IMPLEMENTASI TRIGGER DAN OPTIMALISASI INDEXING UNTUK AUDIT TRAIL BASIS DATA PENJUALAN

Rahma Doni, Mhd Arief Hasan*, Roy Ganda Malau,
Andri Yanto Simatupang, Hengki Saputra Zega, Ricko Oktanta Ginting
Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Lancang Kuning
Jl. Yos Sudarso No.KM. 8, Umban Sari, Kec. Rumbai, Kota Pekanbaru, Riau 28266
m.arif@uilak.ac.id

ABSTRAK

Perkembangan sistem informasi penjualan berbasis basis data meningkatkan efisiensi pengelolaan transaksi, namun juga menimbulkan risiko kesalahan input, manipulasi data, dan penyalahgunaan akses. Permasalahan utama yang muncul adalah kurangnya mekanisme pencatatan perubahan data secara otomatis serta menurunnya kinerja sistem akibat pertambahan data audit yang terus meningkat. Penelitian ini bertujuan mengimplementasikan audit trail berbasis trigger dan menganalisis pengaruh penggunaan indexing terhadap kinerja query pada sistem basis data penjualan. Metode yang digunakan adalah metode eksperimental dengan membandingkan waktu eksekusi query audit sebelum dan sesudah penerapan non-clustered index. Audit trail dibangun menggunakan tabel log dan trigger untuk mencatat setiap operasi INSERT, UPDATE, dan DELETE. Pengujian dilakukan lima kali pada masing-masing kondisi menggunakan parameter waktu eksekusi query. Hasil penelitian menunjukkan rata-rata waktu eksekusi query tanpa index sebesar 61,8 ms, sedangkan dengan index sebesar 55 ms, yang berarti terjadi peningkatan kinerja sekitar 11%. Hasil ini membuktikan bahwa kombinasi audit trail berbasis trigger dan optimasi indexing mampu meningkatkan transparansi, keamanan, serta efisiensi akses data pada sistem basis data penjualan.

***Kata kunci**: audit basis data, trigger, audit trail, indexing, kinerja query*

1. PENDAHULUAN

Perkembangan sistem informasi penjualan berbasis basis data telah memberikan kemudahan dalam pengelolaan transaksi, pengendalian stok, serta penyusunan laporan keuangan secara cepat dan terstruktur. Basis data berperan sebagai pusat penyimpanan dan pengolahan data yang mendukung operasional organisasi. Namun, meningkatnya jumlah transaksi dan aktivitas pengguna juga menimbulkan risiko baru, seperti kesalahan input, manipulasi data, serta penyalahgunaan hak akses. Kondisi ini dapat berdampak pada menurunnya keakuratan data serta berkurangnya tingkat kepercayaan terhadap sistem informasi yang digunakan [1].

Untuk mengatasi permasalahan tersebut, diperlukan suatu mekanisme pengawasan yang mampu mencatat setiap perubahan data secara otomatis dan sistematis. Salah satu pendekatan yang dapat diterapkan adalah audit trail pada basis data. Audit trail memungkinkan seluruh aktivitas perubahan data, seperti perintah INSERT, UPDATE, dan DELETE, direkam ke dalam tabel log sehingga setiap perubahan dapat ditelusuri kembali. Implementasi audit trail dapat dilakukan menggunakan trigger pada basis data, yang bekerja secara otomatis tanpa bergantung pada aplikasi. Dengan adanya mekanisme ini, sistem memiliki jejak digital yang dapat digunakan untuk keperluan pengawasan, evaluasi, maupun investigasi apabila terjadi kesalahan atau penyalahgunaan data [2].

Meskipun audit trail meningkatkan transparansi dan keamanan sistem, penerapannya juga menimbulkan tantangan baru pada aspek kinerja basis data. Tabel audit log akan terus bertambah seiring meningkatnya aktivitas transaksi, sehingga proses pencarian dan analisis data audit berpotensi menjadi lebih lambat. Jika tidak diimbangi dengan teknik optimasi yang tepat, pertumbuhan data log dapat menyebabkan penurunan performa query dan membebani sistem. Beberapa penelitian menunjukkan bahwa query pada tabel berukuran besar tanpa optimasi dapat menyebabkan proses pencarian menjadi tidak efisien[3][4].

Salah satu teknik optimasi yang umum digunakan untuk meningkatkan kinerja query adalah indexing. Index berfungsi mempercepat proses pencarian data dengan mengurangi kebutuhan pemindaian seluruh tabel. Penggunaan index pada kolom yang sering digunakan dalam proses pencarian terbukti mampu meningkatkan efisiensi akses data secara signifikan [5][6]. Oleh karena itu, penerapan indexing menjadi penting dalam mendukung kinerja sistem audit basis data, terutama ketika volume data log terus meningkat.

Berdasarkan kondisi tersebut, penelitian ini difokuskan pada implementasi audit trail berbasis trigger pada sistem basis data penjualan serta analisis pengaruh penerapan indexing terhadap kinerja query audit. Penelitian ini bertujuan untuk membangun mekanisme pencatatan perubahan data yang otomatis dan konsisten, sekaligus mengukur sejauh mana penggunaan non-clustered index dapat meningkatkan efisiensi akses data audit. Dengan demikian, penelitian ini tidak hanya menekankan aspek keamanan dan transparansi data, tetapi juga

mempertimbangkan kinerja sistem agar tetap optimal meskipun volume data audit terus bertambah.

Untuk mencapai tujuan tersebut, penelitian dilakukan melalui beberapa tahapan yang terstruktur. Proses diawali dengan perancangan tabel audit log yang berfungsi menyimpan histori perubahan data. Selanjutnya, trigger diimplementasikan pada tabel transaksi untuk mencatat setiap aktivitas INSERT, UPDATE, dan DELETE secara otomatis. Setelah sistem audit berjalan, dilakukan pengujian kinerja query audit tanpa penggunaan index sebagai kondisi awal. Tahap berikutnya adalah penerapan non-clustered index pada kolom-kolom yang sering digunakan dalam pencarian data audit, seperti identitas pengguna dan waktu perubahan. Pengujian kinerja kemudian dilakukan kembali untuk membandingkan waktu eksekusi query sebelum dan sesudah optimasi. Hasil pengujian tersebut dianalisis untuk mengetahui pengaruh penerapan indexing terhadap efisiensi sistem audit basis data penjualan [7].

2. TINJAUAN PUSTAKA

2.1. Audit Basis Data

Audit basis data merupakan proses pengawasan dan pencatatan aktivitas yang terjadi pada sistem basis data untuk memastikan bahwa data dikelola secara aman, akurat, dan dapat dipertanggungjawabkan. Audit basis data berfungsi untuk mendeteksi kesalahan, penyalahgunaan hak akses, serta manipulasi data yang dapat merugikan organisasi.

Selain audit basis data teknis, audit sistem informasi juga banyak diterapkan untuk mengevaluasi tata kelola dan pengendalian sistem. Apriza dan Bina [1] menunjukkan bahwa pengelolaan basis data yang baik harus memperhatikan aspek keamanan, pengendalian akses, serta optimasi query agar sistem tetap andal dan efisien.

Pada konteks sistem informasi akademik, Anggraini [8] membuktikan bahwa framework COBIT 5 pada domain EDM05, APO04, dan BAI10 mampu mengukur tingkat kematangan pengelolaan sistem dan mendeteksi kelemahan pengendalian.

Hasil serupa juga diperoleh oleh Galasca [9] pada audit PORTALSIA, yang menunjukkan bahwa penerapan domain DSS dan MEA COBIT 5 dapat meningkatkan pengawasan, keandalan, dan kualitas layanan sistem informasi akademik.

Di Indonesia, audit sistem informasi berbasis basis data telah banyak diterapkan pada berbagai sektor. Angelo [10] menjelaskan bahwa audit sistem informasi pada sektor perbankan berperan penting dalam menjaga keandalan dan integritas data transaksi.

Penelitian lain oleh Ningsih [11] menunjukkan bahwa audit sistem informasi pada layanan perpustakaan mampu meningkatkan kualitas pengelolaan data dan transparansi layanan .

Pratana Putra et al. [12] juga menegaskan bahwa audit sistem informasi pada sistem jurnal elektronik membantu mengidentifikasi kelemahan pengendalian akses dan potensi penyalahgunaan data. Siahaan et al.[13] menemukan bahwa audit pada sistem penerimaan mahasiswa baru dapat mencegah manipulasi data pendaftaran dan meningkatkan kepercayaan pengguna terhadap sistem.

Berdasarkan penelitian-penelitian tersebut, audit basis data dapat disimpulkan sebagai komponen penting dalam menjaga keamanan, keakuratan, dan keandalan sistem informasi berbasis basis data.

2.2. Audit Trail

Audit trail merupakan mekanisme pencatatan seluruh aktivitas yang terjadi pada sistem, khususnya aktivitas perubahan data dalam basis data. Audit trail menyimpan informasi mengenai siapa yang melakukan perubahan, kapan perubahan dilakukan, dan data apa yang diubah. Informasi ini sangat penting untuk keperluan pengawasan, forensik digital, dan evaluasi sistem.

Yulisara [2] menyatakan bahwa audit trail pada sistem pemerintahan mampu meningkatkan transparansi dan akuntabilitas pengelolaan data.

Penelitian Putri [14] juga menunjukkan bahwa penerapan audit trail dalam sistem kepegawaian daerah membantu organisasi dalam melakukan pengawasan dan evaluasi terhadap aktivitas pengguna sistem.

Dalam konteks sistem informasi penjualan, audit trail menjadi sangat penting karena data transaksi bersifat sensitif dan bernilai tinggi. Dengan adanya audit trail, setiap perubahan data transaksi dapat ditelusuri sehingga meminimalkan risiko kecurangan dan kesalahan input.

2.3. Trigger dalam Basis Data

Trigger merupakan prosedur khusus dalam basis data yang dijalankan secara otomatis ketika terjadi suatu peristiwa tertentu, seperti perintah INSERT, UPDATE, atau DELETE. Trigger sering digunakan untuk mengimplementasikan mekanisme audit trail karena mampu mencatat perubahan data secara langsung tanpa bergantung pada aplikasi.

Firmansyah [3] menyatakan bahwa penggunaan mekanisme otomatis dalam basis data, seperti trigger dan prosedur tersimpan, mampu meningkatkan konsistensi pencatatan data dan mengurangi risiko manipulasi. Penelitian lain oleh Sugiarto[7] menunjukkan bahwa pemanfaatan mekanisme internal basis data, termasuk trigger, membantu menjaga integritas data pada sistem transaksi berskala besar .

Dengan menggunakan trigger, setiap perubahan pada tabel utama dapat secara otomatis disalin ke tabel audit log, sehingga sistem audit basis data dapat berjalan secara konsisten dan real-time.

2.4. Indexing

Indexing merupakan teknik optimasi basis data yang digunakan untuk mempercepat proses pencarian dan pengambilan data. Index bekerja seperti daftar indeks pada buku, di mana sistem basis data dapat langsung menuju lokasi data yang dicari tanpa harus memindai seluruh tabel.

Panggabean dan Azizah [4] menunjukkan bahwa penerapan indexing pada aplikasi e-commerce mampu menurunkan waktu eksekusi query dan meningkatkan performa sistem secara signifikan Thoib dan Nugraha [5] juga membuktikan bahwa penggunaan full-text index pada MySQL memberikan kinerja pencarian data yang lebih baik dibandingkan tanpa index.

Penelitian oleh Samidi [3] pada sistem rumah sakit menunjukkan bahwa kombinasi table indexing dan query optimization mampu meningkatkan kecepatan pemrosesan data transaksi. Pamungkas [6] menjelaskan bahwa penerapan struktur index yang tepat mampu mengurangi beban pemrosesan query dan meningkatkan responsivitas sistem basis data karena proses pencarian data tidak lagi dilakukan melalui pemindaian seluruh tabel, melainkan melalui mekanisme pencarian berbasis index.

Berdasarkan hasil penelitian tersebut, dapat disimpulkan bahwa indexing merupakan komponen penting dalam mendukung kinerja sistem audit basis data, khususnya ketika tabel audit log terus bertambah seiring meningkatnya aktivitas sistem.

2.5. Penelitian Terdahulu

Penelitian mengenai pengelolaan dan optimasi basis data telah banyak dilakukan, khususnya yang berkaitan dengan keamanan data serta peningkatan kinerja query. Berbagai studi menunjukkan bahwa pertumbuhan volume data dalam sistem informasi menuntut adanya mekanisme pengawasan yang baik sekaligus teknik optimasi agar performa sistem tetap terjaga.

Penelitian yang dilakukan oleh Pamungkas [6] membahas optimalisasi query pada basis data MySQL melalui penerapan index. Hasil penelitian tersebut menunjukkan bahwa penggunaan index pada kolom yang sering digunakan dalam proses pencarian dapat mempercepat waktu eksekusi query secara signifikan dibandingkan dengan tabel tanpa index. Studi ini menegaskan bahwa indexing merupakan salah satu teknik penting dalam meningkatkan efisiensi akses data pada basis data berukuran besar.

Selanjutnya, penelitian oleh Thoib[5] membandingkan performa pencarian data teks dengan dan tanpa penggunaan full-text index pada MySQL. Hasilnya menunjukkan bahwa sistem yang menggunakan teknik indexing memiliki waktu respon yang lebih cepat dan beban pemrosesan yang lebih ringan. Temuan ini memperkuat peran indexing sebagai solusi untuk mengatasi lambatnya proses pencarian data pada tabel dengan jumlah record yang besar.

Dalam konteks pengelolaan basis data secara umum, Firmansyah[3] meneliti optimasi performa query pada sistem layanan finansial berbasis basis data relasional. Penelitian tersebut menunjukkan bahwa penerapan strategi indexing yang tepat mampu meningkatkan efisiensi sistem dan mengurangi waktu pemrosesan data. Hasil serupa juga ditemukan oleh Panggabean dan Azizah [4], yang menyatakan bahwa penggunaan index secara efektif dapat meningkatkan kinerja sistem basis data pada aplikasi dengan intensitas transaksi tinggi.

Sementara itu, penelitian oleh Sugiarto [7] menekankan pentingnya strategi optimasi basis data seperti indexing dan pengelolaan struktur tabel dalam menjaga stabilitas performa sistem. Penelitian ini menunjukkan bahwa kombinasi teknik optimasi dapat memberikan dampak signifikan terhadap peningkatan kecepatan akses data serta efisiensi pemrosesan query.

Di sisi lain, aspek keamanan dan pengawasan data juga menjadi perhatian dalam pengelolaan sistem basis data. Putri [14] membahas pentingnya audit sistem informasi dalam meningkatkan transparansi dan akuntabilitas pengelolaan data. Penelitian tersebut menegaskan bahwa pencatatan aktivitas perubahan data sangat diperlukan untuk meminimalkan risiko penyalahgunaan serta mempermudah proses pelacakan kesalahan dalam sistem informasi.

Berdasarkan berbagai penelitian terdahulu tersebut, dapat disimpulkan bahwa indexing berperan penting dalam meningkatkan performa query, sedangkan mekanisme audit berperan dalam menjaga keamanan dan transparansi pengelolaan data. Namun, sebagian besar penelitian masih membahas kedua aspek tersebut secara terpisah. Oleh karena itu, penelitian ini menggabungkan implementasi audit trail berbasis trigger dengan analisis pengaruh penggunaan indexing terhadap kinerja query pada tabel audit, sehingga diharapkan dapat memberikan solusi yang tidak hanya aman tetapi juga tetap efisien dalam pengelolaan basis data penjualan.

Penelitian lain yang relevan dilakukan oleh Samidi, Iskandar, Fachruroji, Wibowo, dan Khaerani [15] dalam studi Database Tuning in Hospital Applications Using Table Indexing and Query Optimization. Penelitian ini menunjukkan bahwa penerapan indexing pada sistem basis data rumah sakit mampu mempercepat respon query secara signifikan setelah tuning database, sehingga mendukung peran index sebagai teknik optimasi untuk meningkatkan kinerja query pada basis data berukuran besar.

3. METODE PENELITIAN

3.1. Tahapan Penelitian

Gambar 1. Tahapan penelitian

Tahapan penelitian mencakup perancangan tabel audit, implementasi trigger sebagai mekanisme pencatatan otomatis, analisis data audit menggunakan query lanjutan, penerapan non-clustered index pada tabel audit log, pengujian kinerja query, serta analisis hasil pengujian. Seluruh tahapan ini saling terhubung dan membentuk satu kesatuan proses penelitian yang sistematis, sebagaimana ditunjukkan pada Gambar 1 flowchart tahapan penelitian.

3.2. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan eksperimen untuk mengevaluasi kinerja sistem audit basis data pada sistem informasi penjualan. Pendekatan ini dilakukan dengan membandingkan performa sistem sebelum dan sesudah penerapan teknik optimasi berupa indexing pada tabel audit.

Eksperimen dimulai dengan implementasi audit trail menggunakan trigger untuk mencatat perubahan data. Selanjutnya dilakukan pengukuran waktu eksekusi query sebagai kondisi tanpa index, kemudian diterapkan non-clustered index pada kolom yang sering digunakan dalam pencarian data dan dilakukan pengujian ulang. Hasil kedua kondisi tersebut dibandingkan untuk mengetahui pengaruh indexing terhadap peningkatan kinerja query audit..

3.3. Perancangan Tabel Audit

```
CREATE TABLE Audit_DetailTransaksi_Log (
    LogID INT IDENTITY(1,1) PRIMARY KEY,
    Tanggal_Perubahan DATETIME DEFAULT GETDATE(),
    UserID VARCHAR(20),
    Jenis_Aksi VARCHAR(10),
    Data_Lama VARCHAR(MAX),
    Data_Baru VARCHAR(MAX)
);
```

Gambar 2. Tabel audit

Tabel Audit_DetailTransaksi_Log digunakan sebagai mekanisme *audit trail* untuk mencatat setiap perubahan data yang terjadi pada sistem basis data. Tabel ini menyimpan informasi berupa identitas log yang dihasilkan secara otomatis **(**LogID**)**, waktu terjadinya perubahan **(**Tanggal_Perubahan), identitas pengguna yang melakukan perubahan (UserID), serta jenis aksi yang dilakukan seperti *INSERT*, *UPDATE*, atau *DELETE* (Jenis_Aksi). Selain itu, tabel ini juga mencatat kondisi data sebelum dan sesudah perubahan melalui kolom Data_Lama dan Data_Baru**,** sehingga histori perubahan data dapat ditelusuri secara sistematis. Implementasi tabel audit ini bertujuan untuk meningkatkan keamanan data, transparansi, serta mendukung proses pengawasan dan evaluasi pada sistem informasi.

3.4. Implementasi Trigger

Implementasi trigger dilakukan pada tabel transaksi untuk mendukung mekanisme audit trail dalam sistem basis data. Trigger berfungsi untuk secara otomatis mencatat setiap perubahan data yang terjadi, sehingga seluruh aktivitas manipulasi data dapat direkam tanpa memerlukan intervensi manual dari pengguna.

Dalam penelitian ini, trigger dirancang untuk menangani tiga jenis operasi data, yaitu penambahan data (INSERT), perubahan data (UPDATE), dan penghapusan data (DELETE). Setiap kali salah satu operasi tersebut terjadi, sistem secara otomatis menyimpan informasi perubahan ke dalam tabel audit yang telah dirancang sebelumnya. Informasi yang dicatat meliputi jenis aksi, waktu perubahan, identitas pengguna, serta data sebelum dan sesudah perubahan.

Penerapan trigger ini memastikan bahwa sistem memiliki histori perubahan data yang lengkap dan dapat ditelusuri kembali. Mekanisme ini menjadi dasar dalam proses pengujian kinerja query audit pada tahap selanjutnya.

3.5. Implementasi Indexing

Penerapan indexing dilakukan untuk meningkatkan kinerja query pada tabel audit yang terus bertambah seiring dengan aktivitas sistem. Tanpa optimasi, proses pencarian data pada tabel audit berpotensi menjadi lebih lambat karena sistem harus melakukan pemindaian seluruh isi tabel.

Dalam penelitian ini digunakan non-clustered index yang diterapkan pada kolom-kolom yang sering digunakan dalam proses pencarian data audit, seperti kolom identitas pengguna dan waktu perubahan data. Pemilihan kolom tersebut didasarkan pada kebutuhan pengujian query yang berfokus pada pelacakan aktivitas pengguna serta rentang waktu perubahan data.

Penerapan indexing ini bertujuan untuk mempercepat proses pengambilan data dengan menyediakan struktur tambahan yang membantu sistem menemukan data tanpa harus membaca seluruh tabel. Setelah index diterapkan, dilakukan

pengujian kinerja query untuk membandingkan waktu eksekusi sebelum dan sesudah optimasi.

3.6. Parameter dan Metode Pengujian

Pengujian kinerja query audit dalam penelitian ini dilakukan untuk memperoleh data yang objektif dan terukur secara kuantitatif. Parameter utama yang digunakan adalah waktu eksekusi query (execution time) dalam satuan milidetik (ms), karena parameter ini secara langsung mencerminkan kinerja sistem dalam memproses permintaan pencarian data audit.

Pengukuran waktu eksekusi dilakukan menggunakan fitur statistik waktu pada SQL Server. Query yang digunakan dalam pengujian adalah query pencarian data audit berdasarkan kolom UserID dan Tanggal_Perubahan, karena kedua kolom tersebut merupakan atribut yang paling sering digunakan dalam proses audit.

Pengujian dilakukan dalam dua kondisi, yaitu sebelum penerapan index dan setelah penerapan non-clustered index pada kolom UserID dan Tanggal_Perubahan. Setiap kondisi diuji sebanyak lima kali untuk memperoleh hasil yang lebih stabil dan representatif. Nilai waktu eksekusi yang diperoleh kemudian dihitung rata-ratanya menggunakan persamaan berikut:

$$\bar{\bar{T}} = \frac{T_1 + T_2 + T_3 + T_4 + T_5}{n} \quad (1)$$

Dengan:
$\bar{\bar{T}}$ = Rata-rata eksekusi (ms)
T_n = Waktu eksekusi pada pengujian ke-n
n = Jumlah pengujian

Untuk mengetahui tingkat peningkatan kinerja setelah penerapan index, digunakan persamaan:

$$Peningkatan\ Kinerja\ (\%) = \frac{T_{Tanpa\ Index} - T_{dengan\ index}}{T_{Tanpa\ Index}} \times 100\% \quad (2)$$

Metode pengujian dan perhitungan ini digunakan untuk mengevaluasi secara objektif pengaruh penerapan indexing terhadap kinerja query audit pada sistem basis data penjualan.

3.7. Metode Analisis Hasil

Analisis hasil penelitian dilakukan dengan membandingkan kinerja query audit sebelum dan sesudah penerapan indexing. Data yang dianalisis berupa waktu eksekusi query yang telah diperoleh dari proses pengujian pada masing-masing kondisi. Nilai waktu eksekusi dari setiap pengujian dihitung rata-ratanya untuk mendapatkan gambaran kinerja sistem yang lebih stabil dan representatif.

Perbandingan dilakukan dengan melihat selisih rata-rata waktu eksekusi query antara kondisi tanpa index dan dengan index. Selisih tersebut kemudian digunakan untuk menghitung persentase peningkatan kinerja sistem setelah penerapan indexing. Hasil perhitungan ini menjadi dasar dalam menilai efektivitas teknik optimasi yang diterapkan pada tabel audit.

Selain analisis kuantitatif, dilakukan pula analisis deskriptif untuk menjelaskan perubahan kinerja yang terjadi. Analisis ini bertujuan untuk memberikan interpretasi terhadap hasil pengujian, sehingga dapat diketahui sejauh mana indexing berpengaruh dalam mempercepat proses pencarian data audit pada sistem basis data penjualan.

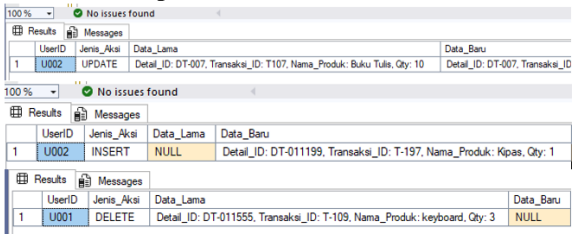
Dengan metode analisis ini, penelitian dapat memberikan kesimpulan yang objektif mengenai hubungan antara penerapan indexing dan peningkatan performa query audit.

4. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi sistem audit basis data serta pengujian kinerja query audit pada dua kondisi, yaitu sebelum dan sesudah penerapan non-clustered index. Hasil tersebut digunakan untuk mengevaluasi kemampuan audit trail berbasis trigger dalam merekam perubahan data serta menilai pengaruh indexing terhadap efisiensi akses data audit.

Pembahasan difokuskan pada tiga hal utama, yaitu perekaman aktivitas perubahan data pada tabel audit, analisis query audit untuk menelusuri histori perubahan, serta perbandingan kinerja query berdasarkan waktu eksekusi. Hasil pengujian disajikan dalam bentuk tabel dan grafik untuk memudahkan analisis. Pembahasan juga dikaitkan dengan konsep basis data seperti table scan dan index seek guna memberikan pemahaman yang lebih komprehensif mengenai dampak optimasi terhadap performa sistem.

4.1. Hasil Implementasi Sistem Audit


100 % No issues found
Results Messages

	UserID	Jenis_Aksi	Data_Lama	Data_Baru
1	U002	UPDATE	Detail_ID: DT-007, Transaksi_ID: T107, Nama_Produk: Buku Tulis, Qty: 10	Detail_ID: DT-007, Transaksi_ID

100 % No issues found
Results Messages

	UserID	Jenis_Aksi	Data_Lama	Data_Baru
1	U002	INSERT	NULL	Detail_ID: DT-011199, Transaksi_ID: T-197, Nama_Produk: Kipas, Qty: 1

Results Messages

	UserID	Jenis_Aksi	Data_Lama	Data_Baru
1	U001	DELETE	Detail_ID: DT-011555, Transaksi_ID: T-109, Nama_Produk: keyboard, Qty: 3	NULL

Gambar 3. Implementasu sitem audit

Implementasi audit trail pada sistem basis data penjualan berhasil merekam setiap aktivitas perubahan data yang terjadi pada tabel transaksi. Mekanisme trigger yang telah diterapkan secara otomatis mencatat operasi INSERT, UPDATE, dan DELETE ke dalam tabel audit tanpa memerlukan intervensi pengguna.

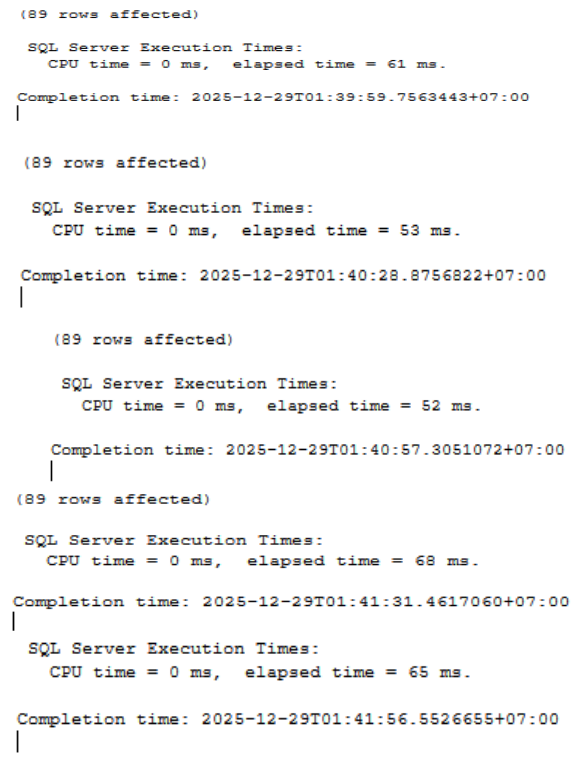
Gambar 3 menunjukkan contoh data yang tersimpan pada tabel audit sebagai hasil dari aktivitas perubahan data. Setiap baris data mencatat informasi penting seperti waktu perubahan, pengguna yang melakukan perubahan, jenis aksi yang dilakukan, serta nilai data sebelum dan sesudah perubahan. Informasi ini membuktikan bahwa sistem audit

mampu menyimpan histori perubahan data secara lengkap dan terstruktur.

Keberhasilan pencatatan ini menunjukkan bahwa mekanisme audit trail berbasis trigger telah berjalan sesuai dengan perancangan pada Bab III. Data audit yang dihasilkan selanjutnya digunakan sebagai objek dalam pengujian kinerja query untuk mengevaluasi pengaruh penerapan indexing terhadap efisiensi akses data.

4.2. Hasil Pengujian Tanpa Index

Pada kondisi tanpa index, SQL Server harus melakukan *Table Scan*, yaitu membaca seluruh baris dalam tabel log untuk menemukan data dengan UserID tertentu. Hal ini menyebabkan waktu eksekusi menjadi lebih lama, terutama ketika ukuran tabel audit semakin besar.



Gambar 4. Hasil Tanpa index

Untuk mengetahui kinerja awal query audit tanpa optimasi, dilakukan pengujian waktu eksekusi query pada tabel Audit_DetailTransaksi_Log tanpa penerapan index. Pengujian dilakukan sebanyak lima kali menggunakan query pencarian berdasarkan UserID. Hasil pengukuran waktu eksekusi query pada kondisi tanpa index ditampilkan pada Tabel 1**.**

Tabel 1. Waktu eksekusi query audit tanpa Index

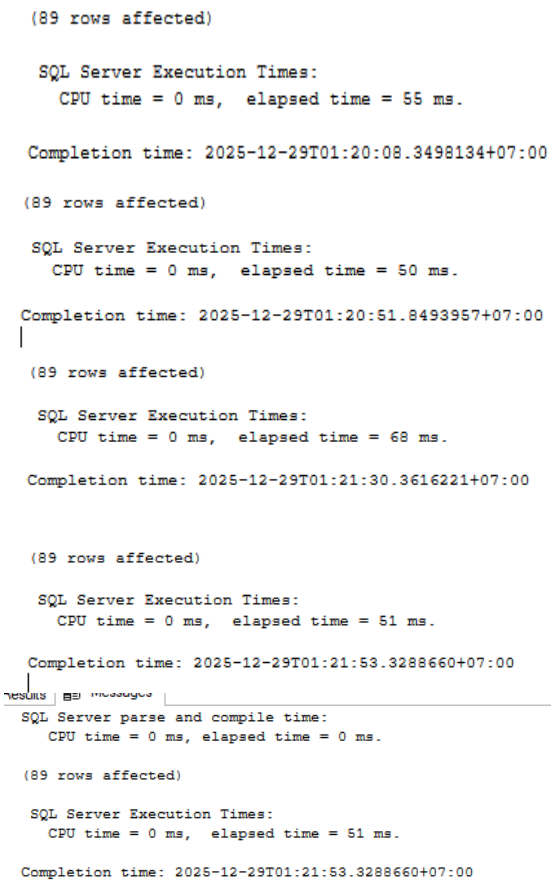
Percobaan	Waktu Eksekusi (ms)
1	61
2	53
3	52
4	68
5	65
Rata-rata	**61,8 ms**

Pengujian kinerja query audit tanpa penggunaan index dilakukan untuk memperoleh gambaran awal performa sistem sebelum optimasi. Query dijalankan sebanyak lima kali dengan kondisi tabel audit belum memiliki index pada kolom UserID. Hasil pengujian menunjukkan waktu eksekusi masing-masing sebesar 61 ms, 53 ms, 52 ms, 68 ms, dan 65 ms, dengan rata-rata waktu eksekusi sebesar 61,8 ms.

Waktu eksekusi yang relatif lebih tinggi terjadi karena sistem melakukan table scan, yaitu memindai seluruh baris tabel audit untuk menemukan data yang sesuai dengan kondisi query. Meskipun jumlah data pada pengujian ini masih terbatas, proses pemindaian tetap dilakukan terhadap seluruh tabel. Hasil ini menunjukkan bahwa tanpa penggunaan index, kinerja query cenderung menurun seiring pertambahan jumlah data audit.

4.3. Hasil dengn Indexing

Setelah non-clustered index diterapkan pada kolom UserID, SQL Server dapat menggunakan *Index Seek* untuk langsung menuju lokasi data yang dicari tanpa harus membaca seluruh tabel log.



Gambar 5. Hasil dengan index

Setelah penerapan non-clustered index pada kolom UserID, pengujian kembali dilakukan untuk mengukur waktu eksekusi query audit. Pengujian dilakukan sebanyak lima kali menggunakan query yang sama seperti pada kondisi tanpa index. Hasil

pengukuran waktu eksekusi query pada kondisi dengan index ditampilkan pada Tabel 2**.**

Tabel 2. Waktu eksekusi query audit dengan index

Percobaan	Waktu Eksekusi (ms)
1	55
2	50
3	68
4	51
5	51
Rata-rata	**55 ms**

Pengujian kinerja query audit dengan penerapan non-clustered index dilakukan untuk melihat pengaruh optimasi terhadap waktu eksekusi. Index dibuat pada kolom UserID, kemudian query yang sama dijalankan kembali sebanyak lima kali. Hasil pengujian menunjukkan waktu eksekusi sebesar 55 ms, 50 ms, 68 ms, 51 ms, dan 51 ms, dengan rata-rata waktu eksekusi sebesar 55 ms.

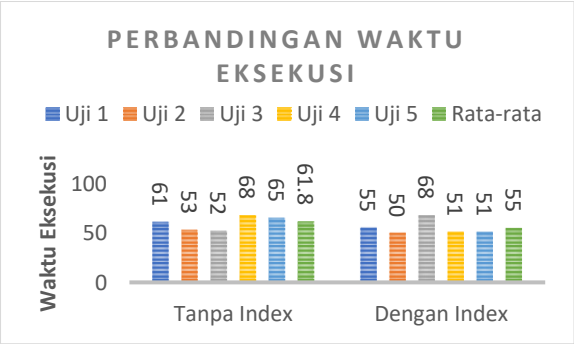
Penurunan waktu eksekusi terjadi karena sistem menggunakan mekanisme index seek untuk langsung menemukan data yang sesuai, tanpa memindai seluruh tabel. Hasil ini menunjukkan bahwa penerapan index mampu meningkatkan efisiensi akses data audit, meskipun jumlah data pada penelitian ini masih relatif terbatas.

4.4 Analisis Perbandingan Kinerja Query

Bagian ini membandingkan rata-rata waktu eksekusi query audit pada kondisi tanpa index dan dengan penerapan non-clustered index.

Tabel 3. Perbandingan kinerja query audit

Kondisi	Data (ms)	Rata-rata
Dengan Index	55, 50, 68, 51, 51	55 ms
Tanpa Index	61, 53, 52, 68, 65	61,8 ms



Gambar 6. Perbandingan waktu eksekusi

Berdasarkan Tabel 3 dan Gambar 6, rata-rata waktu eksekusi query audit tanpa index adalah 61,8 ms, sedangkan dengan index menurun menjadi 55 ms. Selisih waktu sebesar 6,8 ms menunjukkan adanya peningkatan efisiensi setelah penerapan non-clustered index.

Jika dihitung menggunakan rumus peningkatan kinerja:

$$= \frac{61{,}8-55}{61{,}8} \times 100\% \approx 11\% \quad (3)$$

Hasil perhitunan pada persamaan (3) ini menunjukkan bahwa penerapan non-clustered index memberikan peningkatan kinerja sekitar **11%** terhadap query audit. Meskipun peningkatan ini terlihat relatif kecil karena jumlah data audit yang masih terbatas, pada sistem dengan volume data audit yang lebih besar, dampak penggunaan index diperkirakan akan jauh lebih signifikan.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, implementasi audit trail berbasis trigger pada sistem basis data penjualan berhasil merekam seluruh aktivitas perubahan data secara otomatis dan terstruktur. Tabel audit yang dirancang mampu menyimpan informasi penting seperti waktu perubahan, pengguna yang melakukan aksi, jenis operasi, serta nilai data sebelum dan sesudah perubahan, sehingga mendukung kebutuhan pengawasan dan pelacakan histori data.

Pengujian kinerja query audit menunjukkan bahwa penerapan non-clustered index pada kolom yang sering digunakan dalam pencarian data mampu meningkatkan efisiensi akses data. Rata-rata waktu eksekusi query tanpa index sebesar 61,8 ms menurun menjadi 55 ms setelah penerapan index, yang menunjukkan peningkatan kinerja sekitar 11%. Hasil ini membuktikan bahwa teknik indexing berperan penting dalam menjaga performa sistem audit, terutama ketika volume data log terus bertambah.

Pengembangan selanjutnya dapat dilakukan dengan menerapkan mekanisme pengelolaan arsip data audit untuk mengatasi pertumbuhan ukuran tabel log dalam jangka panjang. Selain itu, penelitian berikutnya dapat menguji performa sistem pada jumlah data yang lebih besar serta mengombinasikan teknik optimasi lain seperti partisi tabel atau kompresi data agar efisiensi penyimpanan dan kecepatan akses dapat ditingkatkan secara lebih signifikan.

DAFTAR PUSTAKA

[1] Z. Apriza and U. Bina Darma Tata Sutabri, “Tantangan dan Solusi Pengelolaan Basis Data: Dari Keamanan Hingga Optimalisasi Query,” *Jurnal Sains Student Research*, vol. 3, no. 2, pp. 448–454, 2025, doi: 10.61722/jssr.v3i2.4331.

[2] E. Yulisara, F. Wikron, and N. Alawiyah Hasibuan, “AUDIT SISTEM INFORMASI SRIKANDI PADA KANTOR PEMERINTAHAN MENGGUNAKAN FRAMEWORK COBIT 2019,” *Jurnal Rekayasa Sistem Informasi dan Teknologi*, vol. 2, pp. 1279–1288, 2025.

[3] E. Firmansyah, H. Firdaus, and S. Samidi, “Optimasi Performa Query Subsidi Debitur dengan Index and Table Partition, Subquery and Indexing, dan Parallel Query Execution,” *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 6, pp. 1769–1785, Jul. 2025, doi: 10.52436/1.jpti.824.

[4] S. Panggabean and E. Siti Nur Azizah, "Evaluasi Kinerja Sistem Basis Data Relasional pada Aplikasi E-Commerce Menggunakan Algoritma Indexing," *Jurnal Pustaka Data (Pusat Akses Kajian Database, Analisa Teknologi, dan Arsitektur Komputer)*, vol. 5, no. 1, pp. 70–76, Jun. 2025, doi: 10.55382/jurnalpustakadata.v5i1.998.

[5] Imam Thoib, Beda Puspita Candra, Nafis Sururi, and Danang Satya Nugraha, "Perbandingan Performa Pencarian Data Berbasis Teks dengan Dan Tanpa Full-text Index pada Basis Data MySQL," *INSOLOGI: Jurnal Sains dan Teknologi*, vol. 3, no. 6, pp. 663–673, Dec. 2024, doi: 10.55123/insologi.v3i6.4596.

[6] R. Pamungkas, "OPTIMALISASI QUERY DALAM BASIS DATA MY SQL MENGGUNAKAN INDEX," *Jurnal Komputer, Sistem Informasi, & Manajemen Teknologi*, vol. 1, no. 2, pp. 27–31, 2018.

[7] B. Sugiarto, A. I. Bagusputra, and S. Samidi, "Optimasi Kueri pada Database Oracle Melalui Indeks dan Partisi Tabel untuk Data Besar," *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 7, pp. 1845–1856, Jul. 2025, doi: 10.52436/1.jpti.862.

[8] E. S. Anggraini, M. Aprilsyah, I. N. Hasibuan, L. Asisura, and C. Rizal, "Audit Sistem Informasi Portalsia Menggunakan Framework Cobit 5 Pada Domain EDM05, APO04 dan BAI10," *Jurnal Komputer Teknologi Informasi Sistem Komputer*, vol. 3, pp. 754–762, 2024.

[9] S. A. Galasca, N. Asyiqin, A. H. Haritsyah, and M. R. Handoko, "Audit Portal Sistem Informasi Akademik (PORTALSIA) Menggunakan Framework Cobit 5 Domain DSS dan MEA," *Jurnal Sains, Teknologi & Komputer*, vol. 1, no. 2, pp. 45–50, 2024.

[10] J. Angelo, S. Tania, F. Loren, and Angeline, "Kajian Literatur terhadap Audit Sistem Informasi pada Perusahaan Perbankan," *JDMIS: Journal of Data Mining and Information Systems*, vol. 2, no. 2, pp. 82–89, Aug. 2024, doi: 10.54259/jdmis.v2i2.2216.

[11] Y. Ningsih, S. Alyunita Mega Lestari, I. Kelana Sari, S. Andini, and S. Kaputama, "Audit Sistem Informasi Pelayanan Perpustakaan Binjai Menggunakan Framework Cobit 5," *Jurnal Informatika dan Sains Teknologi*, vol. 1, no. 3, pp. 34–51, 2024.

[12] P. A. Pratama, G. R. Dantes, and G. Indrawan, "AUDIT SISTEM INFORMASI UNIVERSITAS PENDIDIKAN GANESHA DENGAN FRAMEWORK COBIT 5," *Jurnal Sains, Teknologi & Komputer*, vol. 2, pp. 153–156, 2020.

[13] N. B. Siahaan, A. Fadilla Siagian, Z. Syarifudin, and W. A. Fitrah, "Audit Sistem Informasi Penerimaan Mahasiswa Baru Mandiri UINSU Menggunakan Framework COBIT 5 Domain DSS," *Jurnal Komputer Teknologi Informasi Sistem Komputer*, vol. 3, pp. 2962–3022, 2024.

[14] S. A. Putri, M. Khalid, M. Putri, and Z. Fahmi, "Kajian literature review:Inovasi Digital dan Audit Sistem Informasi dalam Pengelolaan Kepegawaian Daerah oleh BKPSDM," *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 4, no. 2, pp. 4178–4188, Jun. 2025, doi: 10.31004/riggs.v4i2.1197.

[15] D. Iskandar, M. Fachruroji, W. Adi Septyo Wibowo, A. A. Khaerani, F. Teknologi Informasi, and U. Budi Luhur, "Database Tuning in Hospital Applications Using Table Indexing and Query Optimization," *Jurnal Pendidikan Tambusai*, vol. 6, pp. 1960–1967, 1960