Jurnal Ilmiah MEDIA SISFO
Vol. 19, No. 2, Oktober 2025

ISSN: 1978-8126
e-ISSN: 2527-7340

Algoritma *Transparent Data Encryption* : Analisis dan Evaluasi Kinerja pada Sistem Keamanan Data At Rest SQL Server

Bita Parga Zen[1], Ipam Fuaddina Adam[2], Riswan Azhari[3]*

Teknik Informatika, Fakultas Teknologi dan Desain, Universitas Ma Chung[1]
Teknik Informatika, Fakultas Informatika, Universitas Telkom[2,3]

Villa Puncak Tidar N-01, 65151, Malang, Jawa Timur, Indonesia[1]
Jl. D.I Panjaitan No. 128 Purwokerto, Jawa Tengah, Indonesia[2,3]
bita.parga@machung.ac.id[1], ipamya@telkomuniversity.ac.id[2], azhary.riswan980@gmail.com[3]

Submitted : **08/05/2025**; Reviewed : **09/06/2025**; Accepted : **14/10/2025**; Published : **31/10/2025**

Abstract

This study evaluates the data-at-rest security enhancement and performance impact of Transparent Data Encryption (TDE) implementation on Microsoft SQL Server by comparing three conditions: no TDE, TDE AES-128, and TDE AES-256. Three databases with identical schemas were prepared. They were filled with a mix of text, numeric, image, and video data. The databases were then tested sequentially according to the methodology. The process included schema and integrity validation, SMK key hierarchy configuration, certificates, DE, encryption status verification, encryption and decryption throughput measurement, backup and recovery time evaluation, I/O stall analysis, and CPU and memory usage monitoring. Data-at-rest security validation was performed using file and log access tests, as well as cross-server recovery requiring keys. The results showed that TDE successfully eliminates file and log readability during direct access. It also requires the presence of keys during recovery, thus significantly reducing the risk of data exposure. Performance impacts are at acceptable levels: average CPU load increases around 1–2% and memory load increases by 1–8% under load. Encryption and decryption throughput between AES-128 and AES-256 are relatively equivalent. Backup operations tend to be slightly slower on TDE, while restores are faster, in line with lower read I/O stalls on the encrypted basis. Overall, TDE provides significant security enhancements with minimal performance compromise. Limitations of this study lie in the test load profile and hardware.

Keywords : database encryption, data security, application performance, sql server, transparent data encryption

Abstrak

Penelitian ini mengevaluasi peningkatan keamanan data-at-rest dan dampak kinerja penerapan *Transparent Data Encryption* (TDE) pada Microsoft SQL Server melalui perbandingan tiga kondisi: tanpa TDE, TDE AES-128, dan TDE AES-256. Tiga basis data dengan skema identik disiapkan, diisi campuran data berupa teks, numerik, gambar, video, lalu diuji berurutan sesuai metodologi melalui validasi skema dan integritas, konfigurasi hierarki kunci SMK, sertifikat, DE, verifikasi status enkripsi, pengukuran throughput enkripsi dan dekripsi, evaluasi waktu cadangan dan pemulihan, analisis I/O stall, serta pemantauan pemakaian CPU dan memori. Validasi keamanan data-at-rest dilakukan dengan uji akses berkas dan log, serta pemulihan lintas server yang membutuhkan kunci. Hasil menunjukkan TDE berhasil menghilangkan keterbacaan berkas dan log ketika diakses langsung serta menuntut keberadaan kunci saat pemulihan, sehingga risiko paparan data berkurang signifikan. Dampak kinerja berada pada tingkat yang dapat diterima: kenaikan CPU rata-rata sekitar 1–2% dan memori 1–8% saat beban, throughput enkripsi dan dekripsi antara AES-128 dan AES-256 relatif setara, operasi cadangan cenderung sedikit lebih lambat pada TDE, sedangkan pemulihan lebih cepat, sejalan dengan I/O stall baca yang lebih rendah pada basis terenkripsi. Secara keseluruhan, TDE memberikan penguatan keamanan yang bermakna dengan kompromi kinerja minimal. Keterbatasan penelitian ini terletak pada profil beban uji dan perangkat keras

Kata kunci : enkripsi database, keamanan data, kinerja aplikasi, sql server, transparent data encryption .

1. Pendahuluan

Penggunaan teknologi di semua aspek ini menghasilkan banyak data yang perlu dikelola. Jika tidak dikelola dengan baik, data tersebut dapat menimbulkan masalah besar karena berisiko mengalami kebocoran [1] [2], [3]. Kebocoran data adalah masalah serius yang mengkhawatirkan individu, perusahaan, dan pemerintah di seluruh dunia [4]. Kebocoran ini terjadi ketika data yang seharusnya bersifat rahasia atau pribadi, seperti informasi pribadi, bisnis, atau pemerintah, terungkap kepada pihak yang tidak berwenang [5]. Dampak dari

ISSN: 1978-8126
e-ISSN: 2527-7340

kebocoran data termasuk hilangnya privasi individu, penyalahgunaan data, dan berbagai konsekuensi serius lainnya[6] [7]. Keamanan data sangat penting karena setiap individu memiliki hak atas perlindungan diri pribadi, keluarga, kehormatan, martabat, dan aset yang mereka miliki, serta hak untuk merasa aman dan terlindungi dari ancaman yang bisa menghalangi mereka melakukan atau tidak melakukan sesuatu yang menjadi hak asasi mereka[6], [8], [9], [10]

Kebocoran data merupakan masalah serius yang mengkhawatirkan, dan banyak faktor yang dapat menyebabkannya, termasuk lemahnya system keamanan data yang digunakan dan serangan dari pihak yang tidak berwenang, seperti para hacker. Dalam dunia siber, serangan SQL Injection merupakan salah satu bentuk serangan yang sangat berdampak pada kebocoran data. Di Indonesia, BSSN (Badan Siber dan Sandi Negara) mencatat bahwa selama tahun 2022 Insiden kebocoran data memasuki top 3 insiden siber dengan jumlah total 399 dugaan [11].

Kebocoran data telah menjadi masalah yang mendesak dalam dunia digital yang terus berkembang. Mencegah, mendeteksi, dan menangani pelanggaran data menjadi prioritas utama bagi organisasi dan pemerintah. Untuk mengatasi resiko kebocoran data, organisasi perlu menerapkan strategi keamanan data yang komprehensif [12]. Salah satu pendekatan yang efektif adalah melalui enkripsi data, khususnya dalam database. Dengan menerapkan enkripsi pada database, data sensitif di dalamnya dapat dilindungi dengan baik, bahkan jika peretas berhasil memasukkan perintah SQL berbahaya [13]. Enkripsi memastikan bahwa data yang tersimpan hanya dapat diakses dengan kunci dekripsi yang tepat, menjaga kerahasiaan dan integritas informasi. Enkripsi adalah proses untuk mengamankan pesan (plain text) yang diubah sedemikian rupa menjadi pesan tersembunyi (ciphertext). Selain itu, memiliki rencana tanggap insiden yang efisien menjadi penting, karena ini memungkinkan organisasi untuk segera menangani dan melaporkan kebocoran data jika terjadi, sehingga dapat mengurangi dampak negatif yang mungkin timbul [14], [15].

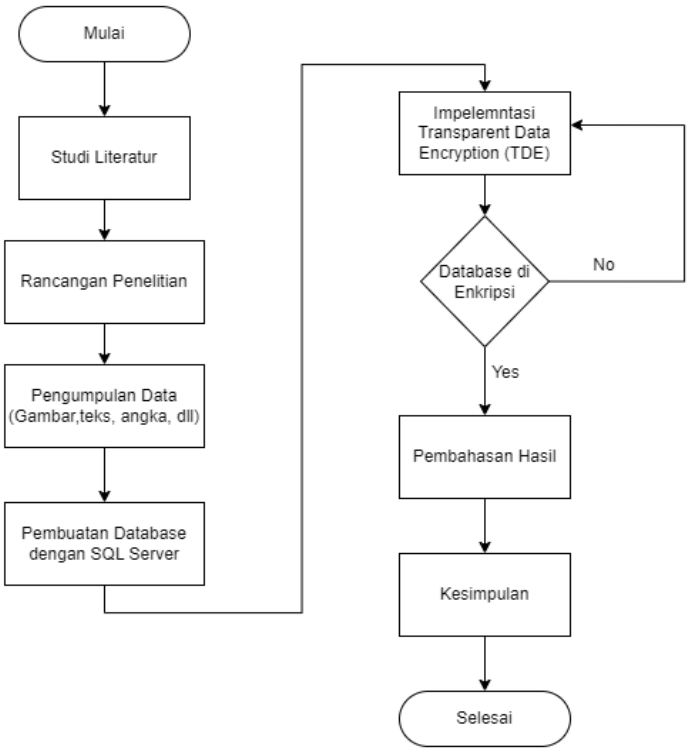
Berpijak pada Penelitian yang dilakukan oleh Sourav Mukherjee dengan judul "Popular SQL Server Database Encryption Choices" mengungkapkan fakta yang mengkhawatirkan: pelanggaran keamanan hampir pasti terjadi pada sebagian besar organisasi saat ini. Hal ini menunjukkan bahwa tantangan keamanan siber semakin kompleks dan organisasi perlu mengambil langkah-langkah proaktif untuk melindungi data mereka. Salah satu solusi yang semakin diakui adalah penggunaan enkripsi pada database, khususnya dalam konteks SQL Server [16] [17][18]

Meskipun sudah banyak para ahli yang membahas penelitian seputar Enkripsi database dan tentunya penelitian ini memiliki kesamaan dengan penelitian terdahulu seperti , Teknik enkripsi, metode enkripsi, dan juga database [19], [20], [21]. Namun penulis akan menegaskan sisi perbedaan penelitiaan ini dengan penelitian sebelumnya. Pertama, perbedaan dari Teknik enkripsi yang menggunakan *Transparent Data Encryption* . Keamanan data dalam kolom dan tabel pada database merupakan tujuan utama dari system Transparent Data Encryption . Kedua, penelitian ini akan diimplementasikan pada database yang akan dikelola oleh Microsoft SQL Server.

ISSN: 1978-8126
e-ISSN: 2527-7340

2. Metodologi

Berikut ini adalah serangkaian tahapan atau proses berpikir dalam pembuatan sistem pakar mulai dari studi literatur hingga pengujian sistem yang penulis lakukan selama penelitian.



Gambar 1. Diagram Alir Penelitian

Berikut adalah penjelasan dari tahapan diagram alir penelitian pada gambar 1.

1. Studi Literatur : Pada penelitian ini berisi tentang pencarian dan pemahaman mengenai literatur yang berkaitan dengan permasalahan, yaitu Penerapan metode enkripsi database Transparent pada database SQL Server untuk meningkatkan kemanan data. Sumber literatur didapatkan dari berbagai jurnal, peper, dan buku yang mendukung pemahaman dalam menyelesaikan penelitian ini.
2. Rancangan Penelitian : Tujuan utama penelitian adalah meningkatkan tingkat keamanan data pada database SQL Server melalui penerapan metode enkripsi database transparan. Fokus utama penelitian adalah mengatasi risiko potensial kebocoran data yang dapat terjadi dalam sistem manajemen database. Permasalahan yang ingin dipecahkan adalah menciptakan lapisan keamanan tambahan yang efektif sebagai perlindunan integritas dan kerahasiaan data dalam lingkungan SQL Server.
3. Pengumpulan Data : Pengumpulan data dalam penelitian ini mencakup berbagai bentuk informasi, termasuk gambar, teks, dan angka, yang akan menjadi dasar untuk pembuatan database. Proses ini bertujuan untuk mengumpulkan elemen-elemen yang diperlukan guna menyusun struktur data yang komprehensif dalam database. Data yang diperoleh akan menjadi landasan utama untuk analisis, interpretasi, dan implementasi metode enkripsi database transparan pada SQL Server, yang menjadi fokus penelitian dalam upaya meningkatkan keamanan data.
4. Pembuatan Database dengan SQL Server : Penelitian ini dimulai dengan identifikasi kebutuhan yang mencakup analisis data yang akan dienkripsi, tingkat keamanan yang diperlukan, dan performa sistem yang diinginkan. Selanjutnya, desain skema database disusun berdasarkan kebutuhan tersebut, mencakup tabel, relasi, tipe data, dan atribut lain yang diperlukan. Database baru kemudian dibuat menggunakan SQL Server dengan parameter konfigurasi yang sesuai. Untuk uji coba penerapan metode enkripsi data transparan, tiga database digunakan: WTDE (tanpa enkripsi transparan) sebagai kontrol untuk membandingkan performa dan keamanan; TDE128

ISSN: 1978-8126
e-ISSN: 2527-7340

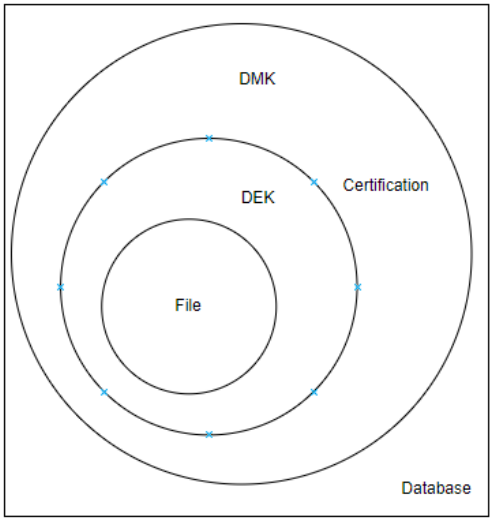
yang menggunakan enkripsi AES128 dengan kunci 128 bit; dan TDE256 yang menggunakan enkripsi AES256 dengan kunci 256 bit.

5. Implementasi Transparent Data Encryption : Penerapan *Transparent Data Encryption* pada *SQL Server* merupakan suatu langkah kunci yang strategis dalam menjaga keamanan data dengan mengenkripsi informasi yang disimpan di dalam *database*. Dalam rangka implementasi teknologi ini, beberapa tahapan konfigurasi di *SQL Server* perlu dilakukan seperti pembuatan *master key, sertifikat, dan database encryption key*, Pembuatan *Master Key: Master key* adalah kunci enkripsi yang disimpan dalam *database master*, Pembuatan Sertifikat: Sertifikat diperlukan untuk mengenkripsi *database encryption key*, Pembuatan *Database Encryption Key*: dan Mengaktifkan *Enkripsi* pada *Database*

3. Hasil dan Pembahasan

3.1. Pembuatan Database

Pengujian yang dilakukan didasarkan pada jenis data yang akan di enkripsi pada teknologi *Transparent Data Encryption* yang dapat diterapkan pada Microsoft *SQL Server* setelah *database* terbentuk. *Transparent Data Encryption* bertujuan untuk meningkatkan keamanan data dengan melakukan enkripsi menyeluruh terhadap informasi yang disimpan. Proses implementasi *Transparent Data Encryption* melibatkan beberapa lapisan enkripsi, termasuk *Service Maste Key* (*SMK)* dan *Database Master Key* (*DMK)*, serta menggunakan certificate protected untuk melindungi *Database Encryption Key* (*DEK)*. Aktivasi *Transparent Data Encryption* dapat dilakukan dengan memberikan syntax yang sesuai, sehingga seluruh proses enkripsi akan berjalan. Sebaliknya, jika tidak diaktifkan, data akan disimpan dalam *database* tanpa enkripsi. Implementasi *Transparent Data Encryption* pada *database* ini menjadi langkah krusial dalam menjaga keamanan data yang tersimpan.



Gambar 2. Lapisan Enkripsi

Bagian ini mencakup penjelesan mengenai pengujian serta analisis kinerja dari implementasi TDE (Transparent Data Encryption) dan membahas hasil dari implmenetasi TDE (Transparent Data Encryption) tersebut. Sesuai dengan diagram alir penilitian yang telah dibahas pada bab sebelumnya untuk proses perancangan database. Berikut adalah pembahasan mengenai perancangan sistem database yang menggunakan teknologi enkripsi TDE (Transparent Data Encryption):

Tabel 1. Pengujian Transparent Data Encryption

Database	**Jenis Data**			
DataWithAES256	Angka	Text	Gambar	Video
DatawithAES128	Angka	Text	Gambar	Video
DatanonTDE	Angka	Text	Gambar	Video

ISSN: 1978-8126
e-ISSN: 2527-7340

Penelitian ini melibatkan pembuatan tiga database SQL Server untuk menguji perbandingan antara penggunaan dan non-penggunaan teknologi Transparent Data Encryption . Perintah untuk membuat database adalah sebagai berikut:

```
CREATE DATABASE TDE256;
CREATE DATABASE TDE128;
CREATE DATABASE WTDE;
```

Setiap database memiliki tiga tabel utama: dbo.users, dbo.filedata, dan dbo.somedata dengan struktur sebagai berikut:

Tabel 2. Struktur Tabel Dbo.Filedata

Nama Kolom	**Tipe Data**	**Keterangan**
id	INT	Auto increment, *Primary key*
nama_file	NVARCHAR(MAX)	
tipe_file	NVARCHAR(MAX)	
ukuran_file	INT	
file_blob	NVARCHAR(MAX)	
User_id	INT	Forgen key ke ‘dbo.users(id)’

Tabel 3. Struktur Tabel dbo.users

Nama Kolom	**Tipe Data**	**Keterangan**
id	INT	Auto increment, *Primary key*
username	VARCHAR(50)	
password	VARCHAR(255)	
email	VARCHAR(255)	

Tabel 4. Struktur Tabel dbo.somedata

Nama Kolom	**Tipe Data**	**Keterangan**
id	INT	Auto increment
SomeText	VARCHAR(255)	

Pembuatan tabel-tabel dilakukan dengan mendefinisikan kolom, tipe data, dan indeks melalui antarmuka grafis atau menggunakan perintah *SQL*. Penetapan kunci utama dan kunci asing menjadi esensial untuk memastikan integritas referensial antar tabel.

ISSN: 1978-8126
e-ISSN: 2527-7340

```
USE namadatabase
CREATE TABLE dbo.filedata ( id INT IDENTITY(1,1) PRIMARY KEY,
   nama_file NVARCHAR(MAX),
   tipe_file NVARCHAR(MAX),
   ukuran_file INT,
   file_blob VARBINARY(MAX),
   user_id INT,
   FOREIGN KEY (user_id) REFERENCES dbo.users(id)
);
CREATE TABLE dbo.users ( id INT IDENTITY(1,1) PRIMARY KEY,
   username VARCHAR(50) NOT NULL UNIQUE,
   password VARCHAR(255) NOT NULL,
   email VARCHAR(255) NOT NULL
);
CREATE TABLE dbo.SomeData(Id INT IDENTITY(1,1), SomeText VARCHAR(255));
```

Tabel-tabel ini akan diisi dengan data yang telah ditentukan untuk menjalankan pengujian. Adapun tabel-tabel yang diuji adalah sebagai berikut:

Tabel 5. Tabel Pengujian

No	Nama Tabel	Pengujian
1	dbo.users	Integritas Data
2	dbo.filedata	Integritas Data Enkripsi dan Dekripsi gambar serta video
3	dbo.somedata	Integritas Data Enkripsi dan Dekripsi text serta angka

Tabel di atas merinci pengujian yang dilakukan pada tiga tabel utama dalam database. Tabel `dbo.users` diuji untuk integritas data, memastikan konsistensi dan validitas skema. Tabel `dbo.filedata` diuji untuk integritas data serta performa enkripsi dan dekripsi pada gambar dan video, menilai efisiensi algoritma enkripsi. Tabel `dbo.somedata` diuji untuk integritas data serta performa enkripsi dan dekripsi pada teks dan angka, mengevaluasi kinerja algoritma enkripsi terhadap data yang lebih sederhana

3.2. Implementasi *Transparent Data Encryption*

Implementasi *Transparent Data Encryption* pada *SQL Server* dilakukan melalui beberapa tahapan penting untuk memastikan keamanan data yang disimpan di dalam *database*. Berikut adalah langkah-langkah yang dilakukan:

1. Membuat *Master Key*
   Tahap pertama adalah pembuatan *Master Key* di *database Master*. *Master Key* ini diperlukan sebagai dasar untuk enkripsi lebih lanjut. Perintah *SQL* untuk membuat *Master Key* adalah sebagai berikut:

```
USE Master;
CREATE MASTER KEY ENCRYPTION
BY PASSWORD = '1nD0N[-5i4';
```

2. Membuat Sertifikat
   Langkah kedua melibatkan pembuatan sertifikat yang dilindungi oleh *Master Key*. Sertifikat ini akan digunakan untuk menghubungkan *database* dengan kunci enkripsi. Perintah *SQL* untuk membuat sertifikat adalah:

```
CREATE CERTIFICATE PWT
WITH SUBJECT = 'Purwokerto';
```

ISSN: 1978-8126
e-ISSN: 2527-7340

3. Membuat *Database Encryption Key*
   Tahap ketiga mencakup pembuatan *Database Encryption Key (DEK)* untuk *database* yang bersangkutan. Ini melibatkan asosiasi antara sertifikat yang telah dibuat dengan *database* dan penentuan algoritma enkripsi yang akan digunakan. Perintah *SQL* untuk langkah ini adalah:

```
USE namadatabase;  -- Ganti dengan nama database yang ingin dienkripsi
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256  -- Algoritma yang digunakan
ENCRYPTION BY SERVER CERTIFICATE PWT;
```

4. Mengaktifkan Enkripsi
   Setelah *Database Encryption Key* dibuat, langkah keempat adalah mengaktifkan enkripsi pada *database* dengan menggunakan perintah *SQL* berikut:

```
ALTER DATABASE namadatabase  -- Ganti dengan nama database yang ingin dienkripsi
SET ENCRYPTION ON;
```

5. Pengecekan status Enkripsi
   Tahap terakhir adalah melakukan pengecekan untuk memastikan bahwa data di dalam *database* telah terenkripsi dengan baik. Perintah *SQL* untuk pengecekan status enkripsi adalah:

```
SELECT Encryption_state, percent_complete, encryptor_type, *
FROM sys.dm_database_encryption_keys;
```

Dengan mengikuti langkah-langkah di atas, implementasi *Transparent Data Encryption* pada *SQL Server* dapat dilakukan secara efektif. Langkah-langkah ini mencakup pembuatan *Master Key*, pembuatan sertifikat, pembuatan *Database Encryption Key*, aktivasi enkripsi, dan pengecekan status enkripsi. Implementasi ini penting untuk meningkatkan keamanan data yang tersimpan di dalam *database*, memastikan bahwa data yang ada terlindungi dari akses yang tidak sah dan potensi ancaman keamanan lainnya.

3.3. Pengujian Kinerja Database

Tabel 6. Tabel Pengujian

Database	**Format Data**	**Pengujian**		
		Validasi Skema	Konsistensi	Retrieval Testing
WTDE	Angka	✓	✓	✓
	Teks	✓	✓	✓
	Gambar	✓	✓	✓
	Video	✓	✓	✓
TDE128	Angka	✓	✓	✓
	Teks	✓	✓	✓
	Gambar	✓	✓	✓
	Video	✓	✓	✓
TDE256	Angka	✓	✓	✓
	Teks	✓	✓	✓
	Gambar	✓	✓	✓
	Video	✓	✓	✓

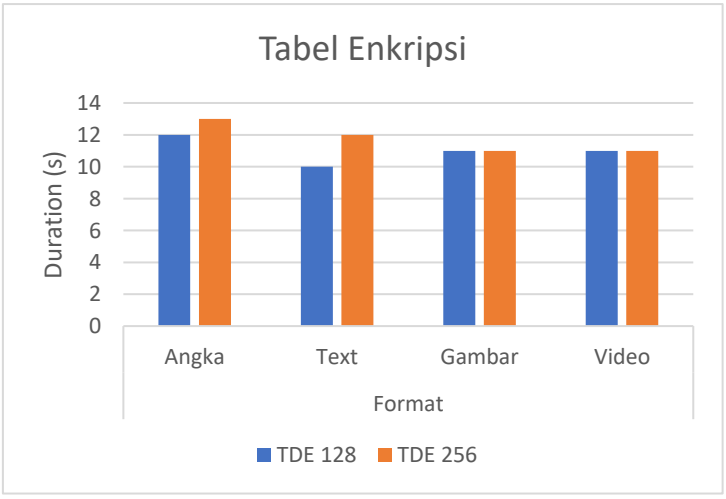
Berdasarkan tabel hasil pengujian menunjukkan bahwa semua tabel dalam ketiga *database* memenuhi kriteria integritas data, dengan semua constraint (*primary key*, *foreign key*, *unique constraint*, dan *not null constraint*) diterapkan dengan benar dan tidak ada pelanggaran integritas data.

3.4. Enkripsi dan Dekripsi Data

Pengujian dilakukan untuk mengevaluasi kecepatan enkripsi dan dekripsi pada database TDE128 dan

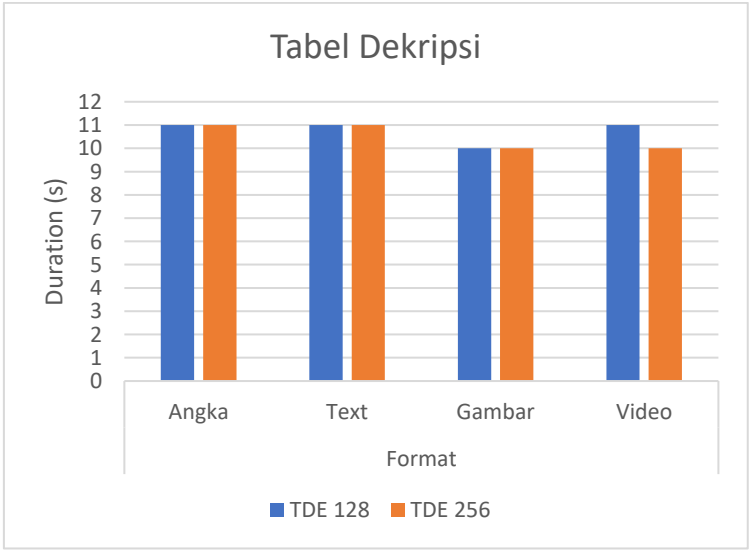
ISSN: 1978-8126
e-ISSN: 2527-7340

TDE256 dengan data sebesar 1 GB. Hasil pengujian dapat ditunjukkan dalam gambar berikut:



Gambar 3. Diagram Enkripsi

Selanjutnya dilakukan pengujian kecepatan dekripsi dengan hasil pengujian ditunjukkan dalam gambar berikut:



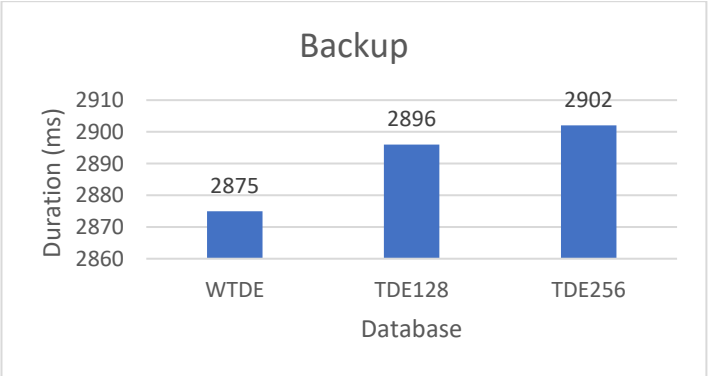
Gambar 4. Diagram Dekripsi

Hasil ini menunjukkan bahwa perbedaan waktu enkripsi dan dekripsi antara TDE128 dan TDE256 tidak signifikan, dengan keduanya dapat memproses data secara efisien.

3.5. Backup dan Restore Data

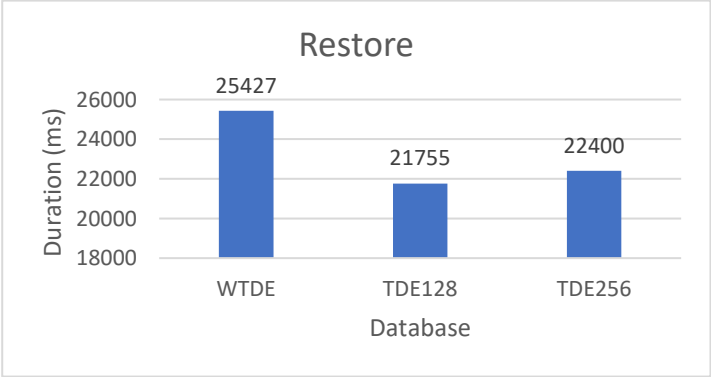
Pengujian dilakukan untuk mengevaluasi kecepatan backup dan restore pada database WTDE, TDE128, dan TDE256, masing-masing dengan ukuran data sekitar 1 GB. Hasil pengujian backup database ditunjukkan dalam gambar berikut:

ISSN: 1978-8126
e-ISSN: 2527-7340



Gambar 5. Diagram Backup

Selanjutnya dilakukan pengujian restore restore database dengan hasil pengujian ditunjukkan dalam gambar berikut:



Gambar 6. Diagram Restore

Hasil pengujian menunjukkan bahwa proses backup pada database terenkripsi (TDE128 dan TDE256) sedikit lebih lambat dibandingkan dengan database tanpa enkripsi (WTDE). Namun, proses restore pada database terenkripsi lebih cepat. Hal ini disebabkan oleh penundaan input dan output pada proses restore yang dapat dilihat pada tabel berikut:

Tabel 6. Analisis Input dan output

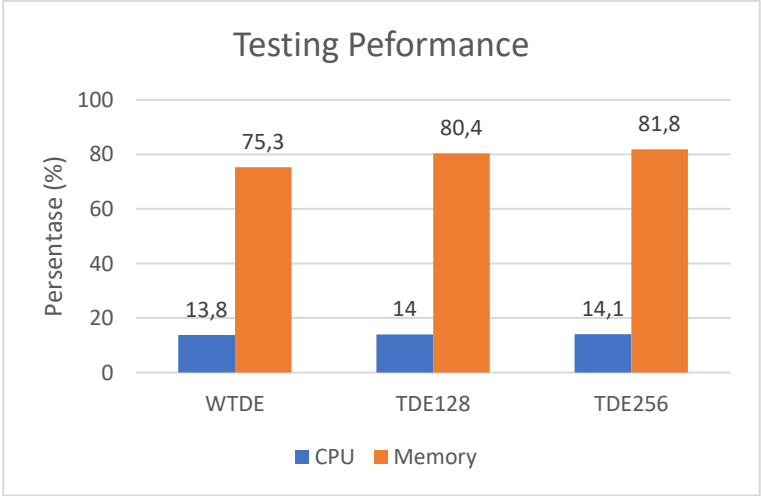
Database	***File_id***	***Num_of_reads***	***Num_of_writes***	***Io_stall_read_ms***	***Io_stall_write_ms***
WTDE	1	69	8	53	1
WTDE	2	289	19	50	2
TDE128	1	70	8	31	1
TDE128	2	246	19	45	2
TDE256	1	70	8	35	1
TDE256	2	246	20	50	2

Berdasarkan tabel, dapat disimpulkan bahwa database WTDE lebih lambat dalam proses restore dibandingkan dengan TDE128 dan TDE256 karena penundaan I/O (Io_stall_read_ms) yang lebih tinggi pada WTDE (53 ms dan 50 ms) dibandingkan dengan TDE128 (31 ms dan 45 ms) dan TDE256 (35 ms dan 50 ms). Meskipun jumlah operasi baca dan tulis (num_of_reads dan num_of_writes) relatif mirip di ketiga database, penundaan I/O yang lebih rendah pada TDE128 dan TDE256 menunjukkan bahwa waktu yang dihabiskan untuk operasi baca dan tulis lebih sedikit, sehingga proses restore lebih cepat pada database yang menggunakan Transparent Data Encryption.

ISSN: 1978-8126
e-ISSN: 2527-7340

3.6. CPU dan Memory

Pengujian dilakukan untuk mengevaluasi persentase penggunaan CPU dan memory pada masing-masing database dengan skenario pengujian beban bertahap. Hasil pengujian adalah sebagai berikut:



Gambar 7. Diagram CPU dan Memory

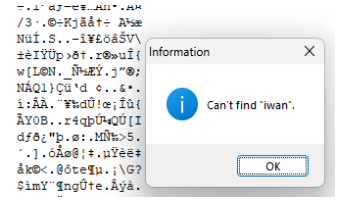
Hasil menunjukkan bahwa penggunaan CPU dan memory lebih tinggi pada database yang terenkripsi, terutama pada TDE256 karena kompleksitas algoritma dan jumlah operasi kriptografi yang lebih tinggi dibandingkan AES128.

3.7. Keamanan Data

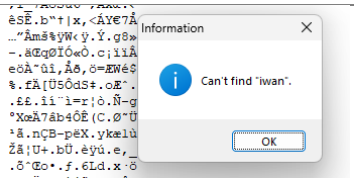
Dalam melakukan pengujian kemanan data penulis membuat dua scenario yitu:

1. Akses file data dan log database secara langsung (di luar SQL Server Management Studio atau alat administrasi database lainnya) untuk memastikan bahwa data tersebut tidak dapat dibaca tanpa proses dekripsi yang sesuai.
2. Backup database yang terenkripsi dan coba restore ke server lain untuk memastikan bahwa data tetap terenkripsi dan hanya bisa dibaca setelah di-decrypt dengan kunci yang tepat.

Tabel 7. Pengujian keamanan

Database	**Pengujian Keamanan *Database***	
	Akses File dan Log	**Restore**
WTDE	00 00 00 00 00 00 00 ..... 00 00 00 00 00 00 00 ..... 00 00 03 00 17 00 1E <...éÓ..... 61 6E 31 32 33 31 32 .l.iwaniwan12312 6D 70 6C 65 2E 63 6F 3iwan@example.co 04 00 00 03 00 17 00 m<...êÓ..... 77 61 6E 31 32 33 31 ..l.iwaniwan1231 61 6D 70 6C 65 2E 63 23iwan@example.c 00 04 00 00 03 00 17 om<...éÓ..... 69 77 61 6E 31 32 33 ...l.iwaniwan123 78 61 6D 70 6C 65 2E l23iwan@example. 04 00 04 00 00 03 00 com<...iÓ..... 6E 69 77 61 6E 31 32 ....l.iwaniwan12 65 78 61 6D 70 6C 65 3123iwan@example	Processed 138072 pages for database 'WTDE', file 'WTDE' on file 1. Processed 2 pages for database 'WTDE', file 'WTDE_log' on file 1. RESTORE DATABASE successfully processed 138074 pages in 35.743 seconds (30.179 MB/sec)
TDE128		Cannot find server certificate with thumbprint '0x01D8D6654E1F499DBC0EAE79C87EF3980709969E'. Msg 3013, Level 16, State 1, Line 7 RESTORE DATABASE is terminating abnormally.

ISSN: 1978-8126
e-ISSN: 2527-7340

TDE256		Cannot find server certificate with thumbprint '0x01D8D6654E1F499DBC0EAE79C87EF3980709969E'. Msg 3013, Level 16, State 1, Line 8 RESTORE DATABASE is terminating abnormally.

Berdasarkan tabel pengujian, menunjukkan bahwa database dengan Transparent Data Encryption lebih aman dibandingkan yang tanpa TDE. Pada skenario pertama, akses file dan log database untuk pembacaan data user iwan di tabel `dbo.users` tidak dapat dilakukan pada database dengan TDE, sementara database tanpa TDE dapat dibaca dan menemukan user iwan. Pada skenario kedua, database dengan TDE tidak bisa di-restore tanpa kunci yang tepat, sehingga data tetap aman, sedangkan database tanpa TDE bisa di-restore dengan mudah, memungkinkan pembacaan data yang ada.Dari dua scenario yang diatas dilakukan pengujian yang menghasilkan data yang berada di database yang menggunakan enkripsi TDE (Transparent Data Encryption) lebih aman dibandingkan database yang tidak menggunakan teknologi enkripsi TDE (Transparent Data Encryption). Akses file dan log database untuk pembacaan data tidak bisa dilakukan pada database yang menggunakan enkripsi TDE (Transparent Data Encryption), sedangkan yang tidak menggunakan TDE (Transparent Data Encryption) bisa dibaca. Untuk Scenario yang kedua Database dengan TDE (Transparent Data Encryption) tidak bisa melakukan restore tanpa kunci yang tepat sehingga data dalam database tidak bisa di baca sedangkan database tanpa TDE (Transparent Data Encryption) dapat di restore dengan mudah sehingga mempermudah dalam pembacaan data yang ada di database.

4. Kesimpulan

Hasil implementasi menunjukkan bahwa proses pembuatan Master key, sertifikat, dan Database Encryption Key (DEK) berjalan lancar, memastikan database berhasil dienkripsi dan terlindungi dari akses tidak sah. Transparent Data Encryption memberikan lapisan keamanan tambahan yang signifikan terhadap ancaman kebocoran data. Pengujian kinerja menunjukkan peningkatan waktu respons query dan penggunaan CPU setelah penerapan TDE. Meskipun ada peningkatan penggunaan sumber daya, perubahan ini masih dalam batas yang dapat diterima, mengingat tingkat keamanan yang diperoleh. Perbedaan kinerja antara database dengan dan tanpa TDE terlihat dalam proses backup dan restore, dengan database tanpa TDE lebih cepat dalam backup, sementara database dengan TDE lebih cepat dalam restore. Penggunaan CPU dan memori meningkat sekitar 1-2% untuk CPU dan 1-8% untuk memori.

Penelitian ini menunjukkan bahwa implementasi TDE pada SQL Server dapat meningkatkan keamanan data secara signifikan, melindungi data saat istirahat, dan memastikan keamanan data bahkan jika terjadi akses tidak sah. Penerapan TDE tidak mengurangi integritas data dan menjaga kerahasiaan informasi penting dalam database. Meskipun terdapat kompromi antara keamanan dan performa, penurunan kinerja masih dalam batas yang dapat diterima dan tidak mengganggu operasi database secara signifikan. Hasil ini menyoroti efisiensi TDE dalam mengamankan data dengan overhead yang minimal, menjadikannya solusi efektif untuk meningkatkan keamanan data dalam lingkungan database

Daftar Pustaka

[1] Y. Li and Q. Liu, “A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments,” *Energy Reports*, vol. 7, pp. 8176–8186, Nov. 2021, doi: 10.1016/j.egyr.2021.08.126.

[2] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, “Towards insighting cybersecurity for healthcare domains: A comprehensive review of recent practices and trends,” *Cyber Security and Applications*, vol. 1, p. 100016, Dec. 2023, doi: 10.1016/j.csa.2023.100016.

[3] M. Fitriana, K. A. AR, and J. M. Marsya, “Penerapan Metode National Institute of Standars and Technology (Nist) Dalam Analisis Forensik Digital Untuk Penanganan Cyber Crime,” *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, vol. 4, no. 1, p. 29, 2020, doi: 10.22373/cj.v4i1.7241.

[4] B. P. Zen, R. A. G. Gultom, and A. H. S. Reksoprodjo, “Analisis Security Assessment Menggunakan Metode Penetration Testing dalam Menjaga Kapabilitas Keamanan Teknologi Informasi Pertahanan Negara,” *Jurnal Teknologi Penginderaan*, vol. 2, no. 1, pp. 105–122, 2020.

ISSN: 1978-8126
e-ISSN: 2527-7340

[5] A. F. Gentile, D. Macrì, E. Greco, and P. Fazio, "IoT IP Overlay Network Security Performance Analysis with Open Source Infrastructure Deployment," *Journal of Cybersecurity and Privacy*, vol. 4, no. 3, pp. 629–649, Aug. 2024, doi: 10.3390/jcp4030030.

[6] I. Made Sukarsa, I. Made, R. Pradana, and P. Wira Buana, "Implementasi Enkripsi dan Otentikasi Transmisi Data ZeroMQ Menggunakan Advanced Encryption Standard," *RESTI*, vol. 1, no. 3, pp. 1149–1156, 2017.

[7] D. Anggarini, B. P. Zen, and M. Pranata, "Security Analysis On Websites Using The Information System Assessment Framework (Issaf) And Open Web Application Security Version 4 (Owaspv4) Using The Penetration Testing Method," *Jurnal Pertahanan*, vol. 8, no. 3, pp. 2549–9459, 2022, doi: 10.33172/jp.v8.

[8] Q. Zhang, "An Overview and Analysis of Hybrid Encryption: The Combination of Symmetric Encryption and Asymmetric Encryption," in *2021 2nd International Conference on Computing and Data Science (CDS)*, Stanford, CA, USA: IEEE, 2021, pp. 616–622.

[9] * Budi, I. Darmawan, and F. Yudha, "Simulasi Dan Analisis Encryption Based Ransomware Untuk Memetakan Evolusi Ransomware," 2019.

[10] A. Rachmayanti and W. Wirawan, "Implementasi Algoritma Advanced Encryption Standard (AES) pada Jaringan Internet of Things (IoT) untuk Mendukung Smart Healthcare," *Jurnal Teknik ITS*, vol. 11, no. 3, 2022, doi: 10.12962/j23373539.v11i3.97042.

[11] Badan Siber dan Sandi Negara, "Honey Project," *https://bssn.go.id/honeynet-project/*, Jakarta, Jan. 12, 2023.

[12] A. Bastian, H. Sujadi, and L. Abror, "Analisis Keamanan Aplikasi Data Pokok Pendidikan (DAPODIK) Menggunakan Penetration Testing Dan SQL Injection," *INFOTECH Journal*, vol. 6, no. 2, pp. 65–70, 2020.

[13] V. Yuniati, G. Indriyanta, and A. Rachmat C., "Enkripsi Dan Dekripsi Dengan Algoritma Aes 256 Untuk Semua Jenis File," *Jurnal Informatika*, vol. 5, no. 1, 2011, doi: 10.21460/inf.2009.51.69.

[14] J. M. Parenreng, S. M. Mustari, and A. Wahid, "E-mail Security System Using El-Gamal Hybrid Algorithm and AES (Advanced Encryption Standard) Algorithm," *Internet of Things and Artificial Intelligence Journal*, vol. 2, no. 1, pp. 1–9, Feb. 2022, doi: 10.31763/iota.v2i1.510.

[15] A. Muhammad Abdullah and A. Muhamad Abdullah, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data Application of Petri Nets in Computer Networks View project Call for papers View project Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data," 2017. [Online]. Available: https://www.researchgate.net/publication/317615794

[16] K. Natarajan and V. Shaik, "Transparent Data Encryption: Comparative Analysis and Performance Evaluation of Oracle Databases," in *Proceedings - 2020 5th International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 137–142. doi: 10.1109/ICRCICN50933.2020.9296168.

[17] E. D. Madyatmadja, A. N. Hakim, and D. J. M. Sembiring, "Performance testing on Transparent Data Encryption for SQL Server's reliability and efficiency," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00520-z.

[18] B. P. Zen, Abdurahman, A. Zafia, I. N. Y. Putro, and F. D. Aditama, "Multi Socket Transmission System Application with Advanced Encryption Standard Algorithm to Support Confidential Medical Data Security," in *Proceedings of the 4th International Conference on Electronics, Biomedical Engineering, and Health Informatics*, A. and C. W. Triwiyanto Triwiyanto and Rizal, Ed., Singapore: Springer Nature Singapore, Apr. 2024, pp. 1–13. doi: https://doi.org/10.1007/978-981-97-1463-6_1.

[19] F. Al Rasyid and B. Parg Zen, "Dead Forensic Analysis Of Qutebrowser And Librewolf Browsers Using The Nist 800-86 Method," vol. 4, no. 5, pp. 1009–1019, 2023, doi: 10.52436/1.jutif.2023.4.5.688.

[20] Bita Parga Zen, Anggi Zafia, and Iwan Nofi Yono Putro, "Network Security Analysis Simulation at the GCS in the UCAV to support the Indonesian Defense Area," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 824–831, 2022, doi: 10.29207/resti.v6i5.4412.