



PENERAPAN NORMALISASI DAN DENORMALISASI UNTUK OPTIMASI PERFORMA DATABASE

Ridha Adjie Eryadi^{1*}, Alfian Maulana², Sahmi Padilaturrohman³, Zidan Zain Shibghatullah⁴

^{1,2,3,4} Program Studi Informatika, Universitas Teknologi Digital, Bandung, Indonesia

E-mail: ridhaadjie@digitechuniversity.ac.id, alfan20124088@digitechuniversity.ac.id

ABSTRACT

In relational database systems, normalization plays a major role in reducing duplicate data while ensuring information remains consistent. Yet, databases designed with full normalization often require multiple join processes when running queries, which can slow down performance. On the other hand, denormalization intentionally keeps certain repeated data to speed up access, especially in systems that rely heavily on reading operations. This research examines how both database design approaches-normalized in 3NF and denormalized structures-perform under similar conditions. Two database schemas that share the same logical design but differ in physical implementation were tested using synthetic datasets on a current Database Management System (DBMS). The evaluation focused on execution time, logical data reads, and storage requirements. From this study, practical insights are expected to help determine which database design strategy is more suitable depending on the characteristics of the workload and operational usage.

Keywords: Normalization, Denormalization, System Performance, Query Efficiency, Database Design.

ABSTRAK

Dalam sistem basis data relational, normalisasi memiliki peranan penting dalam mengurangi duplikasi data sekaligus menjaga konsistensi informasi. Namun, penerapan normalisasi penuh sering kali menuntut proses join yang lebih banyak ketika kueri dijalankan sehingga dapat memperlambat kinerja sistem. Sebaliknya, denormalisasi mempertahankan sebagian redundansi data secara sengaja untuk mempercepat akses informasi, terutama pada lingkungan dengan aktivitas pembacaan yang tinggi. Penelitian ini membandingkan kinerja antara desain basis data ternormalisasi hingga 3NF dan desain yang tidak ternormalisasi dalam kondisi yang setara. Dua skema basis data yang memiliki rancangan logis serupa namun implementasi fisiknya berbeda diuji menggunakan dataset sintesis pada Sistem Manajemen Basis Data (DBMS) modern. Pengujian difokuskan pada waktu eksekusi kueri, jumlah pembacaan logis, serta kebutuhan penyimpanan. Hasil akhir diharapkan mampu memberikan panduan dalam memilih strategi perancangan basis data yang paling tepat sesuai beban kerja dan kebutuhan operasional sistem.

Kata Kunci: Normalisasi, Denormalisasi, Efisiensi Kinerja, Eksekusi Kueri, Desain Basis Data.

Article History

Received: Januari 2026

Reviewed: Januari 2026

Published: Januari 2026

Plagiarism Checker No
235

Prefix DOI :

[10.8734/Kohesi.v1i2.365](https://doi.org/10.8734/Kohesi.v1i2.365)

Copyright : Author

Publish by : Kohesi



This work is licensed
under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)



PENDAHULUAN

Latar Belakang

Normalisasi merupakan metodologi penting dalam arsitektur database relasional, yang bertujuan meminimalkan redundansi sambil menjaga integritas informasi untuk memastikan konsistensi. Penerapan peraturan bentuk normal, termasuk First Normal Form (1NF), Second Normal Form (2NF), dan Third Normal Form (3NF), berkontribusi pada struktur database yang lebih sistematis dan efisien dalam hal penyimpanan data [1].

Namun demikian, dalam aplikasi praktis, proses normalisasi dapat mengakibatkan disagregasi data di beberapa tabel. Dalam keadaan di mana sistem memerlukan pengambilan informasi cepat yang melibatkan banyak operasi gabungan, durasi eksekusi kueri dapat meningkat pesat, sehingga mempengaruhi kinerja secara keseluruhan.

Sebaliknya, denormalisasi menyajikan strategi alternatif yang ditandai dengan pengenalan redundansi yang disengaja. Tujuan utamanya adalah untuk mengurangi ketergantungan gabungan, sehingga mempercepat proses akses data, terutama dalam skenario yang intensif membaca, seperti analisis laporan penjualan atau manajemen inventaris [2].

Denormalisasi sering digunakan untuk memastikan pelestarian data historis, dicontohkan dengan pelacakan perubahan harga dalam transaksi penjualan, sehingga memfasilitasi pelacakan yang tepat dari data masa lalu [3].

Fenomena ini menimbulkan kompromi antara integritas data dan kecepatan akses. Sistem yang dicirikan oleh volume transaksi yang tinggi cenderung memprioritaskan normalisasi untuk mengurangi inkonsistensi, sedangkan sistem analitik cenderung mengadopsi strategi denormalisasi untuk meningkatkan waktu respons saat memproses permintaan data yang ekstensif. Intinya, kedua metodologi ini memang dapat berfungsi secara sinergis, bergantung pada konteks spesifik penerapannya.

Mengingat pertimbangan ini, ada kebutuhan untuk penelitian yang berusaha mengevaluasi sejauh mana variasi dalam struktur desain database mempengaruhi kinerja sistem dalam kerangka operasional pragmatis.

Rumusan Masalah

Mengingat latar belakang yang disebutkan di atas, pertanyaan penelitian berikut telah diartikulasikan:

1. Bagaimana kinerja akses data berbeda antara database yang dinormalisasi (3NF) dan database denormalisasi?
2. Kategori kueri apa yang menghasilkan peningkatan kinerja yang substansif saat menggunakan metodologi denormalisasi?
3. Bagaimana pengaruh normalisasi dan denormalisasi terhadap waktu eksekusi operasi baca (query SELECT) pada basis data?

Tinjauan Literatur

Normalisasi telah terbukti secara efektif mengurangi redundansi data dan mencegah munculnya anomali penyisipan, pembaruan, dan penghapusan [1]. Beberapa penelitian menunjukkan bahwa Third Normal Form (3NF) dapat meningkatkan efisiensi penyimpanan tanpa memperluas ruang data secara signifikan, terutama dalam sistem skala besar.

Sebaliknya, teknik denormalisasi memperkenalkan redundansi untuk mengurangi beban gabungan ekstensif, sehingga secara signifikan meningkatkan eksekusi kueri agregat, terutama dalam konteks aplikasi pelaporan waktu nyata [3] [4] [5]. Metodologi ini dianjurkan untuk sistem yang memprioritaskan kinerja baca daripada konsistensi struktural data.

Sejumlah penelitian juga menggarisbawahi perlunya pemilihan desain database kontemporer untuk mempertimbangkan profil beban kerja, jenis aplikasi, dan frekuensi modifikasi informasi, sehingga memerlukan evaluasi empiris dalam kondisi realistis untuk memastikan strategi desain yang optimal [2].



Pendekatan yang Diusulkan

Studi ini menggunakan metodologi eksperimental kuantitatif untuk membedakan dua model database yang memiliki struktur logis identik namun berbeda dalam implementasi fisik:

Model	Pendekatan	Karakteristik Utama
Model A	Normalisasi (3NF)	Minim redundansi, banyak join
Model B	Denormalisasi	Minim join, akses cepat

Kedua model menjadi sasaran pengujian menggunakan kumpulan data sintetis yang ekstensif dalam Sistem Manajemen Database (DBMS) modern. Parameter untuk evaluasi termasuk waktu eksekusi kueri, efisiensi penyimpanan, dan frekuensi operasi pembacaan data.

Nilai Kebaruan (Inovasi)

Investigasi ini tidak hanya mencakup kerangka kerja konseptual tetapi juga melakukan penilaian empiris kinerja dalam lingkungan operasional otentik.

Ini menggunakan penggabungan tiga kategori kueri utama yang lazim dalam sistem database:

- Query seleksi
- Query join
- Query agregasi

Selain itu, ia memberikan pedoman pragmatis bagi pengembang untuk memastikan skema database yang paling sesuai berdasarkan karakteristik beban kerja.

Penelitian ini menjadi referensi akademik dalam konteks Manajemen Basis Data, khususnya mengenai strategi pemodelan struktur database.

METODE PENELITIAN

Studi ini menggunakan desain penelitian eksperimental kuantitatif untuk meneliti varians kinerja antara dua paradigma desain database yang berbeda normalisasi (3NF) dan denormalisasi di bawah skenario beban kerja yang secara akurat mencerminkan sistem dunia nyata yang ditandai dengan dominasi operasi baca.

Desain Penelitian

Dua model database telah dibangun sebagai entitas komparatif:

Model	Nama	Deskripsi Sistem	Tujuan Desain
A	Skema Normalisasi (3NF)	Data dipisahkan ke tabel customers, products, transactions sesuai kaidah normalisasi	Mengurangi redundansi dan menjaga integritas data [1]
B	Skema Denormalisasi	Data digabung menjadi satu tabel transactions_flat	Mempercepat query karena pengurangan operasi join [2]

Setiap skema dirancang dengan kerangka logis yang identik, namun menunjukkan variasi dalam implementasi fisik.

Desain eksperimental mencerminkan atribut yang melekat dalam sistem transaksi ritel yang mencakup beberapa entitas data, yang tunduk pada akses yang sering dan berulang.



Prosedur Penelitian

Tahap prosedural penelitian dilaksanakan secara bertahap untuk membandingkan kinerja dua desain skema basis data, yaitu skema ternormalisasi (3NF) dan skema terdenormalisasi. Adapun langkah-langkah penelitian yang dilakukan adalah sebagai berikut:

1. Merancang dua model skema basis data, yaitu Model A dengan pendekatan normalisasi hingga Third Normal Form (3NF) dan Model B dengan pendekatan denormalisasi, yang memiliki struktur logis data serupa namun berbeda dalam implementasi fisik.
2. Menghasilkan dataset sintetis menggunakan skrip SQL berbasis generator data. Dataset yang digunakan berjumlah 50.000 record pada masing-masing model basis data untuk memastikan kesetaraan beban pengujian.
3. Menerapkan tiga jenis kueri utama sebagai objek pengujian, meliputi:
 - a) Query seleksi (SELECT) untuk menguji proses pembacaan data,
 - b) Query join yang diterapkan secara eksklusif pada skema ternormalisasi (3NF),
 - c) Query agregasi (SUM dan COUNT) untuk mengukur kinerja pemrosesan data dalam skala besar.
4. Melakukan pengujian pada Database Management System (DBMS) berbasis MySQL/MariaDB dengan menggunakan MySQL Workbench sebagai antarmuka eksekusi kueri.
5. Pengukuran waktu eksekusi kueri dilakukan melalui fitur profiling pada sesi database. Setiap kueri dijalankan dalam kondisi sistem yang sama, dan nilai waktu eksekusi yang dihasilkan digunakan sebagai dasar perbandingan kinerja antar skema.
6. Evaluasi kinerja dilakukan berdasarkan dua parameter utama, yaitu waktu eksekusi kueri (execution time) dan pembacaan logis (logical reads) yang diperoleh dari analisis rencana eksekusi kueri.
7. Hasil pengujian dianalisis dan disajikan dalam bentuk tabel perbandingan kinerja untuk mengidentifikasi perbedaan performa antara skema ternormalisasi dan terdenormalisasi.

Akuisisi Data dan Teknik Pengujian

Akuisisi data dan teknik pengujian dalam penelitian ini dirancang untuk memastikan hasil yang objektif dan dapat direplikasi. Dataset yang digunakan merupakan data sintetis yang dihasilkan secara otomatis menggunakan skrip SQL, sehingga tidak bergantung pada data riil dan dapat digunakan kembali pada pengujian selanjutnya.

Pengujian dilakukan pada lingkungan DBMS berbasis MySQL/MariaDB yang dijalankan melalui XAMPP, dengan MySQL Workbench sebagai alat bantu untuk eksekusi kueri dan pemantauan hasil. Seluruh kueri dijalankan pada masing-masing skema basis data sesuai dengan model yang diuji.

Pengukuran kinerja database dilakukan menggunakan beberapa teknik, antara lain:

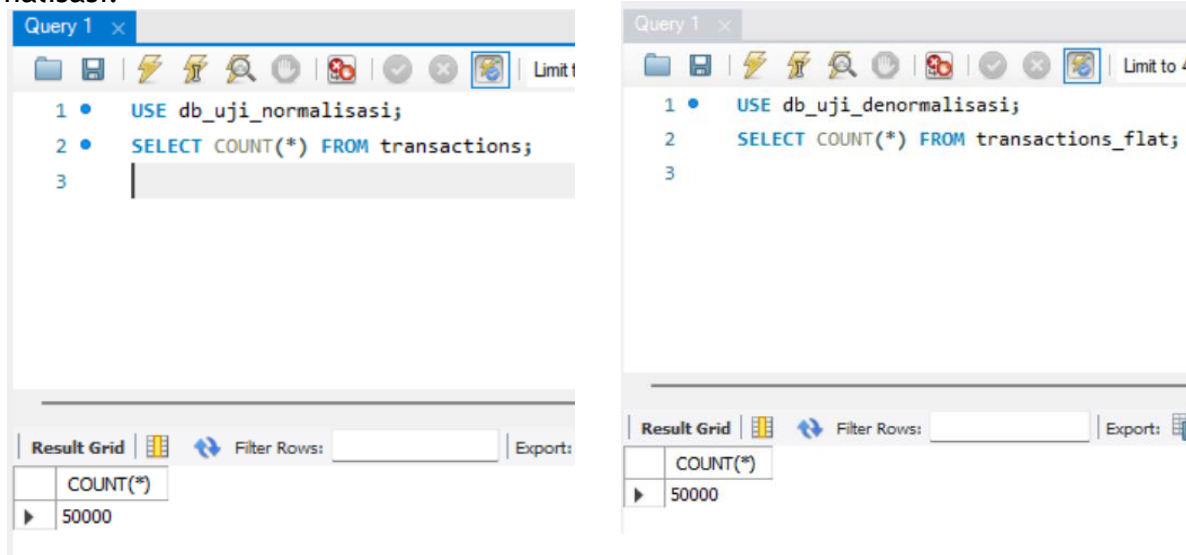
1. Pengukuran waktu eksekusi kueri dengan memanfaatkan fitur profiling pada sesi database untuk mencatat durasi eksekusi setiap kueri.
2. Analisis pembacaan logis (logical reads) dilakukan menggunakan perintah EXPLAIN untuk mengamati rencana eksekusi kueri, termasuk jumlah baris yang diproses, tipe akses data, dan keberadaan operasi join.
3. Verifikasi jumlah data dilakukan menggunakan query COUNT untuk
4. memastikan konsistensi jumlah record pada setiap skema basis data.

Parameter evaluasi yang digunakan dalam penelitian ini meliputi waktu eksekusi kueri dan pembacaan logis sebagai indikator utama kinerja sistem basis data. Pendekatan ini dipilih karena mampu memberikan gambaran yang representatif terhadap efisiensi eksekusi kueri pada masing-masing desain skema.

HASIL DAN PEMBAHASAN

Hasil Pengujian Dataset

Pengujian dilakukan pada dua model basis data, yaitu database ternormalisasi dan database terdenormalisasi. Untuk memastikan objektivitas pengujian, kedua model basis data menggunakan jumlah data yang sama, yaitu sebanyak 50.000 record. Verifikasi jumlah data dilakukan menggunakan perintah `SELECT COUNT(\*)` pada masing-masing tabel utama, yaitu tabel `transactions` pada database normalisasi dan tabel `transactions\_flat` pada database denormalisasi.



Gambar 1 Verifikasi jumlah data pada database ternormalisasi dan terdenormalisasi.

Berdasarkan hasil pengujian, kedua database terbukti memiliki jumlah data yang identik, yaitu masing-masing sebanyak 50.000 record. Dengan kesetaraan jumlah data tersebut, perbandingan performa yang dilakukan pada tahap selanjutnya dapat dikatakan adil dan tidak dipengaruhi oleh perbedaan ukuran dataset, melainkan murni disebabkan oleh perbedaan desain struktur basis data.

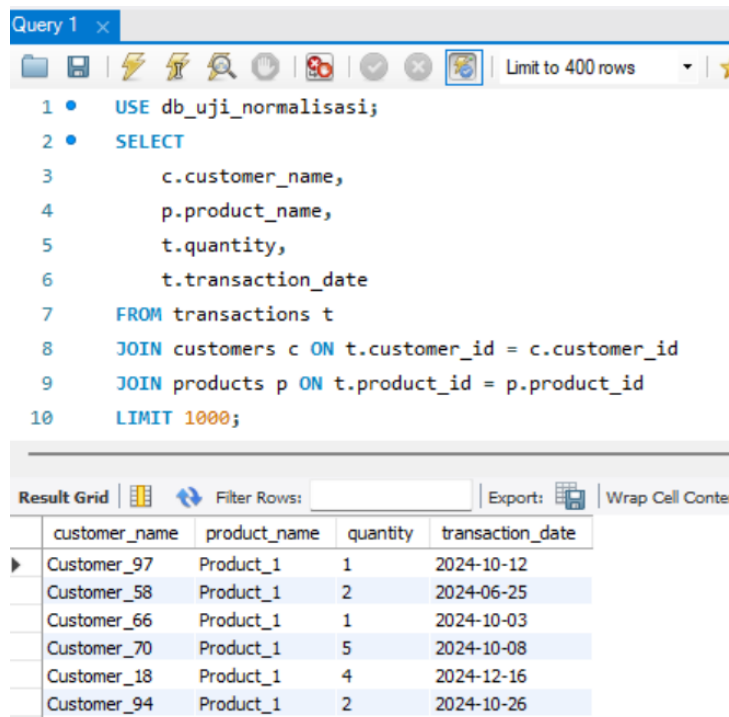
Hasil Query SELECT (Baca Data)

Pengujian awal dilakukan dengan menjalankan query SELECT untuk membaca data dari masing-masing model database. Pada database normalisasi, query dijalankan pada tabel `transactions`, sedangkan pada database denormalisasi query dijalankan pada tabel `transactions\_flat`.

Hasil pengujian menunjukkan bahwa kedua database mampu menampilkan data dengan baik. Namun, terdapat perbedaan struktur data yang signifikan. Database normalisasi menampilkan data transaksi dalam bentuk relasi antar tabel, sedangkan database denormalisasi menyimpan seluruh informasi transaksi dalam satu tabel tanpa memerlukan relasi tambahan. Perbedaan struktur ini menjadi dasar perbedaan performa pada pengujian selanjutnya.

Hasil Query pada Database Normalisasi

Pada database ternormalisasi, pengujian dilanjutkan dengan menjalankan query JOIN yang menggabungkan tabel `transactions`, `customers`, dan `products`. Query ini bertujuan untuk memperoleh informasi transaksi secara lengkap, termasuk nama pelanggan dan nama produk, yang pada desain normalisasi disimpan pada tabel yang berbeda.



Query 1

```

1 • USE db_uji_normalisasi;
2 • SELECT
3     c.customer_name,
4     p.product_name,
5     t.quantity,
6     t.transaction_date
7 FROM transactions t
8 JOIN customers c ON t.customer_id = c.customer_id
9 JOIN products p ON t.product_id = p.product_id
10 LIMIT 1000;

```

Result Grid

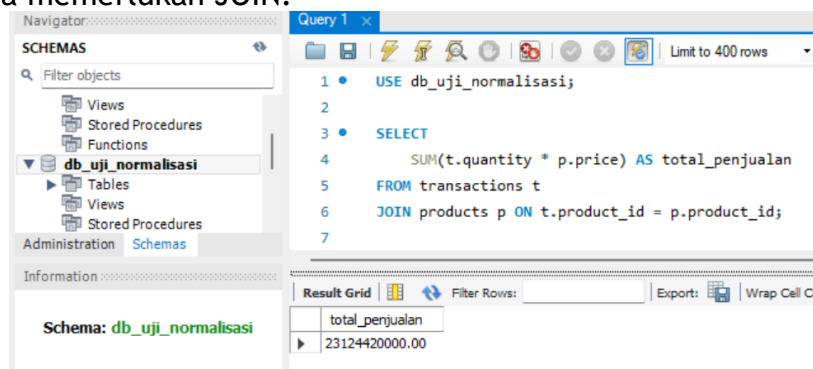
customer_name	product_name	quantity	transaction_date
Customer_97	Product_1	1	2024-10-12
Customer_58	Product_1	2	2024-06-25
Customer_66	Product_1	1	2024-10-03
Customer_70	Product_1	5	2024-10-08
Customer_18	Product_1	4	2024-12-16
Customer_94	Product_1	2	2024-10-26

Gambar 2 Hasil query JOIN pada database ternormalisasi.

Hasil pengujian menunjukkan bahwa database normalisasi memerlukan operasi JOIN untuk menghasilkan data transaksi yang komprehensif. Proses JOIN ini menyebabkan sistem harus melakukan pembacaan data dari beberapa tabel secara bersamaan, sehingga meningkatkan kompleksitas eksekusi query. Sebaliknya, pada database denormalisasi, operasi JOIN tidak diperlukan karena seluruh informasi transaksi telah digabungkan ke dalam satu tabel. Perbedaan inilah yang menjadi faktor utama dalam perbedaan performa pada pengujian selanjutnya.

Hasil Query Agregasi

Pengujian ini dilakukan menggunakan query agregasi untuk menghitung total penjualan. Pada database normalisasi, perhitungan total penjualan dilakukan dengan mengalikan jumlah transaksi dengan harga produk melalui operasi JOIN antara tabel `transactions` dan `products`. Sementara itu, pada database denormalisasi, perhitungan dilakukan langsung pada satu tabel tanpa memerlukan JOIN.



Navigator

SCHEMAS

Filter objects

- Views
- Stored Procedures
- Functions
- db_uji_normalisasi
 - Tables
 - Views
 - Stored Procedures

Administration Schemas

Information

Schema: db_uji_normalisasi

Query 1

```

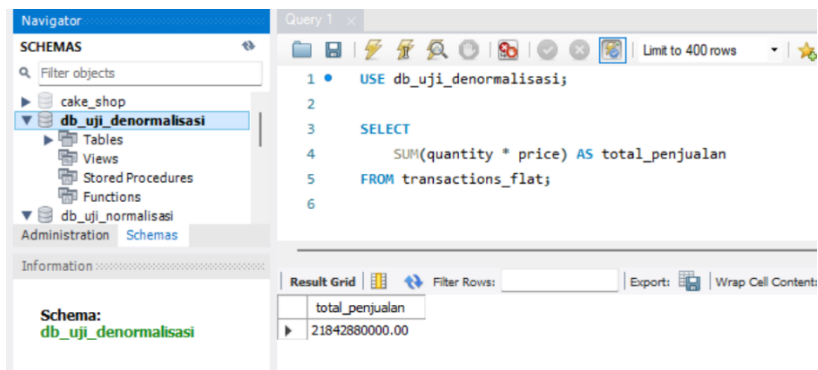
1 • USE db_uji_normalisasi;
2 •
3 • SELECT
4     SUM(t.quantity * p.price) AS total_penjualan
5 FROM transactions t
6 JOIN products p ON t.product_id = p.product_id;
7

```

Result Grid

total_penjualan
23124420000.00

Gambar 3 Hasil query agregasi pada database ternormalisasi.

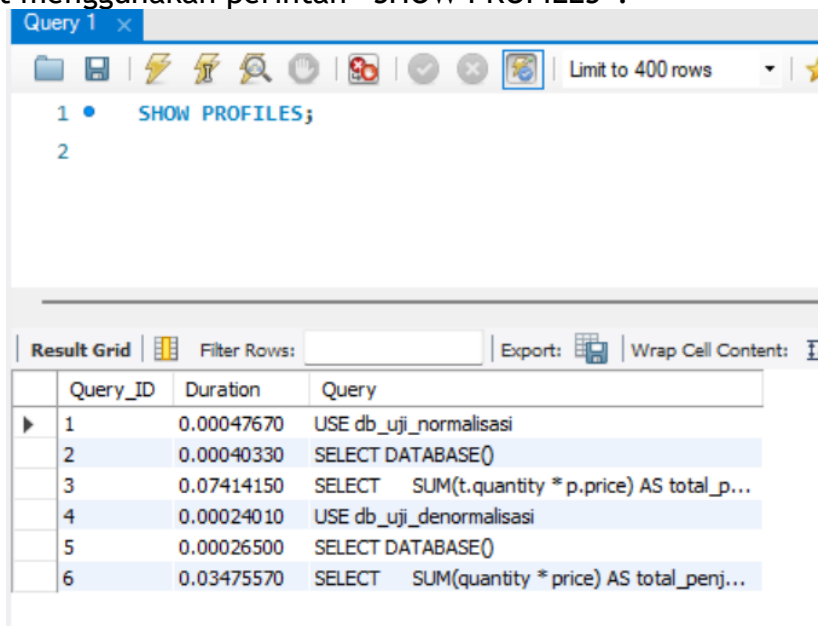


Gambar 4 Hasil query agregasi pada database terdenormalisasi.

Hasil pengujian menunjukkan bahwa kedua model database mampu menghasilkan nilai total penjualan. Perbedaan nilai total yang diperoleh disebabkan oleh penggunaan data sintetis yang dihasilkan secara acak. Oleh karena itu, fokus utama pengujian ini tidak terletak pada nilai hasil agregasi, melainkan pada perbedaan performa eksekusi query yang dihasilkan oleh masing-masing desain basis data.

Analisis Waktu Eksekusi Query

Pengukuran waktu eksekusi query dilakukan dengan mengaktifkan fitur profiling pada sesi database. Query agregasi dijalankan pada masing-masing model database, kemudian waktu eksekusi dicatat menggunakan perintah `SHOW PROFILES`.



The screenshot shows a 'Query 1' window with the command `SHOW PROFILES;`. Below it, a 'Result Grid' displays the following data:

Query_ID	Duration	Query
1	0.00047670	USE db_uji_normalisasi
2	0.00040330	SELECT DATABASE()
3	0.07414150	SELECT SUM(t.quantity * p.price) AS total_p...
4	0.00024010	USE db_uji_denormalisasi
5	0.00026500	SELECT DATABASE()
6	0.03475570	SELECT SUM(quantity * price) AS total_penj...

Gambar 5 Hasil pengukuran waktu eksekusi query menggunakan profiling.

Berdasarkan hasil profiling, database denormalisasi menunjukkan waktu eksekusi yang lebih cepat dibandingkan database normalisasi. Hal ini disebabkan oleh tidak adanya operasi JOIN pada database denormalisasi, sehingga sistem hanya perlu membaca satu tabel untuk memproses query agregasi. Sebaliknya, database normalisasi memerlukan waktu eksekusi yang lebih lama karena harus melakukan penggabungan data dari beberapa tabel sebelum menghasilkan hasil akhir. Meskipun demikian, perbedaan waktu eksekusi yang terjadi masih berada dalam batas wajar dan tidak menyebabkan kegagalan sistem.

Analisis Rencana Eksekusi

Analisis rencana eksekusi query dilakukan menggunakan perintah `EXPLAIN` untuk mengetahui cara sistem manajemen basis data memproses query yang dijalankan. Pada database normalisasi, hasil EXPLAIN menunjukkan adanya operasi JOIN dengan jumlah baris yang diproses lebih banyak. Hal ini mengindikasikan bahwa sistem harus membaca data dari beberapa tabel sebelum menghasilkan hasil akhir.

Query 1

```

1 • USE db_uji_normalisasi;
2
3 • EXPLAIN
4 SELECT
5     SUM(t.quantity * p.price)
6 FROM transactions t
7 JOIN products p ON t.product_id = p.product_id;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
▶	1	SIMPLE	p	ALL	PRIMARY	NULL	NULL	NULL	20	
	1	SIMPLE	t	ref	product_id	product_id	5	db_uji_normalisasi.p.product_id	1327	

Gambar 6 Rencana eksekusi query pada database ternormalisasi.

Query 1

```

1 • USE db_uji_denormalisasi;
2
3 EXPLAIN
4 SELECT
5     SUM(quantity * price)
6 FROM transactions_flat;
7

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
▶	1	SIMPLE	transactions_flat	ALL	NULL	NULL	NULL	NULL	49921	

Gambar 7 Rencana eksekusi query pada database terdenormalisasi.

Pada database denormalisasi, hasil EXPLAIN menunjukkan bahwa query dijalankan tanpa operasi JOIN dan hanya melibatkan satu tabel. Jumlah baris yang diproses lebih sederhana, sehingga rencana eksekusi menjadi lebih efisien. Hasil analisis ini memperkuat temuan sebelumnya bahwa denormalisasi dapat meningkatkan performa query baca, terutama pada query agregasi.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil pengujian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa penerapan normalisasi dan denormalisasi memiliki pengaruh yang berbeda terhadap performa database. Database ternormalisasi mampu menjaga konsistensi dan integritas data dengan baik, namun memerlukan operasi JOIN yang dapat meningkatkan waktu eksekusi query, terutama pada query agregasi.

Sebaliknya, database terdenormalisasi menunjukkan performa yang lebih baik pada operasi baca dan query agregasi karena tidak memerlukan JOIN antar tabel. Namun, pendekatan ini berpotensi menimbulkan redundansi data dan membutuhkan perhatian lebih dalam pengelolaan konsistensi data.



Dengan demikian, pemilihan antara normalisasi dan denormalisasi sebaiknya disesuaikan dengan kebutuhan sistem. Untuk sistem yang berfokus pada transaksi dan konsistensi data, normalisasi lebih disarankan. Sementara itu, untuk sistem yang berorientasi pada pelaporan dan analisis data, denormalisasi dapat menjadi solusi yang lebih efisien.

Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan selanjutnya. Penelitian berikutnya dapat menguji performa database dengan jumlah data yang lebih besar atau pada lingkungan multi-user untuk memperoleh hasil yang lebih komprehensif. Selain itu, pengujian dapat dikembangkan dengan menambahkan indeks atau partisi tabel untuk melihat pengaruhnya terhadap performa database.

Penggunaan sistem manajemen basis data yang berbeda juga dapat menjadi bahan perbandingan agar diperoleh gambaran performa yang lebih luas. Dengan pengembangan tersebut, hasil penelitian diharapkan dapat memberikan kontribusi yang lebih mendalam dalam perancangan basis data yang optimal

DAFTAR PUSTAKA

- [1] M. Hardini et al., "Application of Database Normalization in Increasing Data Storage Efficiency," *International Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 201-211, 2025.
- [2] D. Widodo, "Mengelola Histori Data dengan Pendekatan Denormalisasi Database," *Teknomatika*, vol. 4, no. 1, pp. 89-93, 2011.
- [3] DataCamp, "Database Denormalization: What It Is and When To Use It," 2023. [Online]. Available: <https://www.datacamp.com/tutorial/denormalization>
- [4] sBINUS University, "Pentingnya Melakukan Denormalisasi pada Perancangan Sistem," 2020. [Online]. Available: <https://sis.binus.ac.id/2020/06/15/pentingnya-melakukan-denormalisasi-pada-perancangan-sistem/>
- [5] Visual Paradigm, "Balancing Data Integrity and Performance: Normalization vs Denormalization in Database Design," 2022. [Online]. Available: <https://guides.visual-paradigm.com/balancing-data-integrity-and-performance-normalization-vs-denormalization-in-database-design/>