AUDIT KEAMANAN BASIS DATA MENGGUNAKAN SQL SERVER AUDIT UNTUK DETEKSI AKTIVITAS TIDAK SAH

Padina Di Juni, Mhd Arief Hasan, Yefita Sri Putri Sitompul, Jonatan Alexander, Ibnu Hidayat, Ronaldo Marco Bilbo
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Lancang Kuning
Jl. Yos Sudarso Km. 8 Rumbai, Pekanbaru, Riau
m.arif@unilak.ac.id

ABSTRAK

Keamanan basis data merupakan aspek krusial dalam perlindungan informasi sensitif dari aktivitas tidak sah. Dalam era digital yang semakin berkembang, ancaman terhadap keamanan basis data menjadi lebih kompleks, termasuk risiko akses ilegal, manipulasi data, dan serangan siber. Salah satu metode yang dapat digunakan untuk meningkatkan keamanan adalah dengan menerapkan SQL Server Audit untuk mendeteksi dan mencatat aktivitas mencurigakan pada basis data. Penelitian ini bertujuan untuk mengidentifikasi efektivitas SQL Server Audit dalam mendeteksi aktivitas mencurigakan, menguji skenario audit yang melibatkan akses sah dan tidak sah, serta mengevaluasi implementasi SQL Server Audit dalam meningkatkan keamanan basis data. Pendekatan yang digunakan adalah eksperimental, di mana data simulasi sebanyak 2.515 entri diuji dalam berbagai skenario, termasuk login pengguna, pembaruan, dan penghapusan data. Implementasi melibatkan konfigurasi audit server, pembuatan trigger log, serta analisis log menggunakan Power BI dan Excel. Hasil penelitian menunjukkan bahwa SQL Server Audit mampu mendeteksi aktivitas mencurigakan dengan tingkat akurasi tinggi, mencatat setiap interaksi pengguna secara rinci, serta mengidentifikasi pola akses yang tidak sesuai dengan kebijakan keamanan. Sistem ini juga terbukti andal dalam kondisi beban kerja tinggi dan sesuai dengan standar keamanan ISO 27001. Kesimpulannya, SQL Server Audit merupakan alat yang efektif dalam meningkatkan keamanan basis data dengan memberikan pencatatan audit yang sistematis dan transparan. Studi ini memberikan kontribusi bagi organisasi dalam menerapkan kebijakan keamanan berbasis audit, serta membuka peluang untuk penelitian lanjutan yang mengintegrasikan kecerdasan buatan dalam analisis log secara otomatis.

Kata kunci : *SQL Server Audit, Keamanan Basis Data, Deteksi Aktivitas Tidak Sah, Analisis Log, Audit Basis Data.*

1. PENDAHULUAN

Dalam era digital saat ini, penggunaan *database* telah menjadi hal yang sangat penting dan umum dalam berbagai bidang seperti bisnis, pendidikan, pemerintahan, dan lain-lain. Dalam penggunaannya, *database* menyimpan berbagai informasi sensitif seperti data keuangan, data pribadi, data akademik, dan lain-lain. Oleh karena itu, keamanan *database* menjadi sangat penting untuk melindungi informasi sensitif yang disimpan di dalamnya. Dalam konteks ini, meningkatkan keamanan *database* menjadi prioritas utama, karena semakin kompleksnya teknologi dan semakin banyaknya data yang disimpan, keamanan *database* menjadi tantangan yang semakin besar. Selain itu, serangan *cyber* yang semakin canggih juga membuat keamanan sistem *database* semakin rentan. Oleh karena itu, penelitian tentang sistem keamanan *database* sangatlah relevan dan penting [1], [2].

Analisis *Auditing* akses menjadi hal yang sangat penting, karena mereka memungkinkan organisasi untuk mendeteksi ancaman keamanan seperti deteksi aktivitas tidak sah. Memahami bagaimana data diproses dan siapa yang bertanggung jawab sangat penting. Oleh karena itu, konsep ini tidak boleh diabaikan dalam pengelolaan *database* SQL [3].

Melakukan audit terhadap aktivitas dan akses *database* dapat membantu dalam menemukan potensi masalah keamanan dalam sistem basis data, dan memungkinkan penyelesaiannya dengan cepat. Sebagai fungsi utama, audit memainkan peran kunci dalam memastikan kepatuhan terhadap peraturan dan aturan karena audit melibatkan pemeriksaan dokumen terkait tindakan, praktik, serta perilaku bisnis atau individu [4].

Penelitian sebelumnya telah menganalisis *database logging* dan *auditing*. Penelitian dengan judul Analisis Logging dan Auditing Akses Database SQL Melalui Koneksi Remote Akses Menggunakan Anydesk menggunakan metode NLDC (*Network Development Life Cycle*) yang merupakan teknik analisis terstruktur yang digunakan untuk merencanakan dan mengelola proses pengembangan sistem. Hasil penelitian mendapatkan 2 bagian log yang tercatat yaitu bagian *login* dan bagian *database* atau *transaction log* [3].

Penelitian ini bertujuan untuk mengidentifikasi potensi fitur SQL Server Audit dalam mendeteksi aktivitas mencurigakan, mengimplementasikan scenario audit untuk menguji aktivitas sah dan tidak sah, dan mengevaluasi efektivitas SQL Server Audit dalam meningkatkan keamanan basis data. Penelitian ini juga bertujuan untuk memberikan rekomendasi praktis bagi organisasi yang ingin mengadopsi mekanisme audit sebagai bagian dari strategi keamanan mereka. Dengan demikian, diharapkan hasil

penelitian ini tidak hanya relevan untuk kebutuhan akademis, tetapi juga memberikan kontribusi nyata dalam meningkatkan kesadaran dan praktik keamanan basis data di industri.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Berdasarkan penelitian terdahulu yang dilakukan oleh Aris Susanto, La Ija, dan La Ode Bakrim yang berjudul Audit Sistem Informasi Akademik Berdasarkan Aktivitas Pengguna. Pada proses implementasi sistem informasi, terjadi hal yang tidak diinginkan seperti adanya pihak yang mencari celah yang dimiliki aplikasi dan bertujuan untuk memanipulasi dapat menyebabkan perubahan data pada database. Oleh karena itu, dilakukan pemeriksaan secara forensic untuk mengetahui kejadian manipulasi data tersebut untuk mendapatkan barang bukti digital atau histori data yang telah dimanipulasi atau dihapus. Tujuan proses forensic database untuk memonitoring aktifitas pengguna pada database SIMAK agar dapat mengetahui dan mendeteksi aktifitas pengguna database dan mengetahui siapa yang merubah, data apa yang dirubah, dan kapan dilakukannya [5].

Berdasarkan penelitian terdahulu yang dilakukan oleh I Wayan Gus Arisna, I Made Sukarsa, dan Putu Wira Buana yang berjudul Model Sinkronisasi Database Satu Arah dengan Metode Audit Log Menggunakan Apache Kafka. Pendekatan Audit Log merupakan salah satu objek alternatif yang dapat digunakan dalam mengembangkan sinkronisasi satu arah pada database. Audit Log adalah sebuah metode pencantatan perubahan yang ada pada database dengan menggunakan trigger. Data perubahan database tersebut yang nantinya menjadi acuan dalam sinkronisasi ke database lainnya. Pengiriman perubahan data pada database ke database yang lain memanfaatkan Apache Kafka. Apache merupakan open-source message broker yang memiliki perfoma tinggi sehingga dapat mengirim data secara realtime [6]

2.2. Audit Aktivitas Pengguna

Audit aktivitas pengguna adalah mekanisme yang mencatat setiap interaksi pengguna dengan sistem basis data, termasuk akses, perubahan, dan tindakan lainnya. Data yang dicatat dalam audit ini dapat mencakup informasi seperti waktu akses, identitas pengguna, jenis operasi yang dilakukan, dan hasil dari tindakan tersebut. Penelitian yang dilakukan oleh [5] menunjukkan bahwa audit aktivitas dalam sistem informasi akademik dapat mendeteksi manipulasi data dan menyediakan bukti digital yang relevan untuk investigasi.

2.3. Sinkronisasi Database dengan Audit Log

Metode audit log dalam sinkronisasi database adalah pendekatan yang memanfaatkan catatan perubahan data (log perubahan) sebagai dasar untuk menjaga konsistensi antar database. Audit log adalah mekanisme pencatatan perubahan data yang terjadi dalam sistem, termasuk insert, update, dan delete. Log ini dapat digunakan untuk merekonstruksi perubahan atau mengirimkan pembaruan ke database lain sehingga data tetap tersinkronisasi. Menurut penelitian [6], model sinkronisasi database satu arah dengan Apache Kafka dapat meningkatkan konsistensi data serta memperkuat keamanan melalui mekanisme pencatatan log.

2.4. Audit Keamanan Berdasarkan ISO 27001

Kerangka kerja ISO 27001 banyak digunakan untuk mengevaluasi keamanan sistem informasi. ISO/IEC 27001 adalah standar internasional yang menetapkan persyaratan untuk Sistem Manajemen Keamanan Informasi (*Information Security Management System*/ISMS). Standar ini digunakan oleh organisasi untuk mengelola dan melindungi aset informasi secara sistematis melalui kebijakan, prosedur, dan kontrol keamanan yang diterapkan berdasarkan analisis risiko.

Sebagai contoh, [7] meneliti audit keamanan pada sistem informasi akademik di UNIKOM dan mengidentifikasi control keamanan yang perlu diperbaiki untuk memastikan kesesuaian dengan standar internasional.

2.5. Keamanan Fisik Data

Keamanan fisik data adalah aspek penting dalam perlindungan informasi, yang berfokus pada upaya untuk mencegah akses, pencurian, atau kerusakan terhadap perangkat keras dan infrastruktur tempat data disimpan. Berbeda dengan keamanan digital yang melibatkan perlindungan terhadap serangan siber, keamanan fisik memastikan bahwa perangkat yang menyimpan dan memproses data tetap aman dari ancaman fisik seperti pencurian, bencana alam, atau sabotase.

Studi oleh [8] mengkaji pengelolaan keamanan fisik data menggunakan standar ISO 27001, memberikan panduan yang terstruktur untuk perlindungan fisik data dan infrastruktur TI.

2.6. Strategi Pengamanan Log Server

Penelitian tentang strategi pengamanan log server menyoroti pentingnya penyimpanan log yang terdistribusi dan aman untuk mencegah manipulasi data. Log server adalah komponen penting dalam sistem keamanan informasi karena mencatat berbagai aktivitas dalam sistem, termasuk akses pengguna, perubahan data, dan potensi ancaman keamanan. Oleh karena itu, pengelolaan log yang baik sangat diperlukan untuk memastikan integritas, ketersediaan, dan kerahasiaan data log.

Studi oleh [9] menunjukkan bahwa pengelolaan log yang baik dapat meningkatkan efektivitas deteksi aktivitas mencurigakan dalam sistem basis data.

3. METODE PENELITIAN

Gambar 1. Tahapan Penelitian

Penelitian ini menggunakan pendekatan eksperimental. Pendekatan eksperimental adalah metode yang digunakan untuk mengeksplorasi hubungan sebab-akibat antara variabel dengan melakukan manipulasi pada variabel tertentu [10].

Berdasarkan Gambar 1 pada tahap awal ini akan dilakukan proses penginstallan server dengan menggunakan perangkat lunak Microsoft SQL Server 2019 yang diinstal pada server dengan prosesor Intel Core i3 dan RAM 8 GB, kami menganalisis dataset simulasi yang terdiri dari 2.515 entri dengan data dummy, yang mencakup kolom UserID, Action, Timestamp, dan Status untuk mengevaluasi performa dan efektivitas sistem.

Pada tahap kedua ini akan dilakukan proses implementasi SQL Server Audit. Pada tahap ini, fitur audit dikonfigurasi dengan membuat audit log yang akan mencatat berbagai aktivitas di basis data. Selain itu, dilakukan konfigurasi trigger audit, yaitu mekanisme yang secara otomatis merekam aktivitas tertentu, baik yang sah (authorized) maupun tidak sah (unauthorized). Implementasi ini menjadi fondasi untuk memastikan aktivitas basis data dapat dipantau secara menyeluruh.

Pada tahap ketiga ini akan dilakukan proses pengujian, di mana dilakukan simulasi berbagai skenario aktivitas sah dan tidak sah di basis data. Hasil dari simulasi ini dicatat dalam log audit, yang kemudian dianalisis untuk melihat apakah SQL Server Audit mampu mendeteksi dan mencatat setiap aktivitas secara akurat. Tahap ini bertujuan untuk menguji keandalan mekanisme audit dalam mendeteksi potensi ancaman maupun aktivitas yang tidak sesuai dengan kebijakan keamanan.

Pada tahap keempat/terakhir ini akan dilakukan evaluasi. Fokus dari tahap ini adalah menilai akurasi dan efektivitas SQL Server Audit dalam mencatat aktivitas yang terjadi di basis data. Evaluasi ini penting untuk memahami sejauh mana fitur audit mampu meningkatkan keamanan basis data dengan mengidentifikasi aktivitas mencurigakan secara tepat.

3.1. Pembuatan Basis Data dan Tabel

Proses dimulai dengan mendefinisikan basis data audit DB_{Audit} dan table-tabel $T_{TestData}$ serta $T_{AuditLogin}$. Table-tabel ini berfungsi sebagai penyimpanan data aktivitas basis data dan log login pengguna [11]. Basis data dibuat menggunakan perintah CREATE DATABASE, sementara table dibuat dengan perintah CREATE TABLE yang mendefinisikan kolom-kolom utama seperti UserID, Action, Timestamp, dan Status.

3.2. Pengisian Data Simulasi

Data simulasi dihasilkan menggunakan rumus pengacakan. Nilai $UserID$ dihitung berdasarkan modulus dari fungsi hash terhadap NEWID. Aktivitas ($Action$) ditentukan dengan pembagian modulus 4 untuk menghasilkan Tindakan seperti ‘Login’, ‘Update Data’, ‘Delete Record’, atau ‘Access File’. Waktu kejadian ($Timestamp$) dihitung dengan menambahkan nilai acak dalam menit terhadap tanggal awal. Status ($Status$) dihasilkan secara acak dengan nilai ‘Success’ atau ‘Failed’.

3.3. Konfigurasi Audit Server

Audit server diaktifkan dengan mendefinisikan objek audit $Audit_{Server}$ menggunakan perintah CREATE SERVER AUDIT, yang diarahkan untuk menyimpan log ke file pada direktori tertentu. Status audit server kemudian diaktifkan menggunakan perintah ALTER SERVER AUDIT dengan parameter STATE = ON.

3.4. Pembuatan Log Login dengan Trigger

Untuk mencatat aktivitas login, logon trigger dibuat dengan fungsi $L_{AuditLogin}$. Trigger ini secara otomatis mencatat informasi seperti nama pengguna, waktu login, jenis aktivitas, nama host, dan Alamat IP ke table $T_{AuditLogin}$ setiap kali terjadi percobaan login ke server.

3.5. Verifikasi dan Analisis Log

Data log diverifikasi menggunakan query $Log_{Verifikasi}$, yang membaca file log dengan perintah sys.fn_get_audit_file. Setelah diverifikasi, log dapat diekspor ke alat analitik seperti Power BI atau Excel untuk analisis lebih lanjut**.** Proses analisis ini membantu mengidentifikasi pola aktivitas pengguna atau potensi anomaly.

4. HASIL DAN PEMBAHASAN

4.1. Membuat Basis Data

Proses implementasi dimulai dengan membuat sebuah basis data khusus yang dalam representasi rumus dinyatakan sebagai:

$$DB_{Audit} = CREATE\ DATABASE\ AuditDatabaseKelompok2 \quad (1)$$

Basis data ini bertugas menyimpan seluruh informasi yang berkaitan dengan audit aktivitas pengguna dalam sistem [12].

4.2. Membuat Tabel

Selanjutnya, dua tabel utama dibuat untuk mendukung proses ini. Table pertama, yang disebut TestData, digunakan untuk menyimpan data simulasi aktivitas pengguna [13]. Dalam bentuk rumus, table ini direpresentasikan sebagai:

$$T_{TestData} = CREATE\ TABLE\ (UserID, Action, Timestamp, Status) \quad (2)$$

Tabel kedua, AuditLogin, dirancang untuk mencatat log aktivitas login pengguna [14]. Dalam bentuk rumus, table ini dapat dinyatakan sebagai:

$$T_{AuditLogin} = CREATE\ TABLE\ (AuditID, LoginName, EventTime, EventType, HostName, IPAddress) \quad (3)$$

Dengan kombinasi basis data DB_{Audit} dan table-tabel $T_{TestData}$ serta $T_{AuditLogin}$, sistem audit ini dirancang untuk mencatat, menyimpan, dan menganalisis aktivitas pengguna secara efektif. Table TestData berfungsi untuk menyimulasikan berbagaai aktivitas basis data yang akan diaudit, sementara table AuditLogin mencatat semua aktivitas login yang terjadi dalam sistem.

4.3. Mengisi Dataset Simulasi

Selanjutnya, proses simulasi data dilakukan dengan mengisi table TestData menggunakan data dummy. Representasi matematis dari pengisian tabel ini adalah:

$$D_{TestData} = \bigcup_{i=1}^{n} (UserID_i, Action_i, Timestamp_i, Status_i) \quad (4)$$

Yang menunjukkan bahwa tabel $T_{TestData}$ diisi dengan Kumpulan dari indeks $i = 1$ hingga $i = n$. Setiap baris data terdiri dari empat elemen utama: UserID, Action, Timestamp, dan Status.
Nilai $UserID_i$ dihitung dengan rumus:

$$UserID_i = |CHECKSUM(NEWID)| mod\ 1000 \quad (5)$$

Di mana CHECKSUM(NEWID) menghasilkan angka acak, dan modulus 1000 memastikan bahwa nilai UserID selalu berada dalam rentang 0 hingga 999. Nilai $Action_i$ merepresentasikan tindakan pengguna dan ditentukan berdasarkan nilai modulus dari indeks i terhadap 4:

$$Action_i = \begin{cases} 'Login' & jika\ i\ mod\ 4 = 0 \\ 'Update\ Data' & jika\ i\ mod\ 4 = 1 \\ 'Delete\ Record' & jika\ i\ mod\ 4 = 2 \\ 'Access\ File' & jika\ i\ mod\ 4 = 3 \end{cases}$$

Dengan pembagian ini, tindakan-tindakan seperti login, pembaruan data, penghapusan data, dan akses file dibagi secara merata dalam data simulasi. Waktu kejadian $Timestamp_i$ dihitung menggunakan fungsi DATEADD, yang menambahkan sejumlah menit acak ke tanggal awal tertentu (‘2023-01-01’) :

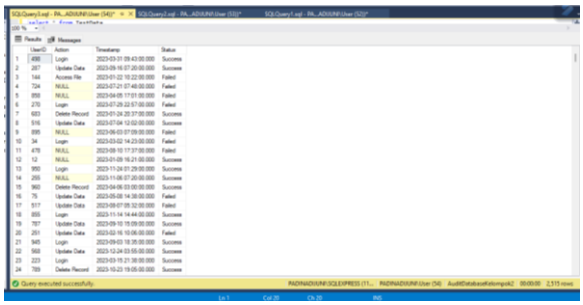
$$Timestamp_i = DATEADD(MINUTE, |CHECKSUM(NEWID)|) \quad (7)$$

Rumus (7) memastikan bahwa setiap tindakan memiliki waktu kejadian unik yang tersebar selama satu tahun (525,600 menit).

Status dari tindakan $Status_i$ ditentukan berdasarkan nilai modulus dari indeks i terhadap 2:

$$Status_i = \begin{cases} 'Success' & jika\ i\ mod\ 2 = 0 \\ 'Failed' & jika\ i\ mod\ 2 = 1 \end{cases} \quad (8)$$

Dengan rumus (8), status berhasil (*Success*) atau gagal (*Failed*) ditentukan secara acak dengan peluang yang sama seperti pada Gambar 2 dibawah ini.



Gambar 2. Tabel TestData

Proses ini menghasilkan tabel data dummy yang lengkap dengan informasi yang beragam dan acak, mencerminkan aktivitas pengguna dalam sistem secara realistis. Dengan data simulasi ini, sistem audit dapat diuji secara efektif untuk memverifikasi bahwa log aktivitas ditangkap dan dianalisis dengan benar.

4.4. Konfigurasi SQL Server Audit

Selanjutnya, proses konfigurasi audit server dimulai dengan mendefinisikan objek audit server, yang direpresentasikan dalam rumus berikut:

$$Audit_{Server} = CREATE\ SERVER\ AUDIT\ (TO\ FILE: 'C:\AuditLogs\ServerAudit.log') \quad (9)$$

Rumus (9) menunjukkan bahwa sebuah objek audit server dibuat menggunakan perintah CREATE SERVER AUDIT. Objek tersebut diarahkan untuk mencatat semua aktivitas yang diaudit ke dalam file

log yang disimpan di direktori C:\AuditLogs\ dengan nama file ServerAudit.log. File log ini bertugas sebagai media penyimpanan utama untuk semua informasi audit yang dihasilkan oleh server.

Setelah objek audit server dibuat, langkah berikutnya adalah mengaktifkannya. Hal ini direpresentasikan dalam rumus:

$$Audit_{ServerState} = ALTER\ SERVER\ AUDIT\ (STATE = ON) \quad (10)$$

Rumus (10) menggunakan perintah ALTER SERVER AUDIT untuk mengubah status audit server menjadi aktif (*ON*). Status aktif memungkinkan sistem untuk mulai mencatat aktivitas secara real-time ke file log yang telah ditentukan.

Kedua langkah ini bekerja secara sinergis, di mana: $Audit_{Server}$ adalah objek yang bertugas merekan semua aktivitas yang diaudit. $Audit_{ServerState}$ memastikan bahwa objek audit server dalam kondisi aktif sehingga dapat mulai mencatat log.

4.5. Pencatatan Log Login

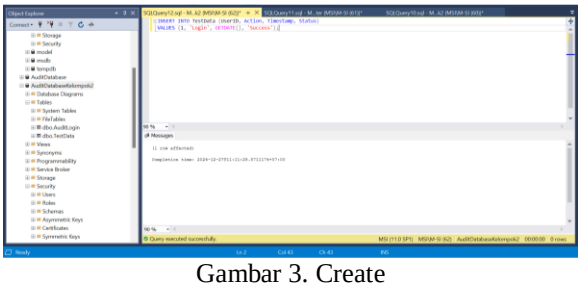
Selanjutnya proses pencatatan log login dalam sistem audit dilakukan dengan menggunakan trigger yang secara otomatis mencatat informasi login ke tabel AuditLogin. Fungsi log login ini direpresentasikan dalam rumus:

$$L_{AuditLogin}(U,T) = \{LoginName, EventTime, EventType, HostName, IPAddress\} \quad (11)$$

Log login ini direalisasikan dengan memasukkan data ke tabel AuditLogin melalui perintah `INSERT INTO`, yang direpresentasikan dalam rumus:

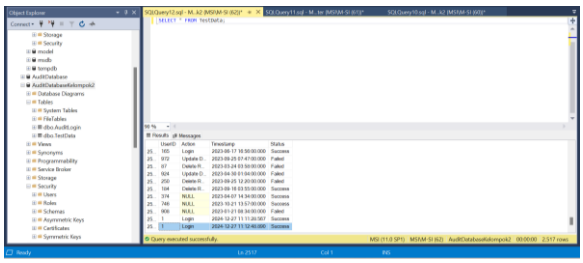
$$L_{AuditLogin}(U,T) = INSERT\ INTO\ T_{AuditLogin} \quad (12)$$

Setiap kali ada aktivitas login di server, trigger secara otomatis mengambil informasi pengguna (UUU) dan waktu kejadian (TTT) untuk disimpan ke dalam tabel audit. Dengan demikian, tabel AuditLogin selalu diperbarui secara real-time tanpa memerlukan campur tangan manual.



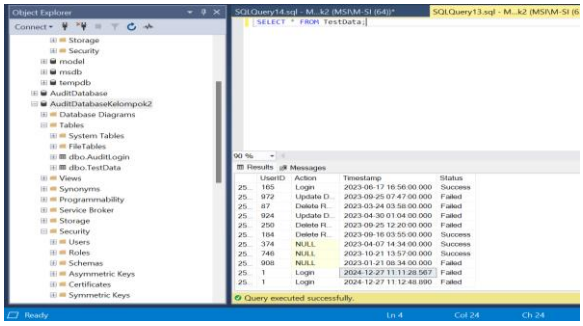
Gambar 3. Create

Gambar 3 menampilkan aktivitas Create dalam sistem, yang mencakup penambahan data baru ke dalam database. Statistik atau grafik pada gambar ini mungkin menunjukkan jumlah operasi create yang berhasil (sah) dibandingkan dengan yang gagal (tidak sah). Tingginya angka aktivitas sah menunjukkan bahwa sistem berhasil menangani proses input data baru secara efektif. Namun, jika terdapat angka aktivitas tidak sah, hal ini bisa menunjukkan potensi masalah validasi input atau upaya memasukkan data yang tidak diotorisasi.



Gambar 4. Read

Gambar 4 berfokus pada aktivitas Read, yang mencerminkan jumlah pengambilan atau akses data dari sistem. Operasi read biasanya lebih sering terjadi dibandingkan operasi lain dalam CRUD, karena akses data adalah fungsi inti dari sebagian besar aplikasi. Grafik atau data yang ditampilkan mungkin menunjukkan tingkat keberhasilan akses (sah) versus upaya akses yang gagal (tidak sah). Jika angka aktivitas tidak sah cukup tinggi, ini dapat menjadi indikasi upaya akses ilegal atau ketidakcocokan izin akses pengguna.

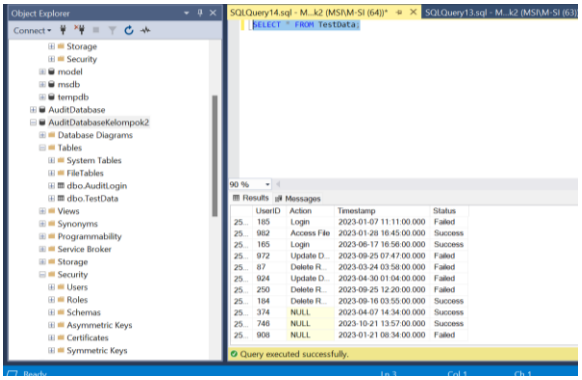


Gambar 5. Update

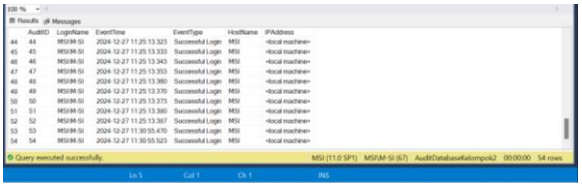
Gambar 5 menggambarkan aktivitas Update, yaitu proses mengubah atau memperbarui data yang sudah ada dalam sistem. Grafik ini kemungkinan menunjukkan seberapa sering aktivitas update berhasil dilakukan oleh pengguna yang memiliki otorisasi dibandingkan dengan upaya update yang tidak diizinkan. Angka aktivitas tidak sah pada update biasanya mencerminkan potensi risiko terhadap integritas data, seperti adanya percobaan untuk memodifikasi data oleh pengguna yang tidak memiliki akses yang benar.

Gambar 6 berkaitan dengan aktivitas Delete, yaitu penghapusan data dari sistem. Operasi delete biasanya lebih jarang dibandingkan create atau read karena sifatnya yang destruktif dan memengaruhi integritas data secara langsung. Grafik atau statistik pada gambar ini kemungkinan menunjukkan jumlah aktivitas penghapusan yang berhasil (sah) versus yang

gagal (tidak sah). Angka penghapusan tidak sah harus diawasi ketat, karena dapat mengindikasikan upaya menghapus data penting secara tidak sah atau tanpa izin yang benar.

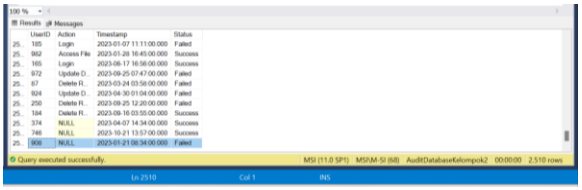


Gambar 6. Delete



Gambar 7. Cek Log Login

Penerapan log login pada Gambar 7 ini memastikan bahwa semua aktivitas login, baik yang sah maupun mencurigakan, tercatat dengan detail. Informasi yang dikumpulkan dari tabel AuditLogin dapat digunakan untuk analisis lebih lanjut, seperti mendeteksi pola login yang tidak biasa, menganalisis aktivitas login dari perangkat tertentu, atau mengidentifikasi potensi ancaman keamanan.

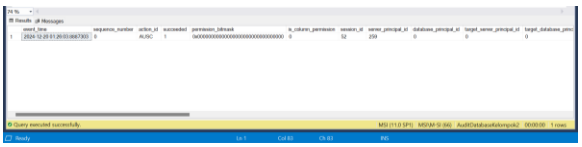


Gambar 8. Cek Data TestData

4.6. Verifikasi dan Analisis Log

Selanjutnya proses verifikasi dan analisis log dalam sistem audit dimulai dengan membaca data log yang telah dicatat oleh server. Langkah ini direpresentasikan dengan rumus:

$$Log_{Verifikasi} = \text{"}\{SELECT * FROM\ sys.fn\_get\_audit\_file('C:\AuditLogs *.sqlaudit', DEFAULT, DEFAULT)\} \quad (13)$$



Gambar 9. Analisis Log Audit

Rumus (13) menunjukkan bahwa data log yang tersimpan dalam file dengan ekstensi `.sqlaudit` di direktori `C:\AuditLogs\` diakses menggunakan fungsi bawaan SQL Server `sys.fn_get_audit_file`. Query ini membaca seluruh log yang telah dikumpulkan oleh server, memungkinkan administrator untuk memverifikasi detail aktivitas yang terekam.

Setelah log diverifikasi, langkah berikutnya adalah menganalisis data tersebut. Proses analisis direpresentasikan dengan rumus:

$$Log_{Analisis} = Export(Log_{Verifikasi})\ ke\ Power\ BI\ atau\ Excel \quad (14)$$

Rumus (14) menunjukkan bahwa data log yang telah diverifikasi diekspor ke alat analitik seperti Power BI atau Microsoft Excel. Ekspor ini bertujuan untuk mempermudah analisis lebih lanjut, seperti membuat visualisasi pola aktivitas, mendeteksi anomali, atau menyusun laporan audit yang lebih komprehensif.

Dalam keseluruhan proses ini: $Log_{Verifikasi}$ berfungsi untuk membaca dan memastikan bahwa data log telah tercatat dengan benar oleh sistem. $Log_{Analisis}$ menggunakan data log yang telah diverifikasi untuk menghasilkan wawasan mendalam melalui alat analitik.

4.7. Proses Implementasi SQL Server Audit

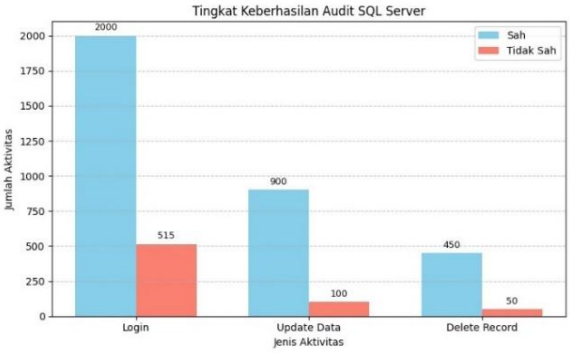
Tahap akhir yaitu proses implementasi SQL Server Audit secara keseluruhan direpresentasikan dalam rumus:

$$S_{Audit} = \{DB_{Audit}, T_{TestData}, T_{AuditLogin}, D_{TestData}, Audit_{Server}, L_{AuditLogin}, Log_{Verifikasi}, Log_{analisis}\} \quad (15)$$

Rumus (15) menggambarkan langkah-langkah utama dalam sistem audit yang mencakup elemen-elemen: DB_{Audit} yaitu basis data yang dibuat untuk menyimpan seluruh informasi terkait audit. Ini adalah fondasi utama dari sistem audit yang dirancang. $T_{TestData}$ yaitu tabel yang digunakan untuk menyimpan data simulasi aktivitas pengguna, mencakup informasi seperti UserID, jenis aktivitas (Action), waktu kejadian (Timestamp), dan status aktivitas (Status). $T_{AuditLogin}$ yaitu tabel yang dirancang khusus untuk mencatat log aktivitas login pengguna. Data ini meliputi nama pengguna, waktu login, jenis aktivitas login, perangkat yang digunakan, dan alamat IP. $D_{TestData}$ yaitu data simulasi yang dimasukkan ke dalam tabel $T_{TestData}$. Data ini mencakup elemen-elemen acak yang mensimulasikan aktivitas pengguna, seperti login, pembaruan data, dan penghapusan data. $Audit_{Server}$ yaitu objek audit yang dikonfigurasi untuk mencatat semua aktivitas ke dalam file log di server. Objek ini diaktifkan untuk memulai proses pencatatan data audit. $L_{AuditLogin}$ yaitu log login yang dihasilkan secara otomatis oleh trigger. Log

ini mencatat semua aktivitas login, termasuk informasi pengguna dan waktu kejadian. $Log_{Verifikasi}$ yaitu proses membaca data log yang telah dicatat oleh server menggunakan query khusus. Langkah ini memastikan bahwa data audit telah tercatat dengan benar. $Log_{analisis}$ yaitu proses ekspor dan analisis data log menggunakan alat seperti Power BI atau Excel. Analisis ini memungkinkan administrator untuk mendeteksi pola aktivitas, anomali, atau potensi ancaman keamanan.

Pada Gambar 10 menunjukkan tingkat keberhasilan audit aktivitas pada SQL Server berdasarkan jenis aktivitas, dengan dua kategori hasil audit, yaitu "Sah" dan "Tidak Sah".



Gambar 10. Tingkat Keberhasilan Audit SQL Server

Grafik tersebut menggambarkan tingkat keberhasilan audit pada SQL Server berdasarkan tiga jenis aktivitas, yaitu Login, Update Data, dan Delete Record, dengan dua kategori hasil audit: Sah (aktivitas yang valid) dan Tidak Sah (aktivitas yang tidak valid). Pada aktivitas Login, terdapat 2000 aktivitas sah dan 515 aktivitas tidak sah, yang menunjukkan bahwa mayoritas login berhasil dilakukan sesuai prosedur, namun terdapat jumlah percobaan login tidak sah yang cukup signifikan. Aktivitas ini berpotensi mencerminkan upaya akses ilegal atau kesalahan pengguna yang harus diperhatikan lebih lanjut.

Untuk aktivitas Update Data, tercatat 900 aktivitas sah dan 100 aktivitas tidak sah. Meskipun aktivitas sah mendominasi, angka tidak sah tetap memerlukan pengawasan karena dapat mengindikasikan pelanggaran atau perubahan data yang tidak diotorisasi. Sementara itu, aktivitas Delete Record memiliki 450 aktivitas sah dan 50 aktivitas tidak sah, yang menunjukkan frekuensi penghapusan data relatif rendah dibandingkan aktivitas lainnya, namun tetap perlu pengawasan ketat mengingat potensi risiko kehilangan data penting akibat aktivitas tidak sah.

4.8. Evaluasi dan Pengujian

Untuk menilai dan menguji keamanan basis data menggunakan SQL Server Audit dalam mendeteksi aktivitas tidak sah, beberapa langkah penting dapat dilakukan secara terstruktur. Tahap pertama adalah mengaktifkan dan mengonfigurasi SQL Server Audit, termasuk pengaturan log untuk merekam aktivitas-aktivitas utama seperti login, perubahan data, atau penghapusan data. Proses ini juga melibatkan penentuan lokasi penyimpanan log agar data audit terlindungi dan mudah diakses saat diperlukan. Selanjutnya, dilakukan pengujian pencatatan aktivitas untuk memastikan bahwa setiap tindakan, baik yang sah maupun tidak sah, tercatat secara lengkap dengan informasi mendetail seperti waktu, pengguna, dan jenis aktivitas.

Langkah berikutnya mencakup analisis data log audit, yang bertujuan untuk mengidentifikasi pola atau aktivitas yang tidak sesuai atau mencurigakan. Analisis ini dilakukan dengan membandingkan log yang tercatat dengan pola aktivitas normal (baseline), sehingga sistem lebih mudah mendeteksi anomali atau indikasi pelanggaran. Untuk menguji efektivitas deteksi, dilakukan simulasi aktivitas tidak sah, seperti mencoba login dengan kata sandi yang salah atau mengakses data tanpa otorisasi. Langkah ini memastikan bahwa sistem dapat mendeteksi dan mencatat aktivitas yang tidak sesuai.

Validasi kepatuhan terhadap kebijakan keamanan menjadi hal penting untuk dilakukan. Langkah ini memastikan bahwa konfigurasi SQL Server Audit sejalan dengan kebijakan internal organisasi atau standar keamanan global, seperti ISO 27001. Sebagai tambahan, diperlukan pengujian keandalan dan ketahanan sistem, dengan cara mengamati performa SQL Server Audit dalam kondisi beban kerja tinggi atau menghadapi banyak permintaan akses secara bersamaan.

5. KESIMPULAN DAN SARAN

Kesimpulan berdasarkan penelitian ini, SQL Server Audit terbukti efektif dalam mencatat dan memonitor aktivitas basis data, termasuk mendeteksi aktivitas yang sah dan tidak sah. Dengan konfigurasi yang tepat, fitur ini mampu memberikan gambaran lengkap tentang berbagai jenis tindakan, seperti login, perubahan data, atau penghapusan data. Hasil implementasi menunjukkan bahwa SQL Server Audit dapat diandalkan untuk mendeteksi aktivitas mencurigakan, seperti upaya login tidak sah atau modifikasi data tanpa izin. Log yang dihasilkan memberikan informasi detail yang mendukung analisis lebih lanjut untuk mengidentifikasi potensi ancaman.

Proses analisis log memainkan peran kunci dalam mendeteksi pola anomali atau pelanggaran. Dengan menggunakan alat tambahan seperti Power BI atau Excel, data log dapat divisualisasikan untuk mempercepat pengambilan keputusan dalam mitigasi risiko keamanan. Penggunaan data simulasi memberikan lingkungan pengujian yang realistis, memungkinkan evaluasi performa audit dalam berbagai skenario, termasuk beban kerja tinggi dan serangan cyber. Ini membuktikan keandalan SQL Server Audit sebagai alat pendukung keamanan basis data. SQL Server Audit dapat membantu organisasi memenuhi standar dan regulasi keamanan data seperti ISO 27001, dengan menyediakan mekanisme

pencatatan yang terstruktur dan komprehensif. Ini menjadikannya elemen penting dalam strategi keamanan data yang lebih luas. Saran untuk penelitian selanjutnya, dapat difokuskan pada integrasi kecerdasan buatan untuk mendeteksi pola mencurigakan di log audit secara otomatis, serta membandingkan SQL Server Audit dengan teknologi audit lainnya.

Penelitian juga dapat memperluas cakupan ke lingkungan *cloud*, mengkaji kinerja SQL Server Audit di bawah beban kerja tinggi, dan mengevaluasi integrasinya dengan standar keamanan seperti NIST atau *Zero Trust Architecture*. Selain itu, pendekatan inovatif seperti penggunaan *blockchain* untuk menjamin integritas log audit serta simulasi serangan *real-time* dapat menjadi kontribusi signifikan dalam meningkatkan keamanan basis data. Studi lanjutan tentang efisiensi biaya implementasi dan pengamanan database IoT juga penting untuk menjawab tantangan teknologi masa depan.

DAFTAR PUSTAKA

[1] M. I. P. N. Nur Indah Septiyani Sirait, "PENTINGNYA SISTEM INFORMASI AUDIT DALAM MENINGKATKAN KEAMANAN," *Kohesi: Jurnal Multidisiplin Saintek*, vol. Volume 5 No 4, 2024, Accessed: Jan. 14, 2025. [Online]. Available: https://ejournal.warunayama.org/kohesi

[2] A. M. Ujung, M. Irwan, and P. Nasution, "Pentingnya Sistem Keamanan Database untuk melindungi data pribadi," *JISKA: Jurnal Sistem Informasi Dan Informatika*, vol. 1, no. 2, p. 44, 2023, [Online]. Available: http://jurnal.unidha.ac.id/index.php/jteksis

[3] I. Joshua and D. W. Chandra, "ANALISIS LOGGING DAN AUDITING AKSES DATABASE SQL MELALUI KONEKSI REMOTE AKSES MENGGUNAKAN ANYDESK," 2023.

[4] L. Yang, *Teaching Database Security and Auditing*, vol. 41. 2009.

[5] A. Susanto, L. Ija, and L. O. Bakrim, "Audit Sistem Informasi Akademik Berdasarkan Aktifitas Pengguna," *Jurnal Sistem Informasi dan Sistem KOmputer*, vol. 5, no. 1, Jan. 2020, Accessed: Feb. 02, 2025. [Online]. Available: https://doi.org/10.51717/simkom.v5i1.40

[6] I. W. G. Arisna, I. M. Sukarsa, and P. W. Buana, "Model Sinkronisasi Database Satu Arah dengan Metode Audit Log Menggunakan Apache Kafka," *Jurnal Ilmiah Teknologi dan Komputer*, vol. 4, no. 2, Aug. 2023, Accessed: Feb. 02, 2025. [Online]. Available: https://ojs.unud.ac.id/index.php/jitter/article/download/104498/50599

[7] M. Budhiningtias Winanti and I. Dzulhan, "AUDIT KEAMANAN SISTEM INFORMASI AKADEMIK DENGAN KERANGKA KERJA ISO 27001 DI PROGRAM STUDI SISTEM INFORMASI UNIKOM."

[8] H. Yahya, M. Aznur, N. Agnestesia, and A. Ikhwan, "Analisis Keamanan Fisik Data Prodi Sistem Informasi UIN Sumatera Utara Medan Menggunakan Standar ISO 27001," *Jurnal Penelitian Dan Pengkajian Ilmiah Eksakta*, vol. 2, no. 1, pp. 39–44, Jan. 2023, doi: 10.47233/jppie.v2i1.675.

[9] A. Afifah Rodhiyatun Nisa, Ananditto Daffa Wijayanto, Arya Prabudi Jaya Priana, and A. Setiawan, "Analisis Log Server untuk mendeteksi Serang DDoS pada Keamaan Jaringan di Website," *Journal of Internet and Software Engineering*, vol. 1, no. 3, p. 17, Jun. 2024, doi: 10.47134/pjise.v1i3.2612.

[10] H. Madiistriyatno and I. Santoso, *METODOLOGI PENELITIAN KUANTITATIF*. 2021.

[11] L. Max, S. Florian, H. Georg, and W. Markus, "Cloud-based User Entity Behavior Analytics Log Data Set [Data Set]," *Jurnal Zenodo*, 2022, Accessed: Jan. 14, 2025. [Online]. Available: https://doi.org/10.5281/zenodo.7119953

[12] B. Bašić, P. Udovičić, and O. Orel, "In-database Auditing Subsystem for Security Enhancement," *Jurnal IEEE*, 2021, doi: 10.23919/MIPRO52101.2021.9596906.

[13] E. Sutanty, Maukar, K. D. Astuti, and Handayani, "Penerapan Model Arsitektur VGG16 Untuk Klasifikasi Jenis Sampah," *Jurnal DECODE: Jurnal Pendidikan Teknologi Informasi*, vol. 3, Sep. 2023, Accessed: Jan. 14, 2025. [Online]. Available: https://doi.org/10.51454/decode.v3i2.331

[14] M. Padri and J. Rahmadian, "Perancangan Aplikasi Point of Sale Berbasis Website Pada PT Lottemart Indonesia," *Jurnal Maklumatika*, vol. 8, no. 1, pp. 80–89, 2021, Accessed: Jan. 14, 2025. [Online]. Available: https://maklumatika.i-tech.ac.id/index.php/maklumatika/article/view/121