

Studi Literatur Komparasi SQL dan NoSQL dalam Pemilihan Basis Data Ideal untuk Skalabilitas Tinggi

Armana Afif Muhamad¹, Dega Adhi Kusuma², Idham Renaldi³, Agung Wibowo⁴

^{1,2,3,4}Fakultas Komputer dan Pendidikan, Teknik Informatika, Universitas Ngudi Waluyo, Semarang, Indonesia.

Email Penulis Korespondensi: armanaafif789@gmail.com

ABSTRAK

Penelitian ini membandingkan basis data SQL dan NoSQL dalam hal skalabilitas tinggi dan efisiensi pengelolaan data. SQL, dengan model relasional dan prinsip *ACID*, unggul dalam konsistensi dan integritas data, sehingga ideal untuk aplikasi yang membutuhkan transaksi tinggi seperti sistem keuangan dan bisnis. Sebaliknya, NoSQL menawarkan fleksibilitas skema, skalabilitas horizontal, dan performa tinggi dalam pengelolaan data besar yang tidak terstruktur, menjadikannya pilihan ideal untuk aplikasi modern seperti media sosial dan *e-commerce*. Dengan menggunakan metode *Systematic Literature Review (SLR)*, penelitian ini menganalisis aspek performa, model data, konsistensi transaksi, dan waktu respon berdasarkan berbagai sumber antara tahun 2018 hingga 2024. Hasil penelitian menunjukkan bahwa SQL lebih efektif untuk data terstruktur dan kebutuhan konsistensi tinggi, sementara NoSQL lebih efisien dalam menangani pertumbuhan data besar dan beban kerja dinamis. Kesimpulannya, pemilihan basis data harus disesuaikan dengan kebutuhan aplikasi, di mana SQL unggul dalam integritas data dan NoSQL menawarkan fleksibilitas serta skalabilitas untuk aplikasi modern.

Kata Kunci: Basis Data Relasional; Basis Data Non-Relasional; Skalabilitas Horizontal; Skalabilitas Vertical; Fleksibilitas Skema Data; Performa Basis Data

ABSTRACT

This research compares SQL and NoSQL databases in terms of high scalability and data management efficiency. SQL, with its relational model and ACID principles, excels in data consistency and integrity, making it suitable for high-transaction applications such as financial and business systems. Conversely, NoSQL offers schema flexibility, horizontal scalability, and high performance in managing large, unstructured data, making it an ideal choice for modern applications like social media and e-commerce. Using the Systematic Literature Review (SLR) method, this research analyzes performance aspects, data models, transaction consistency, and response time based on various sources from 2018 to 2024. The findings indicate that SQL is more effective for structured data and high consistency needs, while NoSQL is more efficient in handling large data growth and dynamic workloads. In conclusion, database selection should be tailored to application needs, with SQL excelling in data integrity and NoSQL offering flexibility and scalability for modern applications.

Keywords: Relational Database; Non-Relational Database; Horizontal Scalability; Vertical Scalability; Schema Flexibility; Database Performance

PENDAHULUAN

Kemajuan teknologi saat ini sangat mempengaruhi paradigma baru di era teknologi informasi, sehingga dalam pengembangan aplikasi diwajibkan menggunakan

basis data. Dengan teknik replikasi, data didistribusikan ke lokasi yang berbeda melalui jaringan lokal atau internet. Sistem pengelolaan basis data biasanya menggunakan satu *server* yang memiliki sistem *database* sendiri, yang memiliki kelemahan yaitu jika *server* utama mengalami gangguan, maka akan mempengaruhi seluruh sistem pengelolaan *database* yang ada (Shafa Salsabila et al., 2021).

SQL adalah bahasa pemrograman standar yang digunakan untuk mengelola basis data relasional. Dalam SQL, data diatur dalam tabel-tabel yang terstruktur menggunakan model relasional, sehingga mempermudah proses penyimpanan dan pengambilan data. SQL banyak digunakan dalam aplikasi bisnis, keuangan, dan operasional karena kemampuannya dalam menjaga konsistensi data melalui model relasionalnya. Beberapa implementasi populer dari SQL termasuk MySQL, PostgreSQL, dan Microsoft SQL Server. SQL juga unggul dalam hal kapasitas penyimpanan yang besar dan keamanan data yang lebih baik, terutama karena desain relasionalnya yang mengurangi kemungkinan kerusakan data (BAIHAQI, 2024). SQL adalah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Secara *de facto*, bahasa ini menjadi standar yang digunakan dalam manajemen basis data relasional. Saat ini, hampir semua *server* basis data mendukung bahasa ini untuk pengelolaan data (Shafa Salsabila et al., 2021).

NoSQL mengacu pada jenis basis data yang tidak menggunakan model relasional seperti SQL. Sebaliknya, NoSQL mendukung model data seperti dokumen, *key-value*, *graph*, dan *column-based*, yang memungkinkan fleksibilitas lebih besar dalam menyimpan dan memproses data tidak terstruktur atau semi-terstruktur. NoSQL berkembang pesat seiring dengan meningkatnya kebutuhan penyimpanan data besar pada aplikasi seperti media sosial, *e-commerce*, dan layanan *cloud*. Salah satu keunggulan utama NoSQL adalah skalabilitas horizontal, yang memungkinkan distribusi data ke beberapa *server*, serta fleksibilitas skema data yang memudahkan perubahan struktur data tanpa memengaruhi keseluruhan system (BAIHAQI, 2024). NoSQL adalah jenis *database* yang mampu menyimpan data terstruktur dan tidak terstruktur dalam jumlah besar untuk keperluan *Big Data*. Salah satu model NoSQL yang populer digunakan adalah *Document Oriented Database* seperti MongoDB. Tantangannya adalah MongoDB menyimpan data dalam bentuk *key-value pair*, sementara sebagian besar *database* yang digunakan saat ini masih menggunakan format tabel (kolom dan baris) dalam *relational database*. Hal ini membuat proses migrasi dari *relational database* menjadi lebih rumit (Nur et al., 2020).

Di sisi lain, muncul perkembangan dalam pengelolaan basis data yaitu NoSQL (Non SQL) dengan fitur yang dianggap lebih fleksibel dan dapat diskalakan secara horizontal. NoSQL tidak menggunakan sistem penyimpanan berbasis tabel, tetapi mengadopsi metode baru seperti *key-value*, *big table*, *document-oriented*, dan *graph*. NoSQL dianggap dapat mengatasi keterbatasan SQL sehingga dikenal sebagai basis data non-relasional. Kinerja SQL dan NoSQL tentu berbeda, sehingga itu perlu dilakukan pengujian kecepatan pencarian data untuk mengetahui perbedaannya. Hasil penelitian ini diharapkan dapat menentukan mana yang lebih cepat dalam pencarian data berskala besar antara SQL dan NoSQL, serta mengetahui akses mana yang lebih cepat dalam memproses pencarian data (Firdaus Abdi & Susanto, 2021).

Pembahasan serupa telah dibahas pada penelitian terdahulu. Pertama, dari jurnal yang berjudul “Perbandingan Kecepatan Pencarian Data SQL dan NoSQL” karya (Firdaus Abdi & Susanto, 2021). Topik penelitian pada jurnal ini adalah menganalisis perbandingan Ketepatan, Kecepatan pencarian Data, dan Efisiensi waktu Pada *database* SQL dan NoSQL. Kedua, dari jurnal yang berjudul “Perbandingan Performa SQL dan NoSQL Dengan PHP Pada 5 Juta Data” karya (Homepage et al., 2021). Topik

penelitiannya membahas mengenai pemilihan aplikasi *database* yang ideal untuk digunakan dan perbandingan kecepatan pemrosesan data antara SQL dan NoSQL. Ketiga, dari jurnal dengan judul “Analisa Perbandingan Kemampuan *Database* NoSQL dan SQL” karya (Shafa Salsabila et al., 2021) Topik penelitiannya sesuai dengan judul yaitu analisis perbandingan kemampuan *database* SQL dan NoSQL, menganalisa mengenai perbandingan kemampuan jenis basis data *Relational* dan *Non-Relational*. Keempat, dari jurnal yang berjudul “Pengukuran Kinerja *Database* SQL dan NoSQL Pada Aplikasi *Ecommerce*” karya (Sofyan et al., 2024). Topik penelitian pada jurnal ini mengukur kinerja *database* SQL dan NoSQL pada aplikasi *E-Commerce* yang berfokus pada pemecahan masalah yang berkaitan dengan peningkatan beban kerja berlebih. Kelima, dari jurnal dengan judul “Analisis Perbandingan Kinerja Waktu Respon MySQL 8.0 dan NoSQL MongoDB Menggunakan Restapi NodeJS Pada Studi Kasus Kelas Online” karya (Halimi et al., 2021). Topik penelitian yang dibahas pada jurnal ini mengenai Perbandingan kinerja waktu respon antara MySQL dan NoSQL MongoDB dapat memberikan gambaran yang lebih jelas mengenai mana yang lebih unggul dalam pengolahan data.

METODOLOGI PENELITIAN

Tahapan penelitian ini menggunakan metode *Systematic Literature Review* (SLR) untuk mengidentifikasi, mengevaluasi, dan menginterpretasi berbagai penelitian sebelumnya mengenai basis data MySQL dan NoSQL. Langkah pertama dalam penelitian ini adalah menentukan topik dan masalah utama, yakni bagaimana perbedaan antara basis data SQL dan NoSQL memengaruhi pemilihan basis data yang sesuai untuk Skalabilitas tinggi. Setelah topik ditentukan, kriteria seleksi jurnal untuk ditinjau juga ditetapkan. Kriteria inklusi mencakup jurnal, buku dan sumber internet yang diterbitkan antara tahun 2018 hingga 2024, sumber yang membahas aspek teknis dan kinerja basis data SQL dan NoSQL, serta yang diindeks di database bereputasi seperti IEEE Xplore, Scopus (Q1-Q4), Google Scholar, dan Sinta (minimal Sinta 1 atau Sinta 2). Sedangkan, kriteria eksklusif diterapkan pada sumber yang membahas teknologi basis data yang usang atau tidak relevan dengan perkembangan teknologi basis data modern.

Data dikumpulkan dengan mencari sumber-sumber relevan di *database* Scopus(Q1-Q4), IEEE Xplore, dan Google Scholar menggunakan kata kunci seperti “*SQL vs NoSQL database comparison*”, “*performance of SQL and NoSQL databases*”, dan “*scalability of NoSQL databases*”. Dari pencarian ini ada 10 jurnal, 10 buku dan 5 sumber internet yang dipilih untuk analisis lebih lanjut. Sumber-sumber ini dianalisis berdasarkan aspek utama seperti metode pengujian kinerja yang digunakan, jenis data yang dikelola (terstruktur atau tidak terstruktur), serta kelebihan dan kekurangan masing-masing basis data dalam berbagai skenario. Analisis ini menghasilkan beberapa temuan penting, seperti keunggulan SQL dalam mengelola data terstruktur yang membutuhkan konsistensi tinggi dan keunggulan NoSQL dalam menangani data tidak terstruktur dengan skalabilitas tinggi. Berikut adalah langkah-langkah penelitian yang dilakukan dalam studi ini berdasarkan metode *Systematic Literature Review* (SLR):

Table 1. Tahapan Metode Penelitian

Tahapan	Aktivitas Utama	Penjelasan
Pemilihan Topik	Identifikasi masalah utama	Fokus pada perbandingan SQL dan NoSQL untuk skalabilitas tinggi.
Kriteria Seleksi	Menentukan inklusi dan eksklusif sumber	Menggunakan jurnal, buku, dan sumber internet

Tahapan	Aktivitas Utama	Penjelasan
		yang relevan dari 2018–2024.
Pencarian Sumber	Pengumpulan referensi dari basis data bereputasi	Menggunakan kata kunci seperti “SQL vs NoSQL database comparison,” dan platform seperti Scopus.
Analisis Data	Evaluasi metode pengujian kinerja, jenis data, serta kelebihan dan kekurangan	Berfokus pada aspek performa, model data, konsistensi transaksi, dan waktu respons.
Sintesis Temuan	Menyusun perbandingan komprehensif berdasarkan analisis	Mengidentifikasi keunggulan SQL untuk data terstruktur dan NoSQL untuk fleksibilitas dan skalabilitas.

Tabel di atas merangkum tahapan utama penelitian, yang menjadi dasar dalam menganalisis kinerja basis data SQL dan NoSQL. Hasil analisis dari berbagai sumber ini kemudian disintesis untuk menyusun perbandingan yang lebih komprehensif antara SQL dan NoSQL. Temuan utama menunjukkan bahwa basis data SQL lebih cocok digunakan dalam aplikasi yang membutuhkan transaksi tinggi dengan konsistensi data yang kuat, sementara NoSQL lebih unggul dalam skenario yang memerlukan skalabilitas horizontal dan kecepatan pengolahan data besar dengan struktur yang fleksibel.

HASIL DAN PEMBAHASAN

Skalabilitas

Salah satu perbedaan utama antara SQL dan NoSQL adalah skalabilitas. SQL menggunakan pendekatan skalabilitas vertikal, yang berarti peningkatan kapasitas dilakukan dengan menambahkan sumber daya pada *server* tunggal, seperti RAM atau CPU. Walaupun efektif untuk skala kecil hingga menengah, pendekatan ini terbatas saat menangani volume data yang sangat besar tanpa arsitektur kluster tambahan. Sebaliknya, NoSQL menawarkan skalabilitas horizontal, memungkinkan penambahan node baru ke dalam kluster untuk mengelola peningkatan data. Pendekatan ini menjadikan NoSQL unggul dalam menangani data dalam skala besar, seperti pada aplikasi analitik *real-time* dan *e-commerce* dengan lalu lintas data yang tinggi. Keunggulan ini membuat NoSQL menjadi pilihan utama untuk aplikasi modern yang memerlukan distribusi data yang luas dan performa tinggi. SQL menerapkan skalabilitas vertikal, di mana peningkatan kapasitas dilakukan dengan menambah sumber daya pada satu *server*, seperti menambahkan RAM atau CPU. Meskipun pendekatan ini efektif untuk skala kecil hingga menengah, terdapat keterbatasan saat menghadapi volume data yang sangat besar tanpa tambahan arsitektur kluster. Sebaliknya, NoSQL menggunakan skalabilitas horizontal, memungkinkan penambahan node baru ke dalam kluster untuk mengelola peningkatan data. Pendekatan ini menjadikan NoSQL unggul dalam menangani data skala besar, seperti pada aplikasi analitik *real-time* dan *e-commerce* dengan lalu lintas data tinggi. Keunggulan ini membuat NoSQL menjadi pilihan utama untuk aplikasi modern yang membutuhkan distribusi data yang luas dan performa tinggi (Yusuf et al., 2021).

SQL meningkatkan kapasitas dengan menambahkan sumber daya pada satu *server*, seperti RAM atau CPU, sebuah pendekatan yang dikenal sebagai skalabilitas

vertikal. Meskipun metode ini efektif untuk skala kecil hingga menengah, ia memiliki keterbatasan dalam menangani volume data besar tanpa tambahan arsitektur klaster. Sebaliknya, NoSQL menggunakan skalabilitas horizontal, yang memungkinkan penambahan node baru ke dalam klaster untuk mengelola peningkatan data. Hal ini membuat NoSQL unggul dalam mengelola data dalam skala besar, seperti aplikasi analitik *real-time* dan *e-commerce* dengan lalu lintas data tinggi. Keunggulan ini menjadikan NoSQL pilihan utama untuk aplikasi modern yang memerlukan distribusi data luas dan performa tinggi. SQL menggunakan pendekatan skalabilitas vertikal, di mana peningkatan kapasitas dilakukan dengan menambah sumber daya pada *server* tunggal, seperti menambahkan RAM atau CPU. Meskipun efektif untuk skala kecil hingga menengah, pendekatan ini memiliki keterbatasan saat menangani volume data yang sangat besar tanpa arsitektur *cluster* tambahan (Yusuf et al., 2021).

Sebaliknya, NoSQL menawarkan skalabilitas horizontal, memungkinkan penambahan node baru ke dalam klaster untuk mengelola peningkatan data. Pendekatan ini membuat NoSQL unggul dalam menangani data skala besar, seperti aplikasi analitik *real-time* dan *e-commerce* dengan lalu lintas data tinggi. Keunggulan ini menjadikan NoSQL pilihan utama bagi aplikasi modern yang membutuhkan distribusi data luas dan performa tinggi. (Firdaus Abdi & Susanto, 2021).

Skalabilitas adalah salah satu faktor utama dalam memilih basis data. SQL mendukung skalabilitas vertikal, yang berarti performa sistem dapat ditingkatkan dengan menambah kapasitas perangkat keras pada satu *server*. Namun, metode ini memiliki batasan karena keterbatasan sumber daya fisik yang tidak bisa ditingkatkan tanpa batas. Sebaliknya, NoSQL mendukung skalabilitas horizontal, memungkinkan distribusi data dan beban kerja ke banyak *server* atau node. Hal ini membuat NoSQL lebih cocok untuk aplikasi yang memerlukan kapasitas besar dan pertumbuhan data yang cepat, seperti media sosial dan *e-commerce* (Putri, 2023; Sanjaya & Dermawan, 2024).

Model Data

Model data dalam basis data SQL dan NoSQL berbeda secara mendasar. Basis data relasional (SQL) menggunakan model data berbasis tabel yang terstruktur dan mendukung operasi relasi antar-tabel dengan menggunakan *Structured Query Language* (SQL). Sebaliknya, NoSQL menawarkan berbagai model data, seperti *key-value*, dokumen, kolom, dan graf, yang memungkinkan penyimpanan data semi-terstruktur atau tidak terstruktur dengan lebih fleksibel (Alifi et al., 2022). Model relasional menekankan integritas data dengan mematuhi prinsip ACID (*Atomicity, Consistency, Isolation, Durability*). Sebaliknya, NoSQL lebih fokus pada skalabilitas horizontal dan ketersediaan tinggi dengan mendukung berbagai skema data yang bervariasi dan denormalisasi (Fadli et al., 2020).

Dalam studi yang dilakukan oleh (Alifi et al., 2022). Pemodelan data NoSQL berbasis dokumen menunjukkan kemampuan adaptasi terhadap kebutuhan aplikasi dengan intensitas pembacaan data yang tinggi (*read-intensive*). Studi ini memperkenalkan pola desain seperti *extended reference* dan *computed* untuk meningkatkan kinerja membaca dengan mengurangi operasi gabungan (*join*). Sebaliknya, model data relasional cenderung lebih efektif untuk aplikasi dengan data yang membutuhkan konsistensi tinggi dan integritas antar-entitas melalui normalisasi data. Meskipun demikian, tantangan utama dalam NoSQL adalah kompleksitas skema data yang perlu dirancang sesuai kebutuhan aplikasi dan beban kerja tertentu (Alifi et al., 2022).

Studi oleh (Fadli et al., 2020). Performa CRUD (Create, Read, Update, Delete) antara basis data SQL dan NoSQL menunjukkan perbedaan yang signifikan. Dalam

pengujian menggunakan Redis sebagai basis data NoSQL dan MySQL sebagai basis data SQL, Redis terbukti memiliki waktu eksekusi rata-rata yang lebih cepat hingga 79,15%. Kecepatan ini disebabkan oleh arsitektur Redis yang berbasis *in-memory* dan model data *key-value* yang sederhana namun efektif untuk operasi dengan intensitas tinggi. Sebaliknya, MySQL menunjukkan waktu eksekusi yang lebih lambat karena bergantung pada penyimpanan disk dan perlunya menjaga konsistensi melalui proses transaksi (Fadli et al., 2020).

SQL mengadopsi model relasional yang terstruktur dengan rapi, di mana data disimpan dalam tabel dengan kolom dan baris yang terdefinisi dengan jelas. Model ini sangat cocok untuk data dengan pola yang tetap dan hubungan yang terdefinisi antara entitas. Di sisi lain, NoSQL menawarkan fleksibilitas yang lebih besar melalui berbagai model data, seperti dokumen, kolom, grafik, dan *key-value*. Fleksibilitas ini membuat NoSQL sangat sesuai untuk data semi-terstruktur dan tidak terstruktur, seperti data sensor IoT dan media digital (Alia et al., 2023; Putri, 2023).

Dengan menggunakan model data relasional, SQL mengorganisir data ke dalam tabel-tabel yang terdiri dari baris dan kolom. Tabel-tabel ini saling berhubungan melalui *primary key* dan *foreign key*, yang menjaga konsistensi integritas data. Model ini sangat efektif dalam mengelola data yang terstruktur dengan baik, sehingga ideal untuk aplikasi seperti sistem keuangan dan manajemen inventori yang memiliki keterkaitan data yang kompleks. Di samping itu, SQL juga memungkinkan pelaksanaan operasi analitik yang canggih menggunakan bahasa SQL standar, menjadikannya pilihan utama untuk sistem dengan kebutuhan analisis data yang tinggi (Alfan Rosid, 2020; Setiyowati & Sri Siswanti, n.d.).

NoSQL hadir dengan model data yang lebih fleksibel seperti dokumen, *key-value*, *graph*, dan *wide-column stores*, yang memudahkan aplikasi untuk menyimpan data tanpa skema yang kaku. Fleksibilitas ini menjadi keunggulan utama dalam pengembangan aplikasi modern seperti media sosial atau platform IoT, di mana struktur data kerap berubah seiring waktu. Misalnya, MongoDB menggunakan model berbasis dokumen yang memungkinkan penyimpanan data dalam format JSON tanpa batasan skema tetap. Kemampuan ini mempercepat pengembangan dan pengelolaan data di lingkungan yang dinamis (BAIHAQI, 2024; Setiyowati & Sri Siswanti, 2021).

SQL menggunakan model data relasional yang terstruktur ke dalam tabel-tabel dengan baris dan kolom. Relasi antar tabel diatur melalui *primary key* dan *foreign key*, menjaga konsistensi integritas data. Model ini unggul dalam menangani data yang sangat terstruktur, ideal untuk aplikasi seperti sistem keuangan dan manajemen inventori dengan relasi data yang kompleks. Selain itu, SQL mendukung operasi analitik yang kuat menggunakan bahasa SQL standar, menjadikannya pilihan utama untuk sistem dengan kebutuhan analisis data yang intensif (Wesclic, n.d.).

NoSQL menawarkan model data yang lebih fleksibel, seperti dokumen, *key-value*, *graph*, dan *wide-column stores*, memungkinkan penyimpanan data tanpa skema tetap. Fleksibilitas ini adalah kelebihan utama dalam pengembangan aplikasi modern seperti media sosial atau platform IoT, di mana struktur data sering berubah seiring waktu. Sebagai contoh, MongoDB menggunakan model berbasis dokumen, yang memungkinkan penyimpanan data dalam format JSON tanpa skema tetap. Fleksibilitas ini memberikan keunggulan dalam mempercepat pengembangan dan pengelolaan data di lingkungan yang dinamis (Telkom University, n.d.).

Dengan demikian, meskipun basis data relasional unggul dalam hal konsistensi dan integritas data, basis data NoSQL lebih baik dalam menangani data besar yang beragam dan dalam aplikasi *real-time*. Pemilihan antara SQL dan NoSQL sebaiknya

didasarkan pada kebutuhan spesifik aplikasi, termasuk jenis data, volume, dan pola akses yang diharapkan (Fadli et al., 2020).

Untuk memberikan gambaran yang lebih jelas mengenai perbandingan penelitian terdahulu tentang model data dalam basis data SQL dan NoSQL, tabel berikut menyajikan ringkasan beberapa studi yang telah dilakukan:

Berikut adalah perbandingan beberapa penelitian terdahulu mengenai model data dalam basis data SQL dan Penelitian

Table 2. perbandingan penelitian

Peneliti	Fokus Penelitian	Temuan Utama
Alifi et al. (2022)	Pemodelan Data NoSQL berbasis dokumen	Model NoSQL lebih fleksibel dalam menangani data semi-terstruktur
Alifi et al. (2022)	Perbandingan performa CRUD antara SQL dan NoSQL	NoSQL menunjukkan lebih cepat untuk operasi read/write dalam jumlah besar
Setiyowati & Sri Siswanti (2021)	Desain Basis Data SQL	SQL lebih unggul dalam konsistensi dan integritas data

Kebaharuan penelitian ini terletak pada analisis yang lebih mendalam tentang perbandingan SQL dan NoSQL dalam konteks skalabilitas tinggi, serta penggunaan metode Systematic Literature Review (SLR) yang menggabungkan berbagai sumber dari tahun 2018 hingga 2024. Selain itu, penelitian ini memberikan wawasan baru dalam mengembangkan model hybrid yang menggabungkan keunggulan SQL dan NoSQL untuk meningkatkan efisiensi sistem.

Konsistensi Transaksi

SQL menekankan konsistensi melalui penerapan prinsip ACID (*Atomicity, Consistency, Isolation, Durability*), yang memastikan data tetap konsisten bahkan saat terjadi kegagalan sistem. Ini membuat SQL menjadi pilihan terpercaya untuk aplikasi yang membutuhkan akurasi data tinggi, seperti sistem keuangan. Sebaliknya, NoSQL menerapkan prinsip BASE (*Basically Available, Soft state, Eventual consistency*), yang mengorbankan konsistensi kuat demi memastikan ketersediaan dan kecepatan. Pendekatan ini lebih cocok untuk aplikasi yang dapat menerima konsistensi data yang tertunda, seperti sistem analitik data besar dan platform streaming (Putri, 2023; Sanjaya & Dermawan, 2024).

SQL memastikan konsistensi transaksi dengan menerapkan prinsip-prinsip ACID (*Atomicity, Consistency, Isolation, Durability*). Dengan demikian, setiap perubahan data harus diselesaikan sepenuhnya atau tidak sama sekali. Kelebihan ini sangat krusial untuk aplikasi yang membutuhkan tingkat akurasi tinggi, seperti sistem perbankan atau pengolahan pembayaran, di mana kegagalan transaksi bisa menimbulkan masalah serius. Sistem ini memastikan bahwa data tetap konsisten meskipun terjadi gangguan eksternal atau kegagalan sistem (Alfan Rosid, 2020; Setiyowati & Sri Siswanti, 2021).

Sebaliknya, NoSQL menggunakan prinsip CAP theorem (*Consistency, Availability, Partition Tolerance*) yang memungkinkan adanya *trade-off* antara konsistensi dan ketersediaan. Keunggulan pendekatan ini adalah kemampuannya untuk memberikan ketersediaan tinggi dalam sistem terdistribusi yang skalabel. Sebagai contoh,

MongoDB mendukung konsistensi eventual, yang memungkinkan data terbaru tersedia secara bertahap di seluruh node. Pendekatan ini sangat efektif untuk aplikasi seperti layanan streaming atau media sosial, di mana prioritas diberikan pada ketersediaan data yang cepat daripada konsistensi langsung (BAIHAQI, 2024).

SQL memastikan konsistensi transaksi dengan menerapkan properti ACID (*Atomicity, Consistency, Isolation, Durability*), sehingga setiap perubahan data harus diselesaikan sepenuhnya atau tidak sama sekali. Keunggulannya ini sangat penting dalam aplikasi yang membutuhkan akurasi tinggi, seperti sistem perbankan atau pengolahan pembayaran, di mana kegagalan transaksi dapat menyebabkan masalah serius. Sistem ini memastikan bahwa data selalu dalam keadaan konsisten, meskipun ada gangguan eksternal atau kegagalan sistem (CloudAja, n.d.).

NoSQL, di sisi lain, mengadopsi model konsistensi eventual, yang berarti data akan konsisten seiring waktu, tetapi mungkin tidak langsung setelah penulisan. Pendekatan ini meningkatkan ketersediaan dan toleransi partisi, sesuai dengan teorema CAP. Keunggulan dari pendekatan ini adalah kemampuannya untuk memberikan ketersediaan tinggi dalam sistem terdistribusi yang skalabel. Hal ini sangat efektif untuk aplikasi seperti layanan siaran langsung (*Streaming*) atau media sosial, di mana prioritas diberikan pada ketersediaan data yang cepat daripada konsistensi langsung (IDS, 2023).

Kecepatan Waktu Respon

SQL memberikan performa terbaik pada data terstruktur dengan kueri yang dioptimalkan menggunakan indeks dan skema tetap. Namun, untuk data dengan volume besar atau akses tidak terstruktur, NoSQL lebih unggul. Arsitektur terdistribusi NoSQL memungkinkan pembagian beban kerja ke beberapa node, sehingga waktu respons tetap cepat meskipun data terus bertambah. Contoh aplikasi yang memanfaatkan keunggulan ini termasuk mesin pencari dan sistem rekomendasi (Alia et al., 2023; Putri, 2023).

Untuk operasi CRUD (*Create, Read, Update, Delete*), MongoDB menawarkan waktu respon yang lebih cepat dibandingkan dengan MySQL. Sebagai ilustrasi, dalam pengujian yang dilakukan pada data hingga 150.000 entri, MongoDB consistently memperlihatkan performa yang lebih unggul dalam kecepatan respon, dengan kecepatan rata-rata sekitar 25 ms lebih baik daripada MySQL (Halimi et al., 2021).

Dalam pengujian kinerja untuk aplikasi *e-commerce*, MongoDB terbukti memiliki waktu eksekusi yang lebih cepat dibandingkan dengan MySQL. Misalnya, MongoDB membutuhkan waktu lebih sedikit untuk melakukan operasi penyimpanan (*create*) dan pembacaan (*read*) data, terutama pada dataset yang lebih besar. Contohnya, untuk menyimpan data sebanyak 1000 entri, MongoDB hanya memerlukan waktu 14,48 menit, sementara MySQL membutuhkan 53,97 menit. Begitu pula dalam operasi pembacaan data sebanyak 1000 entri, MongoDB hanya memerlukan 3 detik, sedangkan MySQL membutuhkan 5 detik (Sofyan et al., 2024).

Keunggulan dan Kelemahan

Adapun perbandingan keunggulan dan kelemahan antara *database* SQL dan NoSQL Untuk memberikan pemahaman yang lebih mendalam mengenai kelebihan dan kekurangan antara basis data SQL dan NoSQL, berikut ini tabel perbandingan berdasarkan analisis dari berbagai aspek yang diuji dalam penelitian.

Table.3 Perbandingan Keunggulan dan Kelemahan SQL dan NoSQL

Aspek	SQL	No SQL
Struktur Data	Terstruktur (relasional), menggunakan tabel dengan relasi antar tabel.	Non-relasional, mendukung berbagai model data seperti dokumen, <i>graph</i> , <i>key-value</i> .
Kinerja Operasi CRUD	Lebih lambat dibandingkan NoSQL pada data besar.	Lebih cepat, terutama untuk data besar dan tidak terstruktur.
Kinerja Query Relasional	Sangat baik untuk query kompleks menggunakan JOIN atau penggabungan tabel.	Kurang optimal dalam query relasional kompleks.
Skalabilitas	Kurang fleksibel untuk skalabilitas horizontal.	Skalabilitas horizontal yang sangat baik, cocok untuk data yang tumbuh pesat.
Penyimpanan Data	Membutuhkan ruang penyimpanan lebih besar. Contoh: 610 KB untuk dataset tertentu.	Lebih efisien dalam penyimpanan. Contoh: hanya 388 KB untuk dataset yang sama.
Kecepatan Waktu Respon	Lebih lambat pada operasi <i>create</i> dan <i>read</i> untuk data besar.	Lebih cepat pada operasi <i>create</i> dan <i>read</i> , terutama untuk data besar.
Kompleksitas Implementasi	Relatif lebih sederhana untuk aplikasi dengan kebutuhan relasi tabel yang jelas.	Membutuhkan desain data yang lebih fleksibel tetapi bisa menjadi rumit tanpa skema yang jelas.
Kasus Penggunaan	Aplikasi dengan data terstruktur dan relasi yang kompleks, seperti sistem keuangan atau ERP.	Aplikasi dengan data besar, semi-terstruktur atau tidak terstruktur, seperti <i>e-commerce</i> , <i>big data</i> .
Kelemahan Utama	kurang efisien untuk menangani data yang besar atau tidak terstruktur, serta memiliki keterbatasan dalam hal skalabilitas horizontal.	tidak optimal untuk menangani query relasional yang kompleks, sehingga memerlukan desain data yang lebih matang dan terstruktur.

Keamanan

Keamanan adalah aspek penting dalam memilih basis data. SQL memiliki fitur keamanan yang mapan, seperti kontrol akses berbasis peran, enkripsi, dan logging aktivitas pengguna. Fitur-fitur ini menjadikan SQL pilihan yang ideal untuk aplikasi yang menangani data sensitif. Sebaliknya, NoSQL terus berkembang dalam hal keamanan, dengan semakin banyak sistem yang mendukung autentikasi pengguna dan enkripsi. Namun, kurangnya standarisasi dalam implementasi NoSQL dapat menjadi tantangan bagi pengembangan (Putri, 2023; Sanjaya & Dermawan, 2024).

SQL terkenal dengan keamanan yang kuat, termasuk dukungan untuk *Role-Based Access Control* (RBAC) yang memungkinkan pengaturan hak akses hingga tingkat kolom atau baris. Ini merupakan keunggulan utama dalam melindungi data sensitif, karena hanya pengguna dengan hak akses tertentu yang dapat mengelola atau melihat data. Selain itu, SQL mendukung enkripsi bawaan, memperkuat keamanannya dalam sistem yang memerlukan perlindungan data tinggi (Alfan Rosid, 2020; Setiyowati & Sri Siswanti, 2021).

Sebaliknya, NoSQL menawarkan pendekatan keamanan yang beragam dan fleksibel. Sebagian besar basis data NoSQL, seperti MongoDB, mendukung kontrol akses granular dan enkripsi data pada tingkat koneksi. Keunggulan ini membuatnya cocok untuk lingkungan terdistribusi, di mana data perlu dienkripsi selama proses komunikasi antar node. Namun, karena NoSQL bersifat lebih fleksibel, sering kali diperlukan desain keamanan khusus untuk memastikan perlindungan data yang optimal (BAIHAQI, 2024; Setiyowati & Sri Siswanti, 2021).

SQL dikenal dengan keamanan yang kuat, termasuk dukungan untuk kontrol akses berbasis peran (*Role-Based Access Control* atau *RBAC*), yang memungkinkan pengaturan hak akses hingga tingkat kolom atau baris. Ini adalah keunggulan utama dalam melindungi data sensitif, karena hanya pengguna dengan hak akses tertentu yang dapat mengelola atau melihat data tersebut. Selain itu, SQL mendukung enkripsi bawaan, yang memperkuat keamanan dalam sistem yang membutuhkan perlindungan data tinggi (cmlabs, n.d.).

NoSQL, di sisi lain, menawarkan pendekatan keamanan yang beragam dan fleksibel. Beberapa basis data NoSQL menyediakan fitur keamanan seperti autentikasi, otorisasi, dan enkripsi untuk melindungi data dari akses yang tidak sah serta mengurangi risiko pencurian data. Namun, karena sifat NoSQL yang lebih fleksibel, keamanan sering kali perlu dirancang khusus untuk memastikan perlindungan data yang optimal (cmlabs, n.d.).

SIMPULAN

Penelitian ini menyimpulkan bahwa pemilihan basis data harus sesuai dengan kebutuhan aplikasi. SQL sangat cocok untuk sistem yang mengutamakan konsistensi data, seperti aplikasi keuangan dan bisnis. Sebaliknya, NoSQL lebih unggul untuk aplikasi modern yang memiliki data besar, semi-terstruktur, dan tidak terstruktur, seperti media sosial dan *e-commerce*. Kemampuan skalabilitas horizontal dan fleksibilitas skemanya membuat NoSQL menjadi pilihan utama untuk kebutuhan pertumbuhan data yang cepat. Implementasi yang tepat dari kedua jenis basis data ini dapat meningkatkan efisiensi dan kinerja sistem sesuai kebutuhan spesifiknya.

Untuk penelitian di masa depan, disarankan untuk melakukan eksperimen langsung dengan menerapkan basis data dalam skenario dunia nyata guna mengukur performa dengan lebih akurat. Selain itu, pengembangan model hybrid yang menggabungkan keunggulan SQL dan NoSQL dapat menjadi solusi inovatif untuk meningkatkan efisiensi sistem. Penelitian lebih lanjut juga diperlukan untuk menganalisis aspek keamanan dan pengelolaan akses data antara kedua jenis basis data. Menggunakan metodologi yang lebih luas, seperti studi kasus pada industri tertentu, akan memberikan wawasan yang lebih mendalam. Terakhir, penelitian mendatang bisa berfokus pada pengembangan algoritma optimasi untuk pemilihan basis data berdasarkan kebutuhan aplikasi yang spesifik.

DAFTAR PUSTAKA

- Alfan Rosid, M. (2020). *BASIS DATA UNTUK INFORMATIKA*.
- Alia, P. A., Maspupah, A., Cahyono, W. A., Setyadinsa, R., Febriana, R. W., Mair, Z. R., Remawati, D., Setyadi, A. T., Wijaya, A., Na'am, J., Khairunusi, Y., Prasetyo, D., Kartiko, E. Y., Prayogo, J. S., & Kartiko, E. Y. (2023). *Buku Sistem Basis Data*.
- Alifi, M. R., Semiawan, T., Lieharyani, D. C. U., & Hayati, H. (2022). Pemodelan Data Relasional pada NoSQL Berorientasi Dokumen. In *Jurnal Nasional Teknik Elektro dan Teknologi Informasi* / (Vol. 11, Issue 3).
- BAIHAQI, Y. I. (2024). *SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI*.
- CloudAja. (n.d.). *Apa itu SQL dan NoSQL? Perbedaan serta Keunggulannya*. Retrieved January 7, 2025, from <https://www.cloudaja.com>
- Fadli, A., Zulfa, M. I., Widhi Nugraha, A. W., Taryana, A., & Aliim, M. S. (2020). Analisis Perbandingan Unjuk Kerja Database SQL dan Database NoSQL Untuk Mendukung Era Big Data. *JURNAL NASIONAL TEKNIK ELEKTRO*, 9(3). <https://doi.org/10.25077/jnte.v9n3.774.2020>
- Firdaus Abdi, M., & Susanto, A. (2021). PERBANDINGAN KECEPATAN Pencarian Data SQL DAN NOSQL. *Jurnal Teknologi Informasi*, 5(1).
- Halimi, A., Sudarmanto, A., & Utami, E. (2021). *ANALISIS PERBANDINGAN KINERJA WAKTU RESPON MYSQL 8.0 DAN NOSQL MONGODB MENGGUNAKAN RESTAPI NODEJS PADA STUDI KASUS KELAS ONLINE*.
- Homepage, J., Budiman, S., Fadhila, F., Ardiyansyah Saputro, V., & Utami, E. (2021). IJCIT (Indonesian Journal on Computer and Information Technology) Perbandingan Performa SQL dan NoSQL Dengan PHP Pada 5 Juta Data. In *IJCIT (Indonesian Journal on Computer and Information Technology)* (Vol. 6, Issue 1).
- IDS. (2023). *Perbedaan SQL dan NoSQL: Mana yang Tepat untuk Proyek Anda?* Retrieved January 7, 2025, from <https://www.ids.com>
- NoSQL Expert. (2022). *Mengapa NoSQL Cocok untuk Aplikasi Modern?* Retrieved January 7, 2025, from <https://www.nosqlexpert.com>
- Nur, M., Utomo, Y., Negeri, P., & Pandang, U. (n.d.). *Pengembangan Model Migrasi Database Relational ke NoSQL Memanfaatkan Metadata SQL*. <https://doi.org/10.31963/elekterika.v17i2.2212>
- Putri, R. (2023). *Buku Ajar BASIS DATA*. www.medsan.co.id
- Sanjaya, I. M. D., & Dermawan, K. T. (2024). *MANAJEMEN-BASIS-DATA-TEORI-DAN-IMPLEMENTASI(buku)*.
- Setiyowati, & Sri Siswanti. (2021). *PERANCANGAN BASIS DATA*.
- Shafa Salsabila, N., Cahya Ramdani, F., Nur Afifah, M., & Naufal Aditya, H. (2021). *Analisa Perbandingan Kemampuan Database NoSQL dan SQL*. <https://www.researchgate.net/publication/348436410>
- Sofyan, M., Pujas, D., Ikhsan Amar, M., Arif, M. E., & Mustamin, M. M. (2024). *Pengukuran Kinerja Database SQL dan NoSQL Pada Aplikasi E-Commerce*. <https://elektroda.uho.ac.id/>
- Telkom University. (n.d.). *Mengenal SQL dan NoSQL*. Retrieved January 7, 2025, from <https://www.telkomuniversity.ac.id>
- Wesclic. (n.d.). *Keunggulan SQL dalam Manajemen Data*. Retrieved January 7, 2025, from <https://www.wesclic.com>
- Yusuf, A., Muhammad, O., Reza Riansyah, M., Kusnawi, dan, Yani KM, J. A., Gambut, K., Selatan ahmadyusuf, K., Negeri, S., Jl Nusantara No, B., Selatan, L., Banjarbaru, K., Selatan omarmuhammadaa, K., Bandangtirta Agung

Sambangan, P., Bati Bati, K., Tanah Laut, K., & Selatan, K. (2021). *Studi literatur: perbandingan basis data NewSQL*.