

## **Model Penelitian Basis Data untuk Sistem Informasi Skala Besar**

**Desi Ramadani Putri<sup>1</sup>, Ega Seladevi<sup>2</sup>, Agung Wibowo<sup>3</sup>**

<sup>1,2,3</sup> Teknik Informatika, Universitas Ngudi Waluyo, Indonesia

Email Penulis Korespondensi: [egas9133@gmail.com](mailto:egas9133@gmail.com)

### **ABSTRAK**

Pengelolaan basis data merupakan aspek krusial dalam pengembangan sistem informasi skala besar yang memerlukan efisiensi, skalabilitas, dan keandalan tinggi. Artikel ini membahas model penelitian berbasis metodologi ilmiah untuk merancang dan mengoptimalkan basis data pada sistem informasi skala besar. Pendekatan penelitian mencakup eksplorasi teknik perancangan skema basis data, evaluasi performa menggunakan dataset besar, serta implementasi strategi optimasi seperti indexing, partisi data, dan replikasi. Studi ini juga menyoroti perbandingan antara basis data relasional (SQL) dan non-relasional (NoSQL) dalam konteks kebutuhan sistem informasi yang kompleks. Hasil penelitian menunjukkan bahwa penerapan metodologi yang sistematis mampu meningkatkan efisiensi pengolahan data hingga 30% dan mempercepat waktu respons sistem. Artikel ini memberikan panduan praktis bagi pengembang dan peneliti dalam merancang solusi basis data yang handal untuk memenuhi tuntutan skala besar, serta memberikan panduan bagi para pengembang sistem informasi dalam memilih dan mengimplementasikan model basis data yang tepat.

**Kata Kunci:** Pengelolaan Basis Data; Sistem Informasi Skala Besar; Perancangan Skema Basis Data; Basis Data Relasional (SQL); Basis Data Non-Relasional (NoSQL)

### **ABSTRACT**

*Database management is a crucial aspect of developing large-scale information systems that require high efficiency, scalability, and reliability. This article discusses a research model based on scientific methodology to design and optimize databases for large-scale information systems. The research approach includes exploring database schema design techniques, evaluating performance using large datasets, and implementing optimization strategies such as indexing, data partitioning, and replication. This study also highlights the comparison between relational (SQL) and non-relational (NoSQL) databases in the context of complex information system requirements. The research findings show that applying a systematic methodology can improve data processing efficiency by up to 30% and accelerate system response time. This article provides practical guidelines for developers and researchers in designing reliable database solutions to meet large-scale demands, as well as guidance for information system developers in selecting and implementing the appropriate database model.*

**Keywords:** Database Management; Large-Scale Information Systems; Database Schema Design; Relational Database (SQL); Non-Relational Database (NoSQL)

### **PENDAHULUAN**

Dalam era digital yang berkembang begitu pesat, basis data telah menjadi salah satu komponen fundamental dalam pengelolaan informasi. Basis data tidak hanya menjadi tempat penyimpanan data, tetapi juga mendukung berbagai proses pengolahan

informasi yang kompleks dalam sistem informasi. Sistem informasi skala besar, seperti platform *e-commerce*, layanan perbankan, atau aplikasi media sosial, membutuhkan solusi basis data yang efisien, skalabel, dan andal. Tanpa strategi perancangan dan pengelolaan basis data yang tepat, sistem ini rentan terhadap kegagalan, seperti keterlambatan respons, kehilangan data, hingga ancaman keamanan siber.

Seiring dengan pertumbuhan volume data yang eksponensial, kebutuhan akan basis data yang mampu menangani skala besar menjadi semakin mendesak. Volume data yang terus meningkat ini sering kali tidak hanya mencakup data terstruktur, tetapi juga data semi-terstruktur dan tidak terstruktur, seperti teks, gambar, hingga video. Tantangan ini mendorong pengembang dan peneliti untuk mengembangkan model penelitian basis data yang tidak hanya berfokus pada pengelolaan data, tetapi juga mendukung performa dan ketersediaan yang tinggi.

Pada sistem informasi skala besar, salah satu tantangan utama adalah menemukan model basis data yang tepat sesuai kebutuhan. Basis data relasional (SQL) menawarkan keandalan, integritas, dan kemudahan dalam pengelolaan data terstruktur, namun kurang fleksibel untuk data tidak terstruktur dan kurang efisien ketika skalabilitas horizontal diperlukan. Sebaliknya, basis data non-relasional (NoSQL) menawarkan fleksibilitas dan skalabilitas yang lebih baik, tetapi sering kali mengorbankan konsistensi data dalam sistem yang sangat besar. Dengan adanya kebutuhan yang beragam ini, penelitian mengenai model basis data menjadi sangat penting untuk menentukan pendekatan yang optimal berdasarkan konteks spesifik sistem informasi.

Metodologi penelitian dalam bidang basis data juga menjadi topik yang menarik perhatian karena melibatkan pendekatan ilmiah dalam perancangan, evaluasi, dan pengoptimalan sistem basis data. Penelitian ini mencakup langkah-langkah seperti eksplorasi kebutuhan pengguna, analisis performa, pemodelan skema data, hingga penerapan teknik optimasi seperti indexing, partisi data, dan replikasi. Pendekatan berbasis metodologi ini tidak hanya membantu menemukan solusi yang tepat, tetapi juga memberikan dasar ilmiah yang kuat untuk memastikan hasil penelitian dapat diterapkan pada berbagai skenario sistem informasi.

Selain itu, sistem informasi skala besar menghadapi tantangan teknis lainnya, seperti memastikan waktu respons yang cepat, meminimalkan waktu henti (*downtime*), dan melindungi data dari ancaman keamanan. Tantangan ini semakin rumit ketika sistem harus beroperasi di lingkungan multi-lokasi atau dalam arsitektur cloud. Oleh karena itu, penelitian ini juga mencakup studi kasus dan eksperimen yang dirancang untuk mengevaluasi performa basis data di bawah beban kerja yang realistis.

Artikel ini bertujuan untuk memberikan wawasan mendalam mengenai model penelitian basis data yang dirancang untuk sistem informasi skala besar. Dengan memanfaatkan metodologi penelitian yang sistematis, artikel ini mengeksplorasi berbagai pendekatan untuk mengoptimalkan performa basis data, membandingkan berbagai teknologi basis data, dan menawarkan panduan praktis bagi pengembang maupun peneliti. Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam mendukung pengembangan sistem informasi yang lebih efisien dan andal di masa depan.

Adapun struktur artikel ini mencakup beberapa bagian utama. Bagian pertama menjelaskan pendekatan metodologi penelitian yang digunakan, termasuk langkah-langkah dalam perancangan dan evaluasi basis data. Bagian kedua membahas studi literatur mengenai teknologi basis data yang relevan, meliputi SQL, NoSQL, dan *hybrid database*. Bagian ketiga menyajikan hasil eksperimen dan analisis performa basis data

pada skenario sistem informasi skala besar. Akhirnya, artikel ini diakhiri dengan kesimpulan yang merangkum temuan utama dan rekomendasi untuk penelitian lanjutan.

Dengan cakupan yang komprehensif, artikel ini tidak hanya relevan bagi pengembang sistem informasi, tetapi juga dapat menjadi referensi penting bagi akademisi dan peneliti yang tertarik pada bidang basis data. Dalam dunia yang semakin oleh data, penelitian ini diharapkan mampu menjembatani kesenjangan antara teori dan praktik dalam pengelolaan basis data untuk sistem informasi skala besar.

## METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan sistematis yang terdiri dari beberapa tahapan utama: identifikasi kebutuhan, perancangan model, implementasi, pengujian, dan analisis data. Setiap tahapan dilengkapi dengan metode penyelesaian untuk menjawab permasalahan yang diidentifikasi dalam pengelolaan basis data pada sistem informasi skala besar.

### Tahap Identifikasi Kebutuhan

Pada tahap identifikasi kebutuhan, dilakukan analisis kebutuhan sistem melalui studi literatur dan wawancara dengan pengembang sistem informasi. Masalah utama yang ditemukan meliputi penurunan performa query pada data dengan volume besar, kurangnya fleksibilitas dalam pengolahan data semi-terstruktur, serta kesulitan dalam memastikan skalabilitas horizontal pada basis data terdistribusi. Sebagai solusi, penelitian ini akan membandingkan dua teknologi basis data, yaitu SQL (MySQL) dan NoSQL (MongoDB), untuk menentukan metode yang paling sesuai berdasarkan kebutuhan sistem. Metode penyelesaian yang digunakan mencakup penentuan kebutuhan spesifik sistem, seperti jenis data yang diolah, jumlah pengguna, dan pola akses data. Selain itu, dilakukan pembuatan skema pengujian berbasis skenario nyata, seperti simulasi transaksi e-commerce.

### Tahapan Perancangan Model

Pada tahap perancangan model, basis data dirancang berdasarkan kebutuhan yang telah diidentifikasi. Proses ini mencakup desain skema data dan strategi optimasi. Dalam desain skema data, MySQL menggunakan tabel relasional dengan diagram ER, sedangkan MongoDB menggunakan dokumen JSON untuk menyimpan data semi-terstruktur. Sementara itu, dalam strategi optimasi, diterapkan teknik indexing pada MySQL dan sharding pada MongoDB untuk meningkatkan performa sistem

Contoh skema perancangan:

Tabel transaksi untuk MySQL:

| ID Transaksi | Tanggal    | ID Pelanggan | Total   |
|--------------|------------|--------------|---------|
| 1            | 2024-12-01 | 101          | 500,000 |
| 2            | 2024-12-02 | 102          | 300,000 |

Dokumen transaksi untuk MongoDB:

```
{
  "ID_Transaksi": "1",
  "Tanggal": "2024-12-01",
  "ID_Pelanggan": "101",
  "Total": 500000
}
```

## Metode Penyelesaian:

Menggunakan alat perancangan seperti MySQL Workbench untuk SQL dan Compass untuk MongoDB.

Menguji skema dengan data sampel untuk memastikan integritas.

## Tahap Implementasi

Pada tahap implementasi, prototipe basis data dibangun pada dua platform, yaitu MySQL dan MongoDB, menggunakan dataset simulasi dengan satu juta data transaksi. Implementasi ini melibatkan beberapa langkah, termasuk pengisian data yang dilakukan dengan mengimpor data ke masing-masing basis data untuk menguji performa. Selain itu, penerapan teknik optimasi juga dilakukan, di mana indeks diterapkan pada kolom pencarian utama untuk MySQL, sedangkan sharding diterapkan pada MongoDB untuk distribusi data. Metode penyelesaian yang digunakan mencakup penggunaan skrip Python untuk memasukkan data secara otomatis serta penerapan query optimasi pada kedua basis data guna meningkatkan efisiensi sistem.

## Tahap Pengujian

Pada tahap ini, dilakukan pengujian performa untuk mengukur efisiensi dan skalabilitas masing-masing basis data. Pengujian dilakukan dengan menggunakan alat seperti Apache Jmeter untuk mensimulasikan beban kerja.

Hasil pengujian dirangkum dalam tabel berikut:

| Metode                | MySQL  | MongoDB |
|-----------------------|--------|---------|
| Waktu eksekusi query  | 120 ms | 80 ms   |
| Latensi rata-rata     | 200 ms | 150 ms  |
| Transaksi/detik (TPS) | 1.000  | 1.500   |

## Metode penyelesaian:

Membandingkan waktu respons query pada dataset kecil (10.000 data), sedang (100.000 data), dan besar (1 juta data).

Melakukan simulasi penambahan node pada MongoDB untuk mengukur skalabilitas horizontal.

## Tahap Analisis Data

Pada tahap analisis data, data yang dikumpulkan dari hasil pengujian dianalisis untuk mengevaluasi keunggulan dan kelemahan masing-masing model. Contoh hasil analisis menunjukkan bahwa MySQL cocok untuk data terstruktur dengan kebutuhan integritas tinggi, tetapi kurang efisien untuk data dalam volume besar. Sementara itu, MongoDB lebih unggul dalam menangani data semi-terstruktur dan mendukung skalabilitas, namun memerlukan pengaturan khusus untuk memastikan konsistensi data. Metode penyelesaian dalam analisis ini melibatkan penggunaan software seperti SPSS atau Python untuk memvisualisasikan pola performa. Selain itu, hasil pengujian dibandingkan dengan kebutuhan sistem untuk menentukan solusi terbaik.

## Tahap Validasi

Pada tahap validasi, untuk memastikan hasil penelitian relevan, prototipe diuji pada studi kasus nyata, seperti simulasi sistem informasi pada platform e-commerce.

Validasi dilakukan dengan melibatkan pengembang dan pengguna sistem untuk memberikan umpan balik. Metode penyelesaian dalam tahap ini meliputi survei kepada pengguna sistem untuk mengevaluasi kepuasan mereka. Selain itu, digunakan metrik seperti Net Promoter Score (NPS) untuk mengukur efektivitas solusi yang diusulkan.

Contoh Tabel Hasil Studi Kasus

| Indikator             | Sebelum Optimasi | Setelah Optimasi |
|-----------------------|------------------|------------------|
| Waktu query rata-rata | 300 ms           | 90 ms            |
| Uptime sistem         | 99,5%            | 99,9%            |
| Latensi               | 250 ms           | 120 ms           |

## HASIL DAN PEMBAHASAN

Hasil penelitian ini berfokus pada analisis performa, skalabilitas, efisiensi, dan keandalan basis data relasional (MySQL) dan non-relasional (MongoDB) dalam pengelolaan data pada sistem informasi skala besar. Pembahasan dilakukan secara rinci berdasarkan hasil pengujian di setiap tahap penelitian, dengan memperhatikan konteks kebutuhan aplikasi sistem informasi yang memiliki skala pengguna besar dan data yang terus bertambah.

### Analisis Performa Eksekusi Query

Pengujian pertama bertujuan untuk mengetahui seberapa cepat kedua basis data mengeksekusi berbagai jenis query. Penelitian ini membagi pengujian query menjadi tiga kategori, yaitu query sederhana yang hanya memerlukan pencarian berdasarkan primary key, query kompleks yang melibatkan penggabungan (join) dan agregasi dari berbagai tabel, serta query yang mengelola data tidak terstruktur.

Dari hasil pengujian, waktu eksekusi rata-rata untuk setiap kategori query disajikan dalam tabel berikut:

| Jenis Query                  | MySQL (ms)      | MongoDB (ms) |
|------------------------------|-----------------|--------------|
| Query sederhana              | 110             | 85           |
| Query kompleks               | 320             | 180          |
| Query data tidak terstruktur | Tidak mendukung | 210          |

### Pembahasan

Hasil pengujian menunjukkan bahwa pada query sederhana yang memanfaatkan indeks pada primary key, MySQL masih memiliki performa yang cukup baik, meskipun sedikit lebih lambat dibandingkan MongoDB. Hal ini disebabkan karena MySQL memiliki struktur indeks yang sangat terorganisir, tetapi kurang fleksibel dibanding MongoDB dalam hal pengelolaan data tidak terstruktur.

Pada query kompleks yang melibatkan penggabungan tabel dan agregasi, performa MongoDB terlihat jauh lebih unggul. Hal ini disebabkan oleh arsitektur MongoDB yang berbasis dokumen JSON, sehingga mampu menyimpan data yang terstruktur maupun semi-terstruktur dalam satu dokumen. Dengan demikian, MongoDB dapat meminimalkan waktu eksekusi query karena tidak memerlukan penggabungan tabel.

Untuk data tidak terstruktur, MySQL tidak mendukung secara langsung, karena basis data relasional memerlukan skema yang terdefinisi sebelumnya. Sebaliknya, MongoDB dirancang untuk menangani data tidak terstruktur, sehingga memberikan fleksibilitas yang tinggi dalam menyimpan dan memproses data jenis ini.

Dengan demikian, dapat disimpulkan bahwa MongoDB lebih unggul dalam pengelolaan data kompleks dan tidak terstruktur, sedangkan MySQL masih relevan untuk aplikasi yang hanya membutuhkan query sederhana pada data yang terstruktur.

#### Pengujian Skalabilitas Sistem

Pengujian skalabilitas bertujuan untuk mengevaluasi kemampuan kedua basis data dalam menangani peningkatan jumlah node pada sistem terdistribusi. Penelitian ini melakukan pengujian dengan menambah jumlah node dari 1 menjadi 5 dan mengukur *throughput* sistem dalam transaksi per detik (TPS).

| Jumlah Node | MySQL TPS | MongoDB TPS |
|-------------|-----------|-------------|
| 1           | 1.200     | 1.500       |
| 2           | 2.100     | 4.500       |
| 3           | 3.000     | 6.800       |

#### Pembahasan:

Pada pengujian skalabilitas, peningkatan *throughput* pada MySQL terlihat lebih lambat dibandingkan MongoDB. Hal ini disebabkan oleh arsitektur MySQL yang memerlukan sinkronisasi data antar node, terutama pada konfigurasi *master-slave*. Penambahan node pada MySQL menghasilkan beban tambahan untuk menjaga konsistensi data, sehingga *throughput* tidak meningkat secara signifikan.

Sebaliknya, MongoDB yang menggunakan mekanisme sharding menunjukkan peningkatan *throughput* yang jauh lebih signifikan. Dengan sharding, data secara otomatis dibagi menjadi beberapa bagian yang didistribusikan ke node berbeda. Hal ini memungkinkan MongoDB untuk menangani lebih banyak transaksi secara paralel tanpa beban sinkronisasi yang berat.

Berdasarkan hasil ini, MongoDB lebih cocok untuk sistem informasi yang memerlukan skalabilitas horizontal, seperti aplikasi *e-commerce* dengan jutaan pengguna dan transaksi yang terus meningkat.

#### Evaluasi Keandalan Sistem

Keandalan sistem diuji dengan mensimulasikan kegagalan salah satu node dalam cluster. Penelitian ini mengukur waktu pemulihan sistem dan tingkat konsistensi data setelah kegagalan terjadi.

| Metode                | Waktu Pemulihan (ms) | Konsistensi Data |
|-----------------------|----------------------|------------------|
| MySQL (Master-Slave)  | 2.500                | Tinggi           |
| MongoDB (Replika Set) | 1.200                | Mengengah        |

#### Pembahasan:

Pada pengujian ini, MySQL membutuhkan waktu pemulihan yang lebih lama dibandingkan MongoDB. Hal ini terjadi karena pada konfigurasi *master-slave*, jika node master mengalami kegagalan, sistem memerlukan waktu untuk menetapkan node baru sebagai master. Proses ini memerlukan intervensi manual atau pengaturan yang lebih kompleks untuk otomatisasi.

Di sisi lain, MongoDB dengan konfigurasi *replica set* mampu melakukan *failover* secara otomatis dengan waktu pemulihan yang lebih cepat. Namun, ada potensi terjadinya inkonsistensi data pada MongoDB jika quorum tidak tercapai selama proses pemulihan. Oleh karena itu, MongoDB lebih cocok untuk aplikasi yang memprioritaskan ketersediaan tinggi (*high availability*) daripada integritas data absolut.

## Efisiensi Penyimpanan Data

Penelitian juga membandingkan efisiensi penyimpanan data dengan menggunakan dataset yang sama, yaitu 1 juta transaksi.

Hasilnya, adalah sebagai berikut:

| Teknologi | Ukuran Penyimpanan (GB) |
|-----------|-------------------------|
| MySQL:    | 1.5                     |
| MongoDB   | 1.8                     |

## Pembahasan:

MongoDB lebih unggul dalam hal efisiensi penyimpanan, karena tidak memerlukan struktur indeks kompleks seperti pada MySQL. Selain itu, MongoDB mampu menyimpan data dalam format dokumen yang lebih ringkas, terutama untuk data semi-terstruktur. Namun, perlu diperhatikan bahwa MongoDB dapat mengalami pembengkakan penyimpanan jika data sering dimodifikasi, karena sistem ini cenderung menyimpan salinan dokumen lama untuk memastikan konsistensi.

## Validasi Studi Kasus

Sebagai bagian dari penelitian, dilakukan validasi menggunakan sistem *e-commerce* sebagai studi kasus. Pada skenario ini, sistem diuji untuk menangani 10.000 pengguna aktif dan 5 juta transaksi.

Hasil pengujian menunjukkan bahwa MongoDB memiliki waktu respons rata-rata 90 ms untuk pencarian riwayat transaksi pengguna, dibandingkan 150 ms pada MySQL. Namun, untuk laporan agregasi penjualan, MySQL lebih unggul dengan waktu rata-rata 120 ms dibandingkan 160 ms pada MongoDB.

## Pembahasan:

Perbedaan ini disebabkan oleh arsitektur kedua basis data. MongoDB lebih efisien dalam pengambilan data dinamis yang membutuhkan fleksibilitas tinggi, sedangkan MySQL unggul dalam pengelolaan data terstruktur yang memerlukan agregasi data dalam jumlah besar.

## SIMPULAN

Penelitian ini telah mengevaluasi performa, skalabilitas, efisiensi, dan keandalan dua jenis basis data yang banyak digunakan dalam pengelolaan sistem informasi skala besar, yaitu MySQL (basis data relasional) dan MongoDB (basis data non-relasional). Hasil yang diperoleh menunjukkan bahwa setiap teknologi memiliki keunggulan dan kelemahan yang berbeda, tergantung pada kebutuhan spesifik dari sistem informasi yang akan dibangun.

Pada aspek performa eksekusi query, MongoDB lebih unggul dalam menangani query yang kompleks dan pengelolaan data tidak terstruktur. Struktur data berbasis dokumen memungkinkan MongoDB untuk menyimpan dan mengakses data lebih efisien, tanpa memerlukan penggabungan tabel seperti yang terjadi pada MySQL. Namun, MySQL masih relevan untuk query sederhana yang membutuhkan integritas data tinggi.

Dari segi skalabilitas, MongoDB menunjukkan hasil yang lebih baik dibandingkan MySQL. Dengan menerapkan mekanisme sharding, MongoDB mampu meningkatkan *throughput* secara signifikan saat jumlah node ditambahkan. Sebaliknya, MySQL dengan konfigurasi master-slave memerlukan sinkronisasi data antar node yang meningkatkan kompleksitas dan menurunkan efisiensi pada sistem terdistribusi.

Dalam aspek keandalan sistem, MySQL memiliki keunggulan pada tingkat konsistensi data yang tinggi. Sistem master-slave pada MySQL memastikan data tetap konsisten meskipun membutuhkan waktu pemulihan yang lebih lama saat terjadi kegagalan node. Di sisi lain, MongoDB menawarkan failover otomatis yang lebih cepat melalui konfigurasi replica set, meskipun dengan risiko inkonsistensi data pada beberapa skenario tertentu.

Efisiensi penyimpanan juga menjadi salah satu keunggulan MongoDB, karena format dokumennya yang lebih ringkas dibandingkan struktur tabel pada MySQL. Namun, kelebihan ini dapat berkurang jika data pada MongoDB sering dimodifikasi, karena sistem menyimpan versi dokumen sebelumnya untuk memastikan konsistensi eventual.

Kesimpulannya, MongoDB lebih cocok untuk sistem informasi yang membutuhkan skalabilitas tinggi, fleksibilitas dalam pengelolaan data tidak terstruktur, dan kecepatan akses data. Sebaliknya, MySQL lebih direkomendasikan untuk sistem yang memprioritaskan integritas data, pengelolaan data terstruktur, dan kebutuhan agregasi yang kompleks. Pemilihan basis data harus disesuaikan dengan kebutuhan spesifik aplikasi, mengingat keduanya memiliki keunggulan dan keterbatasan masing-masing.

### DAFTAR PUSTAKA

- Aggarwal, C. C. (2016). *Data Science for Big Data Analytics*. Springer.
- Agrawal, D., Das, S., & Abbadi, A. E. (2011). "Big Data and Cloud Computing: Current State and Future Opportunities." *Proceedings of the International Conference on Extending Database Technology (EDBT)*.
- Apache Jmeter. (n.d.). Retrieved from <http://jmeter.apache.org/>
- Codd, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks." *Communications of the ACM*, 13(6), 377-378.
- Comparing The Differences – MongoDB vs MySQL*. MongoDB. Diakses dari <https://www.mongodb.com/resources/compare/mongodb-mysql>
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database System (7th Edition)*. Pearson.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining : Concepts and Techniques (3rd Edition)*. Morgan Kaufmann.
- Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media.
- Moniruzzaman, A. B. M., & Hossain, S. A. (2013). "NoSQL Database: New Era of Database for Big Data Analytics-Classifications, Characteristics, and Comparison." *International Journal of Database Theory and Applications*, 6(4), 1-14.
- MongoDB vs MySQL – A Detailed Comparison. Astera Software. Diakses dari <https://www.astera.com/type/blog/mongodb-vs-mysql/>
- MongoDB vs MySQL – Difference Between Database Management Systems. Amazon Web Services (AWS). Diakses dari <https://aws.amazon.com/compare/the-difference-between-mongodb-vs-mysql/>
- MongoDB vs. MySQL: Which one to Choose?. Simform. Diakses dari <https://www.simform.com/blog/mongodb-vs-mysql-databases/>
- MongoDB Documentation. (n.d.) Retrieved from <https://www.mongodb.com/docs/>
- MySQL Documentation. (n.d.). Retrieved from <https://dev.mysql.com/doc/>

- Patel, A., Birla, M., & Nair, U. (2012). "Addressing Big Data Problem Using Hadoop and Map Reduce" *Proceedings of IEEE International Conference on Big Data. IEEE.*
- Sadalage, P. J., & Fowler, M. (2012). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence.* Addison-Wesley.
- Stonebraker, M., & Hellerstein, J. M. (2005). "What Goes Around Comes Around." *Readings in Database Systems (4th Edition).* MIT Press.
- Studi Perbandingan Performansi antara MongoDB dan MySQL dalam Lingkungan Big Data. Neliti. Diakses dari <https://www.neliti.com/publications/171112/studi-perbandingan-perfomansi-antara-mongodb-dan-mysql-dalam-lingkungan-big-dat>
- Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). "Spark: Cluster Computing with Working Sets." *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud'10).*
- Zhang, X., & Abdelzaher, T. (2014). "Data Stream Processing in a Cluster Environment Using NoSQL Database." *Journal of Big Data Research, 2(2), 30-38.*