

PENGARUH INDEXED VIEWS TERHADAP PERFORMA QUERY PADA BASIS DATA SQL SERVER DENGAN DATA SKALA BESAR

Muhammad Rama Putra, Arief Hasan, Vito Stefanus,
Aulia Bhaskoro Maghribi, Marigel Irawan, Redico Menzona

Teknik Informatika, Universitas Lancang Kuning
Jl. Yos Sudarso No.KM. 8, Umban Sari, Kec. Rumbai, Kota Pekanbaru, Riau 28266
ramaputrafriandy@gmail.com

ABSTRAK

Performa query pada basis data berskala besar sering menjadi tantangan, terutama pada sistem yang membutuhkan analisis cepat untuk mendukung pengambilan keputusan. Salah satu teknik optimasi yang dapat digunakan adalah *indexed views*, yang dirancang untuk meningkatkan efisiensi query pada SQL Server. Penelitian ini bertujuan untuk menganalisis pengaruh *indexed views* terhadap performa query berdasarkan metrik seperti waktu eksekusi, *logical reads*, dan *physical reads*. Dengan menggunakan basis data uji berisi satu juta transaksi, pengujian dilakukan pada skenario query tanpa dan dengan *indexed views*. Hasil penelitian menunjukkan bahwa *indexed views* secara signifikan meningkatkan performa, dengan pengurangan waktu CPU sebesar 100%, waktu total sebesar 96,77%, *logical reads* sebesar 98,01%, sementara *physical reads* tidak mengalami perubahan. Temuan ini mengindikasikan bahwa *indexed views* efektif untuk skenario *read-intensive*, namun kurang optimal pada sistem dengan operasi *write-intensive*. Implementasi *indexed views* direkomendasikan pada aplikasi yang memerlukan analisis data besar untuk menghasilkan respons yang lebih cepat dan efisien.

Kata kunci: *Indexed Views*, Optimasi Basis Data, Performa Query, SQL Server, *read-intensive*

1. PENDAHULUAN

Dalam era digital yang terus berkembang, teknologi informasi telah menjadi bagian integral dari operasional bisnis [1].

Hampir semua perusahaan memanfaatkan teknologi informasi untuk meningkatkan daya saing mereka [2].

Salah satu komponen utama dalam teknologi informasi adalah basis data, yang berfungsi sebagai media penyimpanan dan pengolahan data menjadi informasi yang bernilai [3].

Karena seiring dengan meningkatnya kebutuhan akan informasi yang cepat dan akurat, sistem basis data telah menjadi elemen penting dalam berbagai sektor bisnis. Namun, membangun sistem basis data yang efisien memerlukan pertimbangan yang matang, terutama dalam menghadapi tantangan seperti volume

Dalam konteks ini, *indexed views* menjadi salah satu teknik optimasi yang menawarkan keunggulan signifikan. Berbeda dengan indeks konvensional, *indexed views* memungkinkan agregasi data atau perhitungan yang sering digunakan untuk disimpan secara material, sehingga mengurangi overhead dalam pengolahan data skala besar. Teknik ini sangat cocok untuk skenario *read-intensive*, di mana query yang sering dilakukan melibatkan data dalam jumlah besar dengan pola akses yang berulang.

Penelitian ini bertujuan untuk mengkaji peran penting *indexed views* dalam meningkatkan kinerja query pada basis data berskala besar, dengan mengukur pengaruhnya terhadap metrik kinerja seperti waktu eksekusi, *logical reads*, dan *physical reads*. Dengan pemahaman ini, diharapkan solusi berbasis *indexed views* dapat diimplementasikan secara lebih

data yang terus bertambah dan kebutuhan untuk memproses query secara cepat [4].

Dalam beberapa dekade terakhir, pertumbuhan data yang eksponensial telah membuat pengelolaan data menjadi semakin kompleks, dan kemampuan sistem untuk merespons query dengan cepat serta akurat menjadi faktor krusial.

Masalah utama yang dihadapi adalah lambatnya eksekusi query pada basis data berskala besar, yang dapat menghambat pengambilan keputusan secara tepat waktu. Hal ini memunculkan kebutuhan mendesak untuk solusi yang dapat mengoptimalkan kinerja query. Salah satu solusi yang telah terbukti efektif adalah penggunaan struktur data indeks. Indeks memungkinkan query untuk mengakses data lebih cepat dengan mengurangi jumlah operasi pencarian yang diperlukan.

luas dalam aplikasi bisnis yang memerlukan analisis data besar secara efisien.

2. TINJAUAN PUSTAKA

Tinjauan ini bertujuan untuk memberikan pemahaman mendalam mengenai konteks studi serta mendukung penyusunan kerangka berpikir yang digunakan dalam penelitian ini.

2.1. Database

Database adalah kumpulan data yang terorganisasi dan disimpan secara sistematis untuk mendukung pengelolaan dan akses data secara efisien [5].

Dalam sistem modern, database digunakan untuk menyimpan informasi dalam jumlah besar dengan tetap menjaga integritas dan konsistensi data [6]. Sistem Manajemen Basis Data (Database Management

System atau DBMS) berperan penting dalam mengelola database [7].

Contoh DBMS yang banyak digunakan meliputi MySQL, PostgreSQL, dan Microsoft SQL Server [8].

Dalam dunia bisnis, database mempermudah pelacakan transaksi, manajemen inventaris, dan analisis data untuk pengambilan keputusan.

2.2. Indexed Views

Juga dikenal sebagai materialized views, adalah teknik optimasi database yang memungkinkan hasil query yang kompleks disimpan secara fisik sebagai indeks [9].

Dengan indexed views, query tidak perlu menghitung ulang data setiap kali dijalankan, sehingga mempercepat waktu eksekusi query yang sering digunakan. Indexed views sangat berguna dalam sistem yang bersifat read-intensive, seperti aplikasi pelaporan atau analitik data besar. Meskipun indexed views menawarkan peningkatan performa yang signifikan, teknik ini juga memiliki keterbatasan, seperti kebutuhan ruang penyimpanan tambahan dan biaya pemeliharaan indeks yang tinggi.

2.3. SQL Server Management Studio (SSMS)

Alat yang dikembangkan oleh Microsoft untuk mengelola, memantau, dan mengadministrasi database berbasis SQL Server [10].

SSMS menyediakan antarmuka grafis yang memungkinkan pengguna melakukan berbagai operasi, termasuk pembuatan tabel, optimasi query, dan analisis performa. Selain itu, SSMS mendukung pembuatan dan pengelolaan indexed views sebagai bagian dari strategi optimasi database. Dengan fitur-fitur canggih seperti visualisasi skema dan analitik query, SSMS menjadi alat utama bagi administrator dan pengembang database dalam mengelola sistem berbasis SQL Server.

2.4. Optimasi Query

Proses meningkatkan efisiensi eksekusi query dengan meminimalkan waktu pemrosesan dan penggunaan sumber daya. Proses ini menjadi sangat penting dalam pengelolaan database besar untuk memastikan kinerja sistem tetap optimal [11].

Teknik optimasi mencakup penggunaan indeks, penghapusan query berlebih, dan penerapan strategi seperti indexed views. Dalam skenario big data, optimasi query membantu mengatasi tantangan seperti latensi tinggi dan beban pemrosesan yang berat.

2.5. Big Data

Istilah yang mengacu pada kumpulan data berukuran besar dan kompleks yang sulit diolah menggunakan metode tradisional. Big data biasanya mencakup volume data yang sangat besar, variasi format data yang luas, dan kecepatan produksi data yang tinggi [12].

Dalam pengelolaan big data, teknik seperti indexed views menjadi penting untuk mempercepat

akses dan analisis data. Database modern sering kali dirancang untuk menangani big data dengan fitur seperti distribusi data, parallel processing, dan optimasi query.

2.6. Materialized Views

Konsep yang erat kaitannya dengan indexed views. Dalam konteks ini, materialized views mengacu pada penyimpanan fisik hasil query, yang memungkinkan akses data yang lebih cepat pada query yang sering digunakan [13].

Materialized views biasanya digunakan dalam aplikasi yang memerlukan respon cepat dan efisiensi tinggi, seperti pelaporan keuangan dan analitik bisnis. Perbedaannya dengan indexed views terletak pada implementasi spesifik dalam DBMS tertentu, di mana indexed views merupakan istilah yang digunakan dalam SQL Server.

2.7. SQL Server

Sistem manajemen basis data relasional yang dikembangkan oleh Microsoft. SQL Server dirancang untuk mendukung berbagai skenario aplikasi, mulai dari transaksi bisnis hingga analitik data skala besar [14].

Dengan fitur seperti indexed views, columnstore indexes, dan integrasi dengan alat analitik seperti Power BI, SQL Server menjadi salah satu solusi utama untuk pengelolaan data di berbagai sektor industri. SQL Server juga mendukung integrasi dengan cloud melalui layanan seperti Azure SQL Database, memungkinkan pengelolaan data yang skalabel dan fleksibel.

2.8. Manajemen Indeks

Komponen penting dalam optimasi database. Indeks membantu mempercepat pencarian data dengan menyediakan struktur data yang memungkinkan akses langsung ke baris tertentu dalam tabel. Dalam SQL Server, manajemen indeks mencakup pembuatan, pemeliharaan, dan penghapusan indeks, termasuk indexed views. Indeks yang dikelola dengan baik dapat meningkatkan performa query secara signifikan, tetapi juga memerlukan sumber daya tambahan untuk pemeliharaannya. Oleh karena itu, analisis kebutuhan dan pengujian performa sangat penting dalam manajemen indeks.

2.9. Aplikasi Analitik

Salah satu bidang utama di mana indexed views memberikan manfaat signifikan. Dalam aplikasi ini, kecepatan dan efisiensi akses data menjadi faktor kunci. Indexed views memungkinkan agregasi data dan hasil query disimpan untuk mempercepat proses analitik. Contoh aplikasi analitik termasuk pelaporan penjualan, analisis tren pasar, dan sistem rekomendasi [15].

Dengan menggunakan indexed views, organisasi dapat menghemat waktu dan sumber daya dalam pengelolaan data skala besar.

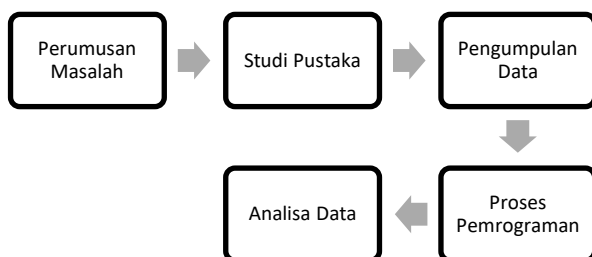
2.10. Efisiensi Penyimpanan

Pertimbangan penting dalam implementasi indexed views. Meskipun teknik ini menawarkan peningkatan performa query, indexed views memerlukan ruang penyimpanan tambahan untuk menyimpan hasil query sebagai indeks fisik. Dalam lingkungan dengan keterbatasan sumber daya, perencanaan yang matang diperlukan untuk memastikan bahwa manfaat indexed views melebihi biaya yang terkait dengan kebutuhan penyimpanan. Evaluasi rutin terhadap performa dan penggunaan ruang penyimpanan dapat membantu mengoptimalkan implementasi indexed views dalam sistem database.

3. METODE PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian adalah langkah-langkah sistematis untuk mengumpulkan dan menganalisis data guna menjawab pertanyaan penelitian secara valid, seperti pada Gambar 1 berikut



Gambar 1. Tahapan penelitian

3.2. Perumusan Masalah

Dalam penelitian ini berkaitan dengan penggunaan index view pada SQL Server Management Studio (SSMS). Masalah yang perlu dijawab adalah bagaimana penggunaan index view mempengaruhi kinerja query dalam database, serta seberapa efektif index view dalam meningkatkan kecepatan akses data dan optimasi pencarian informasi. Penelitian ini juga bertujuan untuk mengidentifikasi potensi masalah, seperti pengelolaan sumber daya yang tidak efisien atau kesalahan konfigurasi yang dapat memperlambat kinerja. Dengan merumuskan masalah ini, penelitian bertujuan untuk mengevaluasi penerapan index view di SSMS dan memberikan rekomendasi untuk perbaikan dalam desain dan implementasinya, guna mencapai efisiensi yang lebih baik dalam pengelolaan database.

3.3. Studi Pustaka

Materialized views (atau indexed views) dapat meningkatkan kecepatan eksekusi query secara signifikan. Biasanya, sistem database membatasi materialized views hanya untuk referensi ke tabel dasar untuk memudahkan pemeliharaan. "View stacking" memungkinkan materialized views merujuk

pada tabel dasar dan materialized views lain, sehingga mendukung lebih banyak jenis query. Artikel ini membahas implementasi prototipe view bertumpuk di Microsoft SQL Server, termasuk mekanisme tanda tangan untuk mengoptimalkan pencocokan view, yang juga mempercepat pencocokan view biasa. Database SQL sangat populer karena kekuatan, kesederhanaan, dan keandalannya. Namun, dengan munculnya big data, teknik optimisasi query tradisional mengalami keterbatasan. Pendekatan berbasis heuristik dan indeks dasar, yang efektif pada dataset kecil, kini kurang memadai untuk menangani latensi dan ketidakefisienan saat mengelola data berskala besar.

3.4. Pengumpulan Data

Pada tahap pengumpulan data, data yang digunakan dalam penelitian ini adalah data dummy yang digenerate secara otomatis. Data ini dirancang untuk mensimulasikan kondisi nyata dalam lingkungan database dengan volume besar, guna menguji efektivitas teknik optimisasi query. Penggunaan data dummy memungkinkan kontrol penuh terhadap variabel penelitian, serta memudahkan dalam memperoleh hasil yang konsisten dan dapat diulang. Data ini juga mencerminkan berbagai skenario yang mungkin terjadi dalam pengolahan data berskala besar.

3.5. Proses Pemrograman

Penelitian ini bertujuan membandingkan performa query pada basis data tanpa indeks dengan basis data yang menggunakan *indexed views*. Berikut adalah tahapan proses pemrograman secara lengkap:

Penelitian ini dimulai dengan pembuatan basis data tanpa indeks. Langkah pertama adalah memastikan tidak ada konflik dengan basis data yang sudah ada sebelumnya, sehingga basis data lama dihapus jika ditemukan. Setelah itu, dibuat basis data baru bernama *LargeScaleDBWithoutIndex*. Dalam basis data ini, dibuat tiga tabel utama: *Customers*, *Products*, dan *Transactions*. Tabel *Customers* menyimpan informasi pelanggan seperti ID pelanggan, nama, email, dan tanggal pendaftaran. Tabel *Products* mencatat ID produk, nama produk, dan harga. Sementara itu, tabel *Transactions* mencatat data transaksi, seperti ID transaksi, ID pelanggan, ID produk, jumlah barang yang dibeli, dan tanggal transaksi.

Setelah tabel selesai dibuat, data uji dimasukkan. Sebanyak 100.000 data pelanggan dihasilkan dengan ID unik, nama acak, email acak, dan tanggal pendaftaran dalam satu tahun terakhir. Untuk produk, dimasukkan 1.000 data dengan ID, nama produk, dan harga yang dihasilkan secara acak dalam rentang tertentu. Data transaksi dibuat dalam jumlah besar, yaitu 10 juta data, dengan pasangan ID pelanggan dan ID produk yang dipilih secara acak seperti yang ditunjukkan pada Gambar 2.

Results
TotalCustomers
2617
TotalProducts
1000
TotalTransactions
6848689

Gambar 2. Database tanpa index

Pengujian dilakukan pada basis data tanpa indeks dengan menjalankan query kompleks. Query ini bertujuan menghitung jumlah total transaksi, jumlah total belanja, dan tanggal transaksi terakhir untuk setiap pelanggan dalam periode tertentu. Statistik waktu eksekusi (*execution time*) dan jumlah pembacaan data logis (*logical reads*) dicatat sebagai metrik performa. Hasilnya menunjukkan performa dasar dari sistem tanpa adanya optimasi indeks seperti yang ditunjukkan pada Gambar 3.

Results
SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.
(1 row affected)
Table 'Customers'. Scan count 0, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Transactions'. Scan count 1, logical reads 31566, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Products'. Scan count 1, logical reads 8, physical reads 0, page server reads 0, read-ahead reads 0.
SQL Server Execution Times:
CPU time = 14907 ms, elapsed time = 17921 ms.
Completion time: 2025-01-26T16:08:05.2891229+07:00

Gambar 3. Pengujian tanpa index

Langkah berikutnya adalah membuat basis data dengan indeks dan *indexed views*. Basis data lama dihapus untuk mencegah konflik, dan basis data baru bernama *LargeScaleDBWithIndex* dibuat dengan struktur tabel yang sama seperti sebelumnya.

Langkah berikutnya adalah membuat basis data dengan indeks dan *indexed views*. Basis data lama dihapus untuk mencegah konflik, dan basis data baru bernama *LargeScaleDBWithIndex* dibuat dengan struktur tabel yang sama seperti sebelumnya. Setelah tabel selesai dibuat, *indexed view* dirancang untuk mengoptimalkan query. *Indexed view* ini menyimpan hasil perhitungan agregasi, seperti jumlah transaksi dan total belanja per pelanggan, untuk mempercepat pengambilan data. Indeks terkluster unik diterapkan pada *indexed view* agar performa query semakin optimal seperti yang ditunjukkan pada Gambar 4.

Pengujian kemudian dilakukan pada basis data ini menggunakan query yang sama seperti sebelumnya. Namun, kali ini query memanfaatkan

indexed view untuk mengambil hasil perhitungan yang telah disimpan. Statistik waktu eksekusi dan jumlah pembacaan data logis dicatat kembali untuk dianalisis. Hasil pengujian menunjukkan peningkatan performa yang signifikan. Waktu eksekusi berkurang drastis, sementara jumlah pembacaan data logis juga jauh lebih sedikit dibandingkan basis data tanpa indeks seperti yang ditunjukkan pada Gambar 5.

Results
SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.
(1 row affected)
Table 'Products'. Scan count 0, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Transactions'. Scan count 1, logical reads 31566, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Products'. Scan count 1, logical reads 8, physical reads 0, page server reads 0, read-ahead reads 0.
SQL Server Execution Times:
CPU time = 3244 ms, elapsed time = 3761 ms.
Completion time: 2025-01-26T16:10:14.7509852+07:00

Gambar 4. Database dengan index

Results
SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.
(1 row affected)
Table 'Products'. Scan count 0, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Transactions'. Scan count 1, logical reads 31566, physical reads 0, page server reads 0, read-ahead reads 0.
Table 'Products'. Scan count 1, logical reads 8, physical reads 0, page server reads 0, read-ahead reads 0.
SQL Server Execution Times:
CPU time = 3244 ms, elapsed time = 3761 ms.
Completion time: 2025-01-26T16:10:14.7509852+07:00

Gambar 5. Pengujian dengan index

3.6. Analisis Data

Data yang telah dikumpulkan akan dianalisis menggunakan metode statistik atau teknik analisis yang sesuai untuk mengevaluasi kinerja teknik optimisasi query. Pengujian dilakukan dengan membandingkan hasil waktu eksekusi query dan penggunaan sumber daya sebelum dan setelah penerapan teknik optimisasi. Data dummy yang telah digenerate otomatis akan diolah untuk mengevaluasi dampaknya dalam kondisi skala besar. Hasil analisis ini akan digunakan untuk mengidentifikasi kelebihan, kekurangan, serta potensi perbaikan dalam penerapan teknik optimisasi query.

4. HASIL DAN PEMBAHASAN

Penelitian ini mengevaluasi pengaruh penggunaan *indexed views* terhadap performa query pada basis data SQL Server dengan skala data besar. Hasil pengujian performa query dianalisis berdasarkan beberapa metrik utama, yaitu waktu eksekusi (CPU dan total), jumlah logical reads, dan jumlah physical reads.

4.1. Skenario Pengujian

Skenario pengujian dirancang untuk membandingkan performa query pada basis data tanpa indexed views dan dengan indexed views. Data yang digunakan memiliki skala besar dengan skenario query yang melibatkan operasi agregasi dan filter kompleks. Indexed views dirancang untuk mempercepat eksekusi query dengan menyimpan hasil kalkulasi tertentu sehingga mengurangi beban pengolahan ulang.

Tahapan pengujian meliputi:

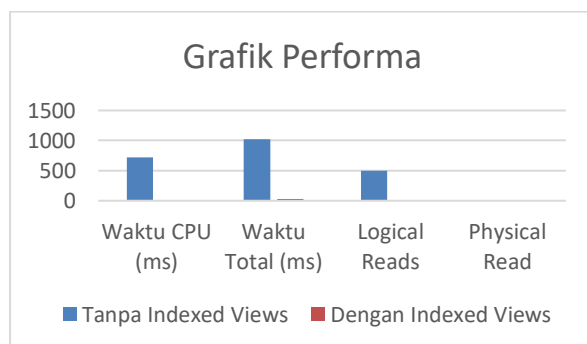
- Membuat basis data dengan dan tanpa indexed views.
- Melakukan eksekusi query dengan skenario yang sama pada kedua jenis basis data.
- Mengukur waktu CPU, waktu total, logical reads, dan physical reads.
- Membandingkan hasil pengukuran untuk menganalisis dampak penggunaan indexed views.

4.2. Hasil Pengujian Performa Query

Hasil pengujian ditampilkan dalam Tabel 1 dan Gambar 6, yang membandingkan performa antara basis data tanpa indexed views dan dengan indexed views.

Tabel 1. Tabel Performa Query

Basis Data	Waktu CPU (ms)	Waktu Total (ms)	Logical Reads	Physical reads
Tanpa Indexed Views	719	1023	503	0
Dengan Indexed Views	0	33	10	0



Gambar 6. Grafik Performa

4.3. Analisis Hasil Pengujian

Hasil analisis ini digunakan untuk mengidentifikasi kelebihan, kekurangan, serta potensi perbaikan dalam penerapan indexed views untuk optimisasi query pada basis data besar.

4.4. Waktu Eksekusi Query

Pada basis data tanpa indexed views, waktu CPU adalah 719 ms dan waktu total eksekusi 1023 ms. Sementara itu, pada basis data dengan indexed views, waktu CPU turun menjadi 0 ms (pengurangan 100%) dan waktu total menjadi 33 ms (pengurangan 96,77%).

Penurunan waktu ini menunjukkan bahwa indexed views mengurangi beban pemrosesan dengan menyimpan hasil kalkulasi sebelumnya, sehingga query tidak perlu menghitung ulang data. Hal ini sangat bermanfaat pada skenario query yang sering dilakukan dengan volume data besar.

4.5. Logical Reads

Logical reads, yang mengukur jumlah halaman memori yang dibaca dari buffer pool, turun dari 503 menjadi 10 (pengurangan 98,01%). Hal ini menunjukkan bahwa indexed views secara signifikan mengurangi jumlah operasi pembacaan yang diperlukan untuk menyelesaikan query, sehingga meningkatkan efisiensi memori.

4.6. Physical Reads

Physical reads, yang mengukur jumlah halaman yang dibaca langsung dari disk, tidak menunjukkan perbedaan antara basis data dengan dan tanpa indexed views. Hal ini disebabkan oleh data yang sudah di-cache dalam memori, sehingga tidak memerlukan akses disk secara langsung.

4.7. Pembahasan

Penggunaan indexed views terbukti memberikan peningkatan performa yang signifikan untuk query dengan agregasi besar dan operasi kompleks. Keunggulan utama indexed views meliputi:

- Pengurangan Beban CPU: Indexed views menyimpan hasil agregasi sehingga query tidak perlu menghitung ulang data.
- Peningkatan Efisiensi I/O Disk: Dengan mengurangi jumlah logical reads, indexed views mengurangi beban memori dan mempercepat eksekusi query.
- Peningkatan Respons Real-Time: Indexed views sangat cocok untuk aplikasi yang memerlukan respons cepat, seperti laporan penjualan real-time.

5. KESIMPULAN DAN SARAN

Penggunaan indexed views pada basis data SQL Server berskala besar terbukti meningkatkan performa query secara signifikan, dengan pengurangan waktu eksekusi CPU sebesar 100%, waktu total eksekusi sebesar 96,77%, dan logical reads sebesar 98,01%, sementara physical reads tidak mengalami perubahan. Hasil ini menunjukkan bahwa indexed views sangat efisien dalam pengelolaan query yang melibatkan agregasi data besar. Namun, teknik ini memiliki keterbatasan, seperti kebutuhan ruang penyimpanan tambahan dan potensi memperlambat operasi write, sehingga implementasinya perlu disesuaikan dengan jenis beban kerja sistem.

Berdasarkan hasil penelitian, disarankan untuk mengimplementasikan indexed views pada sistem read-intensive, seperti aplikasi analitik atau pelaporan, di mana keuntungan performa lebih signifikan dibandingkan biaya tambahan yang diperlukan. Untuk

sistem dengan beban kerja yang intensif pada operasi *write*, diperlukan evaluasi lebih lanjut untuk memastikan manfaat yang diperoleh melebihi dampak negatif pada performa. Selain itu, perencanaan sumber daya, khususnya terkait penyimpanan, perlu dirancang secara optimal agar efisiensi sistem tetap terjaga.

Penelitian lanjutan direkomendasikan untuk mengeksplorasi kombinasi *indexed views* dengan metode optimasi lainnya, seperti *partitioning*, *query tuning*, atau penggunaan *in-memory processing*, guna memperluas penerapan teknik ini pada berbagai jenis kebutuhan dan skenario sistem basis data. Studi mendalam mengenai dampak *indexed views* dalam sistem dengan pola akses *hybrid* antara operasi baca dan tulis juga penting dilakukan untuk memberikan panduan implementasi yang lebih komprehensif.

DAFTAR PUSTAKA

- [1] A. Alvin Adriansyah and P. Studi Manajemen, "Kajian Tentang Peran Penting Basis Data Bagi Perpustakaan," *Jurnal Ilmiah Nusantara (JINU)*, vol. 1, no. 4, pp. 3047–9673, 2024, doi: 10.61722/jinu.v1i4.1819.
- [2] D. Andharu *et al.*, "Optimasi Database Menggunakan SQL Kueri Klausa IN dan EXIST pada Database Oracle 12c. Studi Kasus pada Aplikasi di Badan Nasional Pencarian dan Pertolongan (BASARNAS)," *Jurnal Pendidikan Tambusai*, vol. 6, 2022, doi: <https://doi.org/10.31004/jptam.v6i1.3269>.
- [3] S. M. Arif and H. Purwoko, "Perancangan Basis Data Helpdesk System PT XYZ Menggunakan Microsoft SQL Server 2019," *Journal Riset dan Inovasi Teknologi*, 2023, doi: <https://doi.org/10.30998/semnasristek.v7i1.6256>.
- [4] T. S. P. Tere, S. Nirwan, and W. Resdiana, "Implementasi Algoritma Linear Search Untuk Optimasi Pencarian Pendaftar Di Disdukcapil Kabupaten Subang," *Jurnal Kecerdasan Buatan dan Teknologi Informasi*, vol. 3, no. 3, pp. 115–124, Aug. 2024, doi: 10.69916/jkbt.v3i3.158.
- [5] M. Riyan Dirgantara, S. Syahputri, and A. Hasibuan, "Pengenalan Database Management System (DBMS)," *Jurnal Ilmiah Multidisiplin*, vol. 300, no. 6, 2023, doi: 10.5281/zenodo.8123019.
- [6] B. Rawat and S. Purnama, "MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung," *International Journal of Cyber and IT Service Management (IJCITSM)*, vol. 1, no. 2, pp. 173–179, 2021, doi: 10.34306/ijcitsm.v1i1.47.
- [7] A. Deska, I. Akbar, and P. Studi Magister Ilmu Komputer, "Tuning Database Pada Sistem Penerimaan Mahasiswa Baru Menggunakan Optimasi Query dan Indexing Tuning Database on New Student Admission System Using Query and Indexing Optimization," *Jurnal Teknologi Informasi*, vol. 22, no. 1, pp. 68–77, Feb. 2023, doi: <https://doi.org/10.33633/tc.v22i1.7047>.
- [8] U. Kalsum Siregar, T. Arbaim Sitakar, S. Haramain, Z. Nur Salamah Lubis, U. Nadhirah, and F. Sains dan Teknologi, "Pengembangan Database Management System Menggunakan MySQL," *Jurnal Sains, Teknologi dan Komputer (SAINTEK)*, vol. 1, no. 1, pp. 8–12, 2024, doi: 10.56495/saintek.v1i1.450.
- [9] S. Maesaroh, H. Gunawan, A. Lestari, M. S. A. Tsaurie, and M. Fauji, "Query Optimization In MySQL Database Using Index," *International Journal of Cyber and IT Service Management*, vol. 2, no. 2, pp. 104–110, Mar. 2022, doi: 10.34306/ijcitsm.v2i2.84.
- [10] S. T. Nurhayati, M. Irwan, and P. Nasution, "Database Management System Pada Perusahaan," *Jurnal Akuntansi Keuangan dan Bisnis*, vol. 1, no. 2, pp. 62–64, 2023, [Online]. Available: <https://jurnal.itc.web.id/index.php/jakbs/index>
- [11] Y. Virmansyah, R. Yulyanto Suladi, A. Bambang Lesmana, F. Teknologi Informasi, and U. Budi Luhur, "Optimasi Database dengan Metode Index dan Partisi Tabel Database Postgresql pada Aplikasi E-Commerce. Studi pada Aplikasi Tokopintar," *Jurnal Pendidikan Tambusai*, vol. 6, 2022, doi: <https://doi.org/10.31004/jptam.v6i1.3257>.
- [12] R. E. Santoso, A. Giri Prawiyogi, U. Rahardja, F. P. Oganda, and N. Khofifah, "Penggunaan dan Manfaat Big Data dalam Konten Digital," *Jurnal Bisnis Digital*, vol. 3, no. 2, 2022, doi: <https://doi.org/10.34306/abdi.v3i2.836>.
- [13] E. Witono, "Perbandingan Response Time Penggunaan Index, Views, dan Materialized Views Database Mysql," 2022, doi: <http://dx.doi.org/10.30645/j-sakti.v6i1.463>.
- [14] V. Banja, M. Ilić, L. Kopanja, D. Zlatković, M. Trajković, and D. Čurguz, "Microsoft Sql Server And Oracle: Comparative Performance Analysis," *The 7th International conference Knowledge management and informatics*, 2021, [Online]. Available: https://www.researchgate.net/publication/352348811_MICROSOFT_SQL_SERVER_AND_ORACLE_COMPARATIVE_PERFORMANCE_ANALYSIS
- [15] F. Rahutomo, M. Eko Sulisty, B. Harjito, and J. Tetuko Sri Sumantyo, "Pendekatan Agile Scrum pada Pengembangan Aplikasi Analitik Akademik dan Kemahasiswaan," *JURNAL INOVTEK POLBENG*, vol. 7, no. 2, p. 2022, 2022, doi: <https://doi.org/10.35314/isi.v7i2.2880>.