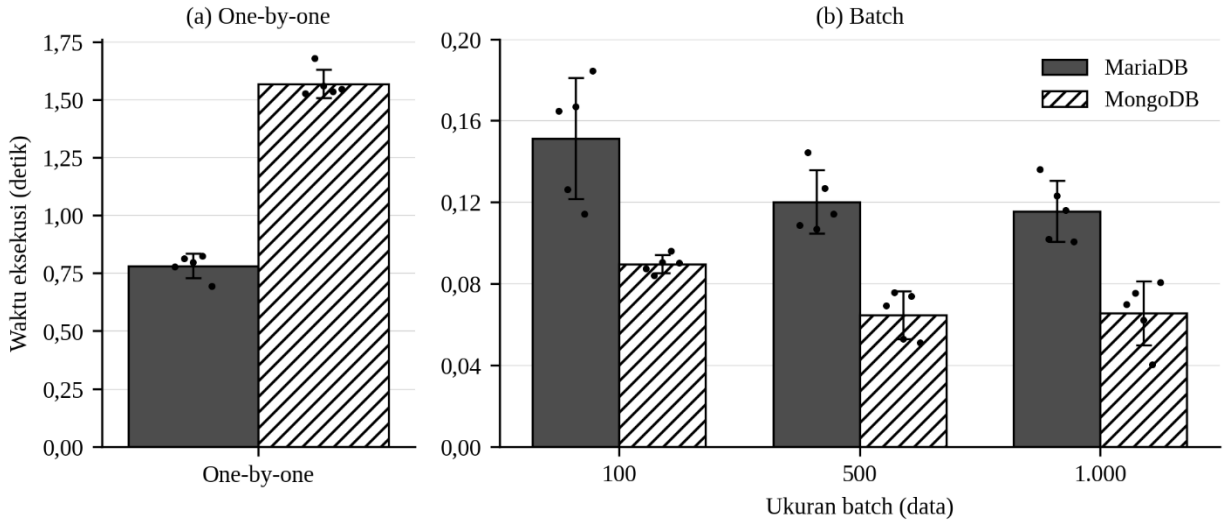
Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

Tabel 3. Waktu Eksekusi Operasi *INSERT* 10.000 Data pada Setiap Skenario (n = 5)

Sistem	Skenario	Rata-rata (detik)	Simpangan Baku (detik)	Minimum (detik)	Maksimum (detik)
MariaDB	*One-by-one*	0,7820	0,0518	0,6945	0,8254
MariaDB	*Batch* 100	0,1513	0,0297	0,1142	0,1843
MariaDB	*Batch* 500	0,1201	0,0156	0,1067	0,1442
MariaDB	*Batch* 1.000	0,1155	0,0149	0,1007	0,1361
MongoDB	*One-by-one*	1,5691	0,0622	1,5260	1,6781
MongoDB	*Batch* 100	0,0896	0,0044	0,0842	0,0961
MongoDB	*Batch* 500	0,0645	0,0118	0,0510	0,0758
MongoDB	*Batch* 1.000	0,0656	0,0157	0,0404	0,0806

Tabel 3 menyajikan ringkasan waktu eksekusi untuk delapan kombinasi sistem basis data dan skenario pemasukan. Setiap baris memuat rata-rata dari lima pengulangan, simpangan baku sampel sebagai ukuran variasi antarpengulangan, serta nilai minimum dan maksimum yang tercatat; seluruhnya dinyatakan dalam detik, dan nilai yang lebih rendah berarti pemasukan 10.000 data selesai lebih cepat. Rata-rata terendah pada MariaDB terdapat pada *batch* 1.000 (0,1155 detik) dan tertinggi pada *one-by-one* (0,7820 detik), sedangkan pada MongoDB rata-rata terendah terdapat pada *batch* 500 (0,0645 detik) dan tertinggi pada *one-by-one* (1,5691 detik). Simpangan baku terkecil terdapat pada MongoDB *batch* 100 (0,0044 detik) dan terbesar pada MongoDB *one-by-one* (0,0622 detik). Perbandingan antarskenario dijelaskan pada paragraf berikut.



Gambar 2. Rata-rata waktu eksekusi *INSERT* 10.000 data pada MariaDB dan MongoDB: (a) metode *one-by-one* dan (b) metode *batch* berukuran 100, 500, dan 1.000 data. Batang menunjukkan rata-rata, garis vertikal menunjukkan simpangan baku sampel, dan titik hitam menunjukkan hasil setiap pengulangan (n = 5).

Gambar 2 memperlihatkan pola yang sama dengan Tabel 3. Pada panel (a), batang MongoDB lebih tinggi daripada MariaDB pada metode *one-by-one*, sedangkan pada panel (b), batang MongoDB lebih rendah daripada MariaDB pada ketiga ukuran *batch*. Garis vertikal menunjukkan simpangan baku, dan titik menunjukkan hasil setiap pengulangan sehingga sebaran data dapat dilihat langsung.

Pada kedua sistem, metode *batch* menghasilkan waktu eksekusi yang jauh lebih rendah daripada metode *one-by-one*. Pada ukuran *batch* 1.000, rata-rata waktu eksekusi turun sekitar 85,2% pada MariaDB (dari 0,7820 menjadi 0,1155 detik) dan sekitar 95,8% pada MongoDB (dari 1,5691 menjadi 0,0656 detik).

Peningkatan ukuran *batch* dari 100 ke 500 menurunkan rata-rata waktu eksekusi pada kedua sistem, yaitu sekitar 20,6% pada MariaDB dan 28,0% pada MongoDB. Namun, rentang hasil *batch* 100 dan *batch* 500 pada MariaDB saling tumpang tindih sehingga penurunan tersebut perlu ditafsirkan dengan hati-hati. Dari *batch* 500 ke *batch* 1.000, perubahan rata-rata sangat kecil dibandingkan simpangan baku: selisihnya 0,0046 detik pada MariaDB, sedangkan pada MongoDB rata-rata *batch* 1.000 (0,0656 detik) justru sedikit lebih tinggi daripada *batch* 500 (0,0645 detik). Pada kedua sistem, rentang hasil kedua ukuran tersebut saling tumpang tindih sehingga dengan lima pengulangan tidak dapat disimpulkan bahwa *batch* 1.000 lebih cepat daripada *batch* 500.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

Perbandingan antarsistem menunjukkan pola yang berbalik. Pada metode *one-by-one*, MariaDB lebih cepat (0,7820 detik dibandingkan 1,5691 detik), dan MongoDB membutuhkan waktu sekitar 2,0 kali lebih lama. Pada *batch* 100, 500, dan 1.000, MongoDB lebih cepat, dan rata-rata waktu MariaDB masing-masing sekitar 1,7; 1,9; dan 1,8 kali rata-rata waktu MongoDB. Pada keempat skenario tersebut, rentang minimum sampai maksimum kedua sistem tidak saling tumpang tindih.

Pada skenario berdurasi pendek, variasi hasil antarpengulangan relatif besar dibandingkan rata-ratanya. Sebagai contoh, hasil pengulangan MongoDB *batch* 1.000 berkisar antara 0,0404 dan 0,0806 detik, dan hasil MariaDB *batch* 100 berkisar antara 0,1142 dan 0,1843 detik.

Pembahasan

Penurunan waktu eksekusi pada metode *batch* sejalan dengan tujuan mekanismenya, yaitu mengurangi jumlah pemanggilan operasi dari 10.000 menjadi 100, 20, atau 10 pemanggilan (Tabel 2). Dokumentasi MariaDB menyebutkan bahwa memasukkan banyak baris dalam satu pernyataan *INSERT* merupakan optimasi kecepatan (MariaDB, n.d.-b), dan MongoDB menyediakan insertMany() untuk memasukkan banyak dokumen sekaligus (MongoDB, n.d.). Ali dan Arif (2019) juga menyatakan bahwa pemasukan data tunggal membutuhkan waktu lebih lama dan menurunkan kinerja dibandingkan *bulk insert* pada basis data SQL. Hasil penelitian ini memperlihatkan besarnya selisih waktu pada lingkungan yang diuji, tetapi penelitian ini tidak mengukur mekanisme internal yang menyebabkannya.

Tidak adanya perbaikan yang jelas dari *batch* 500 ke *batch* 1.000 mengindikasikan bahwa pada rentang yang diuji sebagian besar penghematan waktu sudah diperoleh pada ukuran *batch* menengah. Penyebab pastinya tidak diuji dalam penelitian ini, dan ukuran *batch* di atas 1.000 tidak termasuk dalam pengujian.

Berbaliknya urutan kinerja antara metode *one-by-one* dan *batch* perlu ditafsirkan sesuai rancangan eksperimen. Pada MariaDB, 10.000 pernyataan *INSERT* dijalankan dalam satu transaksi dengan satu *commit* di akhir, sedangkan pada MongoDB setiap pemanggilan *insert_one()* dijalankan sebagai operasi terpisah dengan pengaturan write concern bawaan. Dokumentasi MariaDB menyebutkan bahwa sinkronisasi data ke disk pada akhir transaksi termasuk hal yang paling memakan waktu dalam pemasukan data, dan menyarankan agar banyak pemasukan berurutan dibungkus dalam satu transaksi untuk menghindari sinkronisasi penuh pada setiap baris (MariaDB, n.d.-a). Perbedaan perlakuan ini mungkin berkontribusi terhadap keunggulan MariaDB pada metode *one-by-one*. Namun, penelitian ini tidak menguji pengaruhnya secara terpisah, misalnya dengan menjalankan MariaDB menggunakan *commit* pada setiap pemanggilan, sehingga penjelasan tersebut berstatus dugaan.

Arah temuan pada metode *batch* sejalan dengan Ali dan Arif (2019), yang melaporkan bahwa pemasukan *batch* pada MongoDB lebih cepat daripada *bulk insert* pada MySQL (3.000.000 data: 67,87 detik berbanding 606,53 detik); dengan Sudarsono et al. (2021), yang melaporkan waktu respons MongoDB yang jauh lebih baik daripada MariaDB pada operasi *INSERT*, dan dengan Makcha dan Omer (2026), yang melaporkan MongoDB lebih baik pada pemasukan data dibandingkan SQL Server. Sebaliknya, hasil *one-by-one* pada penelitian ini berbeda arah dari temuan Sudarsono et al. (2021) dan searah dengan temuan Kohi (2022), yang melaporkan MariaDB secara rata-rata lebih cepat daripada MongoDB pada pemasukan 1 sampai 1.000.000 data, sedangkan Ali et al. (2025) melaporkan bahwa PostgreSQL secara keseluruhan lebih unggul daripada MongoDB. Metode pemasukan tidak diuji sebagai variabel pada Sudarsono et al. (2021) maupun Kohi (2022), sehingga perbedaan arah tersebut tidak dapat dikaitkan dengan metode pemasukan. Penyebab perbedaan tersebut tidak dapat dipastikan dari data yang tersedia, dan perbandingan langsung dibatasi oleh perbedaan sistem pembanding, ukuran data, dan lingkungan pengujian. Keragaman temuan ini menegaskan bahwa kesimpulan perbandingan basis data perlu dikaitkan dengan kondisi pengujiannya.

Beberapa keterbatasan perlu diperhatikan. Pertama, eksperimen dijalankan pada satu laptop dengan kedua basis data berjalan secara lokal, satu skema data sederhana dengan tiga atribut, dan 10.000 data per skenario. Kedua, hanya terdapat lima pengulangan untuk setiap skenario; pada skenario berdurasi puluhan milidetik variasi antarpengulangan relatif besar sehingga perbedaan kecil antarskenario tidak dapat disimpulkan. Ketiga, hasil dipengaruhi konfigurasi server yang tercatat saat pengujian, yaitu MariaDB versi 10.4.32 (bukan versi terbaru) dengan *innodb_buffer_pool_size* sebesar 16 MB dan *innodb_flush_log_at_trx_commit* bernilai 1, sehingga konfigurasi lain dapat memberikan hasil yang berbeda. Keempat, perlakuan *commit* pada MariaDB dan MongoDB tidak setara sebagaimana dijelaskan di atas. Kelima, hanya waktu eksekusi yang diukur, sehingga hasil ini tidak menggambarkan penggunaan CPU, memori, atau metrik kinerja lain.

Lampiran

Tabel L1 memuat waktu eksekusi setiap pengulangan yang menjadi dasar perhitungan Tabel 3. Nomor pengulangan menunjukkan putaran pengukuran, dan urutan kedelapan skenario pada setiap putaran diacak (subbagian 3.4).

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

Tabel L1. Waktu Eksekusi (detik) pada Setiap Pengulangan

Sistem	Skenario	Pengulangan 1	Pengulangan 2	Pengulangan 3	Pengulangan 4	Pengulangan 5
MariaDB	*One-by-one*	0,7794	0,8127	0,7979	0,8254	0,6945
MariaDB	*Batch* 100	0,1648	0,1261	0,1670	0,1142	0,1843
MariaDB	*Batch* 500	0,1087	0,1442	0,1067	0,1268	0,1141
MariaDB	*Batch* 1.000	0,1361	0,1019	0,1231	0,1159	0,1007
MongoDB	*One-by-one*	1,5260	1,6781	1,5601	1,5360	1,5456
MongoDB	*Batch* 100	0,0874	0,0842	0,0904	0,0961	0,0901
MongoDB	*Batch* 500	0,0692	0,0758	0,0528	0,0738	0,0510
MongoDB	*Batch* 1.000	0,0697	0,0754	0,0621	0,0404	0,0806

KESIMPULAN

Penelitian ini membandingkan waktu eksekusi operasi *INSERT* pada MariaDB versi 10.4.32 dan MongoDB versi 8.0.32 menggunakan metode pemasukan *one-by-one* dan *batch* berukuran 100, 500, dan 1.000 data, dengan 10.000 data pada setiap skenario dan lima pengulangan (40 pengukuran). Hasilnya menunjukkan tiga hal. Pertama, metode *batch* jauh lebih cepat daripada metode *one-by-one* pada kedua sistem: pada ukuran *batch* 1.000, rata-rata waktu eksekusi turun sekitar 85,2% pada MariaDB (dari 0,7820 menjadi 0,1155 detik) dan sekitar 95,8% pada MongoDB (dari 1,5691 menjadi 0,0656 detik). Kedua, urutan kinerja kedua sistem berbalik menurut metode pemasukan: pada metode *one-by-one* MariaDB lebih cepat (MongoDB membutuhkan waktu sekitar 2,0 kali lebih lama), sedangkan pada semua ukuran *batch* MongoDB lebih cepat (MariaDB membutuhkan waktu sekitar 1,7 hingga 1,9 kali waktu MongoDB), dan rentang hasil kedua sistem tidak saling tumpang tindih pada keempat skenario. Ketiga, peningkatan ukuran *batch* dari 100 ke 500 menurunkan rata-rata waktu eksekusi pada kedua sistem (pada MariaDB rentang hasilnya masih tumpang tindih), tetapi dari 500 ke 1.000 tidak terdapat perbedaan yang dapat dibedakan dari variasi pengukuran.

Secara praktis, hasil ini memberikan gambaran empiris bahwa pada lingkungan pengujian yang serupa dengan penelitian ini, pemasukan data secara berkelompok lebih menguntungkan daripada pemasukan satu per satu pada kedua sistem, dan ukuran *batch* 500 data sudah menghasilkan waktu yang tidak dapat dibedakan dari ukuran 1.000 data. Hasil ini tidak dimaksudkan untuk menyatakan bahwa salah satu sistem selalu lebih unggul; pemilihan sistem tetap perlu mempertimbangkan kebutuhan aplikasi yang tidak diukur dalam penelitian ini.

Kesimpulan ini terbatas pada satu laptop dengan kedua sistem berjalan secara lokal, skema data sederhana dengan tiga atribut, 10.000 data per skenario, lima pengulangan, dan waktu eksekusi sebagai satu-satunya metrik. Perlakuan *commit* pada kedua sistem tidak setara: MariaDB memakai satu *commit* di akhir, sedangkan MongoDB menerima setiap penulisan secara terpisah dengan pengaturan bawaan, sehingga keunggulan MariaDB pada metode *one-by-one* kemungkinan berkaitan dengan perbedaan tersebut. Konfigurasi server yang tercatat saat pengujian, misalnya *buffer pool* MariaDB sebesar 16 MB, juga dapat memengaruhi hasil.

Penelitian selanjutnya disarankan untuk menguji pengaruh pengaturan *commit* dan *write concern* secara terpisah, misalnya dengan menjalankan MariaDB menggunakan *commit* pada setiap pemanggilan; menambah jumlah pengulangan dan menggunakan uji statistik inferensial; memvariasikan jumlah data, ukuran *batch* di atas 1.000, indeks, dan kompleksitas skema; serta mengukur metrik lain seperti penggunaan CPU dan memori pada versi dan konfigurasi server yang berbeda.

REFERENSI

Ali & Arif, 2019. (2019). Improving The Performance of Big Data Databases Volume 4-Issue 2. *Kurdistan Journal of Applied Research (KJAR) Kurdistan Journal of Applied Research*, *4*(2). https://doi.org/10.24017/science.2019

Ali et al., 2025. (2025). A Comparative Evaluation of PostgreSQL, MongoDB, and MySQL for Ethiopian Migrant Data Management. *CEUR Workshop Proceedings*, *4077*.

Kohi, 2022. (2022). *Ein experimenteller Vergleich und eine Evaluation von MariaDB, MongoDB und CockroachDB*. https://reposit.haw-hamburg.de/handle/20.500.12738/16250%0Ahttps://reposit.haw-hamburg.de/bitstream/20.500.12738/16250/1/BA_Ein experimenteller Vergleich und eine Evaluation von MariaDB%2C MongoDB und CockroachDB_geschwärzt.pdf

Liyanage, D., Pandey, S., Goldstein, J., Cahill, M., Dey, A., Fekete, A., & Röhm, U. (2026). A Benchmark for Databases with Varying Value Lengths. *Lecture Notes in Computer Science*, *16261 LNCS*, 1–19. https://doi.org/10.1007/978-3-032-18070-4_1

Makcha dan Omer (2026). (2026). Experimental Performance Evaluation of Mongodb and Sql Server Under Large-Scale Workloads. *Electronic Journal of University of Aden for Basic and Applied Sciences*, *7*(1), 46–56. https://doi.org/10.47372/ejua-ba.2026.1.498

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 3, **Number** 3, Nopember 2026
https://doi.org/10.62671/suliwa.v3i3.372

MariaDB. (n.d.-a). *How to Quickly Insert Data Into MariaDB*. MariaDB Documentation. Diakses 3 November 2026, dari https://mariadb.com/docs/server/ha-and-performance/optimization-and-tuning/query-optimizations/how-to-quickly-insert-data-into-mariadb

MariaDB. (n.d.-b). *INSERT*. MariaDB Documentation. Diakses 3 November 2026, dari https://mariadb.com/docs/server/reference/sql-statements/data-manipulation/inserting-loading-data/insert

MongoDB, Inc. (n.d.). *db.collection.insertMany()*. MongoDB Manual. Diakses 3 November 2026, dari https://www.mongodb.com/docs/manual/reference/method/db.collection.insertMany/

M. et al. (2022). SciTS: A Benchmark for Time-Series Databases in Scientific Experiments and Industrial Internet of Things. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3538712.3538723

Sudarsono et al. (2021). (2021). Perbandingan Performansi Antara Mongodb Dan Mariadb. *Jurnal Informatika Komputer, Bisnis Dan Manajemen*, *19*(2), 1–11. https://doi.org/10.61805/fahma.v19i2.54

Weißbach dan Springer (2022). (2020). Performance analysis of continuous binary data processing using distributed databases within stream processing environments. *CLOSER 2020 - Proceedings of the 10th International Conference on Cloud Computing and Services Science*, (Closer), 138–149. https://doi.org/10.5220/0009413301380149

Weißbach et al. (2020). (2022). Performance of Databases Used in Data Stream Processing Environments. *International Conference on Cloud Computing and Services Science, CLOSER - Proceedings*, (Closer), 15–26. https://doi.org/10.5220/0011018300003200

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.