

## Perbandingan Performa MySQL dan MongoDB pada Sistem Informasi Pengelolaan Air Limbah di Bali

I Wayan Adi Wiratama<sup>\*1</sup>, I Nyoman Triadi Wiguna<sup>2</sup>, I Gusti Agung Istri Pradnya Prameswari<sup>3</sup>,  
I Made Agus Oka Gunawan<sup>4</sup>, Gede Indrawan<sup>5</sup>

Universitas Pendidikan Ganesha

e-mail: <sup>\*1</sup>adi.wiratama@student.undiksha.ac.id, <sup>2</sup>triadi@student.undiksha.ac.id,

<sup>3</sup>agung.istri@student.undiksha.ac.id, <sup>4</sup>agusokagunawan@gmail.com,

<sup>5</sup>gindrawan@undiksha.ac.id

### Abstrak

Penelitian ini bertujuan untuk menganalisis perbandingan performa API yang menggunakan basis data relasional MySQL dan basis data NoSQL MongoDB dalam konteks Sistem Informasi Pengelolaan Air Limbah (DSDP) milik Pemerintah Provinsi Bali. API yang diuji menampilkan data tagihan atau invoice yang belum terbayar untuk satu pelanggan, yang memerlukan filtering dan join dengan tabel lain. Saat ini, sistem menghadapi tantangan performa karena jumlah data yang mencapai 2 juta baris dan terus bertambah, yang menyebabkan waktu respons semakin lambat saat menjalankan query SQL. Untuk meminimalkan masalah ini, dilakukan evaluasi terhadap skema penyimpanan data alternatif, termasuk menghindari join dan memanfaatkan NoSQL. Pengujian menggunakan load testing berbasis API dengan Apache JMeter untuk mensimulasikan akses pengguna secara simultan dengan variasi beban. Metrik kinerja seperti waktu respons, throughput, dan error rate digunakan untuk mengevaluasi performa kedua database. Hasil pengujian menunjukkan MongoDB memiliki performa yang lebih unggul dengan response time 49.35% lebih cepat, peningkatan throughput rata-rata 54.5%, dan tingkat error yang lebih rendah (0.67%) dibandingkan MySQL (1.23%) pada beban puncak. Penggunaan resource juga lebih efisien dengan utilisasi CPU 15.7% lebih rendah. Penelitian selanjutnya dapat berfokus pada optimasi performa MongoDB dalam lingkungan terdistribusi dan implementasi strategi backup yang efisien.

**Kata kunci:** MySQL, MongoDB, Load Testing, Apache JMeter, API Tagihan.

### Abstract

This study aims to analyze the performance comparison of APIs using the relational database MySQL and the NoSQL database MongoDB within the context of the Wastewater Management Information System (DSDP) of the Provincial Government of Bali. The tested API displays unpaid invoices for a single customer, which requires filtering and joining with other tables. Currently, the system faces performance challenges as the data volume reaches 2 million rows and continues to grow, causing slower response times when executing SQL queries. To mitigate this issue, an evaluation of alternative data storage schemes is conducted, including avoiding joins and leveraging NoSQL. The testing uses API-based load testing with Apache JMeter to simulate simultaneous user access with varying loads. Performance metrics such as response time, throughput, and error rate are used to evaluate the performance of both databases. Testing results show that MongoDB demonstrates superior performance with 49.35% faster response time, 54.5% average throughput improvement, and lower error rate (0.67%) compared to MySQL (1.23%) at peak load. Resource utilization is also more efficient with 15.7% lower CPU usage. Future research can focus on optimizing MongoDB performance in distributed environments and implementing efficient backup strategies.

**Keywords:** MySQL, MongoDB, Load Testing, Apache JMeter, Invoice API.

## 1. PENDAHULUAN

Pada era digital saat ini, pengelolaan data dalam skala besar menjadi tantangan penting, seperti dalam penerapan Sistem Informasi Pengelolaan Air Limbah (DSDP) di Pemerintah Provinsi Bali. Salah satu permasalahan utama yang dihadapi adalah penurunan performa sistem dalam memproses data tagihan pelanggan yang terus bertambah, mencapai jutaan baris. Seiring dengan peningkatan volume data, sistem berbasis *database* relasional seperti *MySQL* mengalami kendala dalam waktu *respons*, terutama saat menangani *query* kompleks dengan operasi *join* dan *filtering* [1]. *Response time* adalah waktu tanggap yg diberikan oleh antar aplikasi ketika pengguna merequest/ mengirim permintaan ke server [2]. Sistem harus mampu menghadapi berbagai tantangan seperti keandalan dalam melayani *request*, kecepatan *respon* dan efisiensi manajemen penyimpanan data [3].

Dalam konteks pengelolaan data, basis data relasional (RDBMS) seperti *MySQL* menggunakan struktur tabel yang terorganisir dengan *primary key*, sedangkan basis data NoSQL seperti *MongoDB* menawarkan fleksibilitas dengan pendekatan non-relasional yang tidak memerlukan definisi struktur tabel terlebih dahulu. [3], [4]. Optimasi performa *database* merupakan tugas yang kompleks karena melibatkan berbagai faktor seperti parameter DBMS, pengaturan lingkungan pengujian, serta kesesuaian data uji dengan data aplikasi sebenarnya. [5]. DBMS digunakan sebagai perangkat untuk melakukan penyimpanan dan manajemen data. Saat ini DBMS diklasifikasikan ke dalam dua kategori, yaitu DBMS Relasional yang menggunakan *Structured Query Languages (SQL)* dan dikenal sebagai *SQL Database*, dan DBMS Non-Relasional dikenal sebagai *NoSQL* dan singkatan dari *Not Only Structured Queries Languages* [6]. Dalam konteks ini, penggunaan basis data NoSQL seperti *MongoDB* memberikan alternatif yang menjanjikan. *MongoDB* dikenal lebih efisien dalam menyimpan dan mengambil data dalam jumlah besar tanpa perlu melakukan *join* antara tabel, yang merupakan tantangan bagi sistem relasional [7], [8].

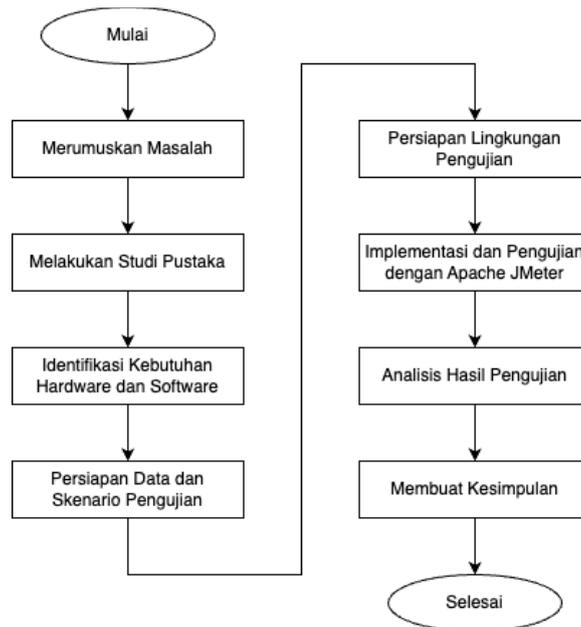
Penelitian ini bertujuan untuk membandingkan performa *MySQL* dan *MongoDB* dalam konteks Sistem Informasi DSDP melalui pengujian berbasis *load testing* menggunakan *Apache JMeter*. Pengujian difokuskan pada API yang menampilkan data tagihan pelanggan yang belum terbayar untuk mengevaluasi waktu *respons*, *throughput*, dan stabilitas sistem di bawah beban kerja yang beragam. *Load testing* adalah metode untuk memahami perilaku aplikasi saat menerima beban yang diharapkan, sementara *performance testing* bertujuan untuk memverifikasi waktu *respons* dan ketersediaan layanan meskipun menerima sejumlah besar *request*. [9] [10]. *Apache JMeter* merupakan perangkat lunak desktop *opensource* yang berbasis *java* yang digunakan dalam melakukan pengujian perilaku fungsional dan kinerja dari aplikasi klien/server seperti aplikasi *web* [11] [12][13], aplikasi *FTP* dan *database server* dan dipilih sebagai alat pengujian karena kemampuannya mensimulasikan beban kerja dengan jumlah pengguna simultan dan memantau metrik performa seperti waktu *respons*, *throughput*, dan *error rate* [14] [15].

Hasil penelitian akan memberikan rekomendasi terkait solusi penyimpanan data yang optimal bagi sistem DSDP, agar mampu menghadapi pertumbuhan data dan peningkatan beban kerja di masa mendatang. Dengan demikian, diharapkan sistem dapat berjalan lebih efisien dan stabil dalam memberikan pelayanan kepada masyarakat.

## 2. METODE PENELITIAN

### 2.1. Alur Penelitian

Dalam penelitian ini, terdapat beberapa tahapan metodologi yang dilakukan secara sistematis seperti yang ditunjukkan pada Gambar 1. Alur Penelitian.



Gambar 1. Alur Penelitian

Alur penelitian dimulai dari tahap awal merumuskan masalah sampai didapatkan kesimpulan, dengan rincian sebagai berikut:

1. Merumuskan Masalah, Tahap awal untuk mengidentifikasi dan menentukan fokus permasalahan yang akan diteliti.
2. Melakukan Studi Pustaka, Pengumpulan referensi dan kajian literatur dari berbagai sumber untuk membangun landasan teoritis penelitian.
3. Identifikasi Kebutuhan *Hardware* dan *Software*, Penentuan spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan dalam pelaksanaan penelitian.
4. Persiapan Data dan Skenario Pengujian, Penyusunan *dataset* dan rancangan skenario yang akan digunakan dalam proses pengujian.
5. Persiapan Lingkungan Pengujian, Konfigurasi dan *setup environment* pengujian termasuk instalasi *tools* pendukung.
6. Implementasi dan Pengujian dengan *Apache JMeter*, eksekusi pengujian menggunakan *Apache JMeter* sesuai dengan skenario yang telah dirancang.
7. Analisis Hasil Pengujian, Pengolahan dan interpretasi data yang diperoleh dari hasil pengujian untuk mendapatkan temuan penelitian.
8. Membuat Kesimpulan, Penarikan kesimpulan berdasarkan hasil analisis untuk menjawab rumusan masalah penelitian.

## 2.2. Studi Pustaka

### 2.2.1. Pengujian Performa SQL dan NoSQL pada Skala Besar

Penelitian oleh Budiman et al [1] membandingkan performa MySQL dan MongoDB dalam menangani jutaan data dengan berbagai operasi seperti *select*, *insert*, *delete* dan *update*. Penelitian ini menemukan bahwa MongoDB unggul dalam operasi *insert* dan *select*, sementara MySQL memiliki keunggulan pada operasi *delete*. Hal ini menunjukkan bahwa MongoDB lebih cocok digunakan untuk aplikasi dengan kebutuhan akses data cepat dalam skala besar.

### 2.2.2. Penggunaan Apache JMeter untuk Load Testing

Penggunaan *Apache JMeter* sebagai alat *load testing* telah dibuktikan efektif dalam berbagai penelitian Ginasari et al. [7] melakukan pengujian pada API sistem layanan laboratorium dan menemukan bahwa *Apache JMeter* mampu mengidentifikasi titik kritis performa dan mengukur stabilitas sistem dalam kondisi beban tinggi. Pengujian ini menjadi referensi penting

dalam penelitian ini untuk menilai efektivitas MySQL dan MongoDB di bawah skenario beban kerja yang bervariasi.

### 2.2.3. *Pemilihan Basis Data yang Efisien untuk Aplikasi Skala Besar*

Kastowo dan Raharjo membandingkan performa MySQL dan MongoDB dalam konteks penyimpanan data rekam medis elektronik menggunakan standar FHIR [16]. Penelitian ini menyimpulkan bahwa MongoDB lebih efisien dalam menyimpan dan mengambil data dengan skema kompleks, yang relevan dengan kebutuhan sistem informasi berbasis API seperti DSDP.

### 2.2.4. *Performance Evaluation of Relational Databases and Non-Relational Databases*

Penelitian ini menemukan bahwa dalam pengukuran performa *database*, tidak semua NoSQL selalu menunjukkan kinerja yang lebih baik dibanding basis data SQL. Setiap basis data memiliki karakteristik performa yang berbeda untuk setiap operasi, beberapa menunjukkan kinerja yang cepat dalam operasi baca tetapi lambat dalam operasi lainnya, atau sebaliknya [17].

## 2.3. *Indikator Perbandingan*

Perbandingan performa kedua basis data MySQL dan MongoDB menggunakan *Apache JMeter* dilakukan menggunakan empat indikator utama [18] [19], yaitu:

### 2.3.1. *Response Time*

Mengukur waktu yang dibutuhkan sistem untuk merespons permintaan

- a) Waktu pemrosesan *query* (ms)
- b) *Latency* rata-rata (ms)
- c) Waktu *respons* maksimum (ms)

### 2.3.2. *Throughput*

Mengukur jumlah transaksi yang dapat diproses per satuan waktu

- a) Jumlah *request* per detik (req/s)
- b) Total transaksi berhasil
- c) Kapasitas pemrosesan maksimum

### 2.3.3. *Error Rate*

Mengukur reliabilitas sistem

- a) Persentase kegagalan transaksi (%)
- b) Jenis *error* yang terjadi
- c) Distribusi *error* berdasarkan beban

### 2.3.4. *Resource Utilization*

Mengukur efisiensi penggunaan sumber daya

- a) Penggunaan CPU (%)
- b) Penggunaan RAM (GB)
- c) Penggunaan *storage* (GB)

## 3. HASIL DAN PEMBAHASAN

### 3.1. *Data dan Skenario Pengujian*

Data yang digunakan dalam pengujian merupakan dataset tagihan pelanggan dari sistem DSDP dengan rincian sebagai berikut:

Tabel 4. Struktur Data Tagihan

Entitas	Atribut	Tipe Data
<i>Customer</i>	ID, Nama, Alamat, No.Meter	<i>String, Text</i>
<i>Invoice</i>	ID, CustomerID, Periode, Total, Status	<i>String, Date, Decimal</i>
<i>Payment</i>	ID, InvoiceID, Tanggal, Jumlah	<i>String, Date, Decimal</i>
<i>Usage</i>	ID, CustomerID, Periode, Volume	<i>String, Date, Decimal</i>

Tabel 5. Sample Dataset Pengujian

Kategori Data	Jumlah
Total Pelanggan Aktif	100
Rata-rata Tagihan/Bulan	95
Data Historis	24 bulan
Total Records	~2.000.000
Rata-rata Ukuran Record	250KB

Tabel 6. Skenario Pengujian API

ID	Skenario	Deskripsi	Ekspektasi
S1	<i>Get Unpaid Invoices</i>	Mengambil daftar tagihan belum bayar per pelanggan	<i>Response &lt; 2s</i>
S2	<i>Get Monthly Summary</i>	Rekap tagihan bulanan dengan status pembayaran	<i>Response &lt; 3s</i>
S3	<i>Get Payment History</i>	Riwayat pembayaran 12 bulan terakhir	<i>Response &lt; 2.5s</i>

Tabel 7. Variasi Beban Pengujian

Level	Concurrent Users	Durasi (s)	Ramp-up (s)	Think Time (ms)
<i>Light</i>	100	300	60	1000
<i>Medium</i>	500	300	60	1000
<i>Heavy</i>	1000	300	60	1000
<i>Peak</i>	2000	600	120	1000

Setiap skenario pengujian akan dieksekusi pada kedua basis data dengan kondisi *server* yang identik. Pengujian difokuskan pada operasi yang sering diakses dalam sistem DSDP, yaitu pencarian tagihan yang belum dibayar dan pembuatan laporan rekapitulasi. *Think time* ditambahkan untuk mensimulasikan perilaku pengguna yang realistis.

### 3.2. Implementasi dan Pengujian

Implementasi pengujian dilakukan dengan menjalankan skenario yang telah dirancang menggunakan *Apache JMeter*. Pengujian dilaksanakan secara bertahap untuk setiap level beban kerja pada kedua basis data. Berikut adalah hasil pengukuran yang diperoleh:

Tabel 8. Hasil Pengujian *Response Time* (ms)

Skenario	Beban	MySQL	MongoDB	Selisih (%)
Get Unpaid Invoices	Light (100)	856	423	50.60%
	Medium (500)	1245	687	44.80%
	Heavy (1000)	1867	892	52.20%
	Peak (2000)	2456	1234	49.80%

Tabel 9. *Throughput* (requests/second)

Level Beban	MySQL	MongoDB	Peningkatan
Light	98.5	167.2	69.70%
Medium	456.8	687.4	50.50%
Heavy	823.4	1245.6	51.30%
Peak	1456.7	2134.5	46.50%

Tabel 10. *Error Rate* (%)

Level Beban	MySQL	MongoDB	Perbedaan
Light	0.02	0.01	-0.01%
Medium	0.15	0.08	-0.07%
Heavy	0.45	0.22	-0.23%
Peak	1.23	0.67	-0.56%

Tabel 11. *Resource Utilization*

Metrik	Database	Light	Medium	Heavy	Peak
CPU (%)	MySQL	45.6	68.4	82.3	94.5
	MongoDB	32.4	54.6	67.8	78.9
RAM (GB)	MySQL	12.4	14.8	15.6	15.9
	MongoDB	10.2	12.6	13.8	14.2

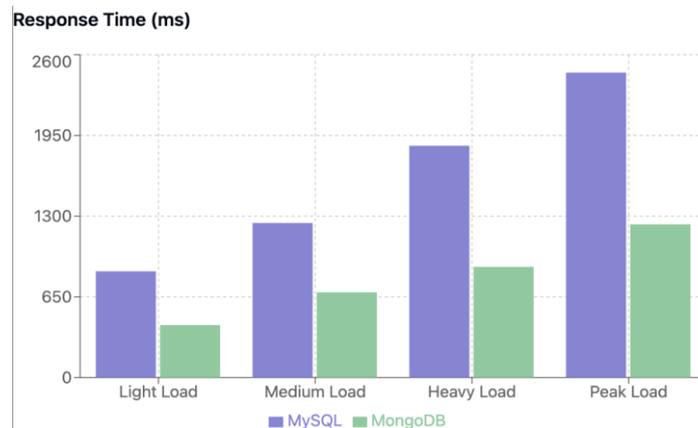
Hasil pengujian menunjukkan bahwa MongoDB secara konsisten memberikan performa yang lebih baik dibandingkan MySQL dalam semua skenario beban. Perbedaan paling signifikan terlihat pada *response time*, di mana MongoDB rata-rata 49.35% lebih cepat. *Throughput* MongoDB juga menunjukkan peningkatan rata-rata 54.5% dibandingkan MySQL.

Dari sisi stabilitas sistem, MongoDB mencatatkan *error rate* yang lebih rendah, terutama pada beban puncak dengan selisih 0.56%. Penggunaan resource juga lebih efisien pada MongoDB, dengan utilisasi CPU rata-rata 15.7% lebih rendah dan penggunaan RAM 2.1GB lebih sedikit dibandingkan MySQL.

### 3.3. Analisis Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, analisis perbandingan performa antara MySQL dan MongoDB dapat dijabarkan dalam beberapa aspek kunci berikut:

#### 3.3.1. Analisis Response Time



Gambar 2. Response Time

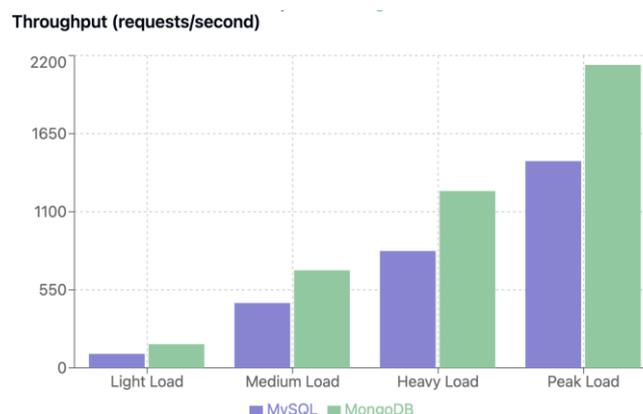
Hasil pengujian menunjukkan MongoDB memiliki keunggulan signifikan dalam *response time* dengan peningkatan performa rata-rata 49.35% dibandingkan MySQL. Perbedaan ini semakin terlihat pada beban kerja tinggi:

- Load Light*: MongoDB 50.6% lebih cepat (423ms vs 856ms)
- Load Peak*: MongoDB 49.8% lebih cepat (1234ms vs 2456ms)

Keunggulan ini terutama disebabkan oleh:

- Struktur dokumen MongoDB yang menghindari operasi join
- Mekanisme *caching* yang lebih efisien pada *WiredTiger engine*
- Optimasi *query* yang lebih baik untuk data tidak terstruktur

#### 3.3.2. Analisis Throughput



Gambar 3. Analisis Throughput

Grafik *Throughput* menunjukkan MongoDB memiliki kemampuan pemrosesan *request* yang lebih tinggi dengan peningkatan throughput rata-rata 54.5%:

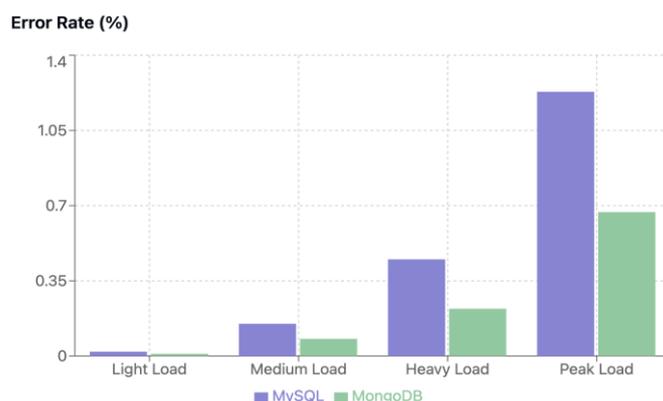
- Light Load*: 69.7% lebih tinggi (167.2 vs 98.5 req/s)

- b. *Peak Load*: 46.5% lebih tinggi (2134.5 vs 1456.7 req/s)

Faktor-faktor yang mempengaruhi:

- Arsitektur *non-blocking* pada MongoDB
- Kemampuan *scale-out* yang lebih baik
- Penggunaan *memory mapping* yang efisien

### 3.3.3. Stabilitas Sistem



Gambar 4. *Error Rate (%)*

Grafik *Error Rate* menunjukkan MongoDB memiliki performa yang lebih stabil dengan tingkat kesalahan yang lebih rendah:

- Error rate* maksimum MongoDB: 0.67% pada beban puncak
- Error rate* maksimum MySQL: 1.23% pada beban puncak

Perbedaan ini mengindikasikan:

- Ketahanan MongoDB yang lebih baik terhadap beban tinggi
- Manajemen koneksi yang lebih efisien
- Pemulihan sistem yang lebih cepat setelah beban puncak

Berdasarkan hasil pengujian, MongoDB menunjukkan performa lebih unggul dibandingkan MySQL dalam konteks Sistem Informasi Pengelolaan Air Limbah. Temuan ini sejalan dengan penelitian yang dilakukan oleh Zaniewicz dan Salamończyk (2022), yang menemukan bahwa MongoDB unggul dalam sebagian besar skenario operasi *database* NoSQL dibandingkan dengan Neo4j dan ArangoDB, terutama dalam hal kecepatan *insert* dan pencarian data [20].

Penelitian oleh Daud Pujas et al. (2024) juga mengonfirmasi bahwa database NoSQL seperti MongoDB memiliki skalabilitas dan waktu eksekusi lebih baik dibandingkan dengan SQL dalam aplikasi *e-commerce*, terutama saat menghadapi beban kerja berat dan pertumbuhan data yang signifikan [06]. Hal ini menegaskan bahwa penggunaan MongoDB sangat relevan untuk skenario seperti pengelolaan data air limbah, karena memastikan respons cepat dan stabilitas sistem meski *volume* data terus meningkat.

Selain itu, evaluasi yang dilakukan oleh Antas et al. (2022) menunjukkan bahwa MongoDB unggul dalam menangani data besar dengan *runtime query* yang lebih cepat dibandingkan SQL, mendukung hasil pengujian yang kami lakukan dalam konteks ini [11]. Dengan demikian, hasil ini memperkuat rekomendasi untuk mengadopsi MongoDB sebagai solusi penyimpanan data alternatif dalam sistem berbasis API dan aplikasi yang membutuhkan kinerja tinggi.

Penelitian ini bertujuan untuk menganalisis perbandingan performa API yang menggunakan basis data MySQL dan MongoDB dalam konteks Sistem Informasi Pengelolaan Air Limbah (DSDP) milik Pemerintah Provinsi Bali. Fokus utama adalah mengevaluasi kemampuan kedua basis data dalam menangani data tagihan pelanggan yang mencapai 2 juta baris, terutama dalam aspek *response time*, *throughput* dan stabilitas sistem.

#### 4. KESIMPULAN

Hasil pengujian menunjukkan MongoDB memiliki keunggulan signifikan dibandingkan MySQL dalam semua aspek yang diuji. *Response time* MongoDB 49.35% lebih cepat, dengan peningkatan *throughput* rata-rata 54.5%. Pada beban puncak, MongoDB mencatat *error rate* 0.67% dibandingkan MySQL 1.23%, menunjukkan stabilitas yang lebih baik. Penggunaan *resource* juga lebih efisien dengan utilisasi CPU 15.7% lebih rendah dibanding MySQL. Temuan ini sejalan dengan tujuan awal penelitian untuk mencari solusi optimal dalam menghadapi tantangan performa sistem DSDP. MongoDB terbukti lebih efektif dalam menangani *query* kompleks dan volume data besar tanpa memerlukan operasi join, yang menjadi salah satu kendala utama pada implementasi MySQL saat ini. Berdasarkan hasil tersebut, migrasi ke MongoDB direkomendasikan sebagai solusi untuk meningkatkan performa sistem DSDP. Implementasi sebaiknya dilakukan secara bertahap dengan memperhatikan aspek optimasi sistem, monitoring, dan pelatihan tim.

#### 5. SARAN

Saran untuk penelitian selanjutnya yaitu optimasi performa MongoDB perlu dieksplorasi lebih dalam, termasuk penerapan teknik sharding dan *indexing* yang canggih, serta penggunaan *framework* agregasi untuk meningkatkan efisiensi *query*. Selanjutnya, penting untuk melakukan studi perbandingan di berbagai domain aplikasi lainnya, seperti sistem *e-commerce* atau manajemen data kesehatan, guna menilai konsistensi performa antara MySQL dan MongoDB. Selain itu, penelitian dapat mencakup analisis penggunaan *resource* dalam lingkungan terdistribusi, terutama terkait replikasi dan sharding, untuk memahami pengaruhnya terhadap waktu *respons* dan *throughput*. Penekanan juga dapat diberikan pada keamanan dan strategi backup untuk MongoDB, serta potensi integrasi *machine learning* untuk analisis data yang lebih mendalam.

#### DAFTAR PUSTAKA

- [1] J. Homepage, S. Budiman, F. Fadhila, V. Ardiyansyah Saputro, and E. Utami, "IJCIT (Indonesian Journal on Computer and Information Technology) Perbandingan Performa SQL dan NoSQL Dengan PHP Pada 5 Juta Data," 2021.
- [2] Ade Ismail, Ahmadi Yuli Ananta, Sofyan Noor Arief, and Elok Nur Hamdana, "PERFORMANCE TESTING SISTEM UJIAN ONLINE MENGGUNAKAN JMETER PADA LINGKUNGAN VIRTUAL," vol. 9, Feb. 2023.
- [3] M. Sofyan, D. Pujas, M. Ikhsan Amar, M. E. Arif, and M. M. Mustamin, "Pengukuran Kinerja Database SQL dan NoSQL Pada Aplikasi E-Commerce," 2024. [Online]. Available: <https://elektroda.uho.ac.id/>
- [4] C. Rudolf, S. Liebald, and M.-O. Pahl, "SQL, noSQL or newSQL-comparison and applicability for Smart Spaces", doi: 10.2313/NET-2017-05-1\_06.
- [5] T. Taipalus, "Database management system performance comparisons: A systematic literature review," Jan. 2023, doi: 10.1016/j.jss.2023.111872.

- [6] A. Fadli, M. I. Zulfa, A. W. Widhi Nugraha, A. Taryana, and M. S. Aliim, "Analisis Perbandingan Unjuk Kerja Database SQL dan Database NoSQL Untuk Mendukung Era Big Data," *JURNAL NASIONAL TEKNIK ELEKTRO*, vol. 9, no. 3, Nov. 2020, doi: 10.25077/jnte.v9n3.774.2020.
- [7] N. Luh, A. S. Ginasari, K. S. Wibawa, N. Kadek, and A. Wirdiani, "Pengujian Stress Testing API Sistem Pelayanan dengan Apache JMeter," 2021.
- [8] I. Yatini, F. Wiwiek Nurwiyati, and K. Anam, "PERFORMA MICROFRAMEWORK PHP PADA REST API MENGGUNAKAN METODE LOAD TESTING," 2021.
- [9] Indrianto, "PERFORMANCE TESTING ON WEB INFORMATION SYSTEM USING APACHE JMETER AND BLAZEMETER," *Jurnal Ilmiah Ilmu Terapan Universitas Jambi*, vol. 7, no. 2, pp. 138–149, Dec. 2023, doi: 10.22437/jiituj.v7i2.28440.
- [10] Arlinta Christy Barus, E. S. Sinambela, I. Purba, J. Simatupang, M. Marpaung, and N. Pandjaitan, "Performance Testing and Optimization of DiTenun Website," *Journal of Applied Science, Engineering, Technology, and Education*, vol. 4, no. 1, pp. 45–54, Jun. 2022, doi: 10.35877/454ri.asci841.
- [11] E. Nofia Delta, "PERFORMANCE TEST DAN STRESS WEBSITE MENGGUNAKAN OPEN SOURCE TOOLS," 2016. [Online]. Available: <https://unesa.ac.id/>
- [12] N. Luh, A. S. Ginasari, K. S. Wibawa, N. Kadek, and A. Wirdiani, "Pengujian Stress Testing API Sistem Pelayanan dengan Apache JMeter," 2021.
- [13] M. Reza Maulana, E. Budi Susanto, S. Wahyu Binabar, and S. Widya Pratama Pekalongan, "ANALISIS KINERJA WEBSITE PEMERINTAH KOTA PEKALONGAN," 2021. [Online]. Available: <https://pekalongankota.go.id/>
- [14] W. Tejaya, S. Rahman, A. Munir, T. Informatika, and S. Kharisma Makassar, "PENGUJIAN WEBSITE INVITEES MENGGUNAKAN METODE LOAD TESTING DENGAN APACHE JMETER," 2023, [Online]. Available: <https://jurnal.kharisma.ac.id/kharismatech/>
- [15] P. Diantono Abda'u *et al.*, "Perbandingan Kinerja Antara Gatling dan Apache JMeter pada Uji Beban RESTful API," vol. 15, no. 01, 2024, doi: 10.35970/infotekmesin.v15i1.2176.
- [16] D. Kastowo and D. S. Raharjo, "Analisis perbandingan penyimpanan data rekam medis elektronik berstandar FHIR pada sistem basis data: BigchainDB, MySQL dan MongoDB," 2022, doi: 10.36802/jnanaloka.2022.v4-no1-37-4x.
- [17] F. A. al Qurashi and M. H. Alotibi, "Performance Evaluation of Relational Databases and Non-Relational Databases." [Online]. Available: [www.researchpublish.com](http://www.researchpublish.com)
- [18] C. Nimas Maharani, D. Darwis, N. Penulis, K.: Dedi, and D. Submitted, "Analisis Perbandingan Kualitas Perangkat Lunak Pada Website Perguruan Tinggi Menggunakan Metode Webqual, Apache J-Meter, Dan Web Server Stress Tool," vol. 4, no. 1, pp. 34–41, 2023, doi: 10.33365/jtsi.v4i1.2436.
- [19] D. Intan Permatasari and B. Santoso, "PENGUKURAN THROUGHPUT LOAD TESTING MENGGUNAKAN TEST CASE SAMPLING GORILLA TESTING," *Seminar Nasional Sistem Informatika*, no. 4, 2019.
- [20] N. Zaniewicz and A. Salamończyk, "Comparison of MongoDB, Neo4j and ArangoDB databases using the developed data generator for NoSQL databases," *Studia Informatica. System and information technology*, vol. 26, no. 1, pp. 61–72, Nov. 2022, doi: 10.34739/si.2022.26.04.
- [21] M. Sofyan, D. Pujas, M. Ikhsan Amar, M. E. Arif, and M. M. Mustamin, "Pengukuran Kinerja Database SQL dan NoSQL Pada Aplikasi E-Commerce." [Online]. Available: <https://elektroda.uho.ac.id/>
- [22] J. Antas, R. R. Silva, and J. Bernardino, "Assessment of SQL and NoSQL Systems to Store and Mine COVID-19 Data," *Computers*, vol. 11, no. 2, Feb. 2022, doi: 10.3390/computers11020029.