BAB II
TINJAUAN PUSTAKA

2.1. Penelitian sebelumnya

Berdasarkan penelitian oleh Alexander Popov dan Iryna Burachonak pada tahun 2019, berjudul "*Protection of Information Stored in Database Means MS SQL Server (Transparent Data Encryption)*", membahas penerapan *Transparent Data Encryption* di MS *SQL Server*. Fokusnya terletak pada proses enkripsi dengan melibatkan hierarki kunci, seperti *Service Master Key (SMK)*, Kunci *Master Database*, dan Kunci Enkripsi *Database*. Metode *Transparent Data Encryption* dijelaskan sebagai cara mengenkripsi file *database* di tingkat halaman, memastikan keamanan data saat disimpan dan dibaca. Meskipun *Transparent Data Encryption* tidak memperbesar ukuran *database* terenkripsi, penelitian mencatat dampak kinerja dan kerentanan data di tingkat jaringan. Oleh karena itu, penelitian merekomendasikan penggunaan kombinasi enkripsi simetris dan asimetris untuk melindungi data secara menyeluruh[8].

Evaristus Didik Madyatmadja, Aditya Nur Hakim, dan David Jumpa Malem Sembiring pada tahun 2022 meneliti *"Performance Testing on Transparent Data Encryption for SQL Server's Reliability and Efficiency."* Penelitian ini mengeksplorasi dampak *Transparent Data Encryption* terhadap kinerja *Microsoft SQL Server* melalui serangkaian pengujian beban, stres, dan pencadangan. Hasilnya menunjukkan penurunan kinerja sistem, dengan penurunan sekitar 7% dalam kecepatan transaksi per menit dan hingga 15% dalam penggunaan *Central Processing Unit* , memori, dan durasi pencadangan. Meskipun demikian *Transparent Data Encryption* memiliki manfaat keamanan yang membuatnya menjadi pilihan praktis untuk melindungi data sensitif dalam sistem[9].

Natarajan K dan Vaheedbasha Shaik pada tahun 2020, berjudul "*Transparent Data Encryption : Comparative Analysis and Performance Evaluation of Oracle Databases*". Penelitian ini mengevaluasi dampak

Transparent Data Encryption pada kinerja *database* Oracle. Fokusnya adalah analisis *Central Processing Unit* , IO, dan RAM. *Transparent Data Encryption* terbukti meningkatkan optimalisasi *Central Processing Unit* tanpa memengaruhi penyimpanan, tetapi masih menimbulkan kekhawatiran terkait pemanfaatan RAM. Versi terbaru Oracle, khususnya 19c, menunjukkan peningkatan signifikan dalam optimalisasi *Central Processing Unit* dan penyimpanan melalui penggunaan *Transparent Data Encryption* . Namun, dampaknya terhadap RAM tetap menjadi isu utama[7].

Matthew Mcgiffen pada tahun 2022 melakukan penelitian berjudul *"What is Transparent Data Encryption ?"*. Penelitian ini membahas mengenai bentuk enkripsi data yang dilakukan secara teransparan oleh *Transparent Data Encryption* . Teknologi ini mengenkripsi semua data dalam *database*, termasuk file data, file log, file cadangan, dan file snapshot *database*[10].

Devlin Iskandar Saragih dan Paska Marto Hasugian pada tahun 2022 melakukan penelitian dengan judul "*Enkripsi Database Sekolah SMK Pembangunan Dengan Algoritma IDEA*." Penelitian ini membahas penerapan algoritma IDEA dalam mengamankan *database* sekolah, dengan fokus pada operasi enkripsi, dekripsi, dan proses pembuatan kunci. Algoritma IDEA dipilih untuk memberikan tingkat keamanan tinggi dengan kunci rahasia kompleks dan implementasi ekonomis[11].

Muhammad Apit Ruswandi dan Windarto pada tahun 2023 melakukan penelitian berjudul "*Enkripsi Database Sistem Informasi Helpdesk Dengan Algoritme Kriptografi AES-128 dan Vigenere Cipher*". Algoritma kriptografi AES-128 dan Vigenere Cipher di implementasikan pada penelitian ini untuk mengamankan *database* sistem helpdesk selama pandemi COVID-19. Hasilnya menunjukkan efektivitas enkripsi dalam memitigasi kebocoran data dan akses tidak sah, meskipun beberapa keterbatasan diidentifikasi, seperti peningkatan ukuran file *database* setelah enkripsi dan potensi kerentanan injeksi *SQL*[12].

Adelia Marwah Ujung dan Muhammad Irwan Padli Nasution pada tahun 2023 menyajikan penelitian berjudul "*Sistem Keamanan Database*."Penelitan ini mendalam tentang keamanan *database*. Penelitian ini mengeksplorasi aspek keamanan fisik, akses, jaringan, data, aplikasi, dan audit kepatuhan melalui tinjauan literatur dan studi kasus. Solusi potensial dan teknologi inovatif, seperti blockchain dan kecerdasan buatan, dibahas untuk meningkatkan keamanan *database*[13].

George S. Oreku pada tahun 2022 meneliti *"A Study of Online Database Servers: The Case of SQL - Injection, How Evil that could be?"* untuk mengidentifikasi kerentanan serangan injeksi *SQL* pada *server web* di *Tanzania*. Penelitian menyoroti bahwa sebagian besar server *database* online rentan terhadap serangan injeksi *SQL*, dengan rekomendasi mitigasi termasuk penggunaan pernyataan berparameter dan pengimplementasian kontrol akses yang lebih ketat[14].

Dwiky Al Asyam dan Endang Wahyu Pamungkas pada tahun 2023 secara khusus mengeksplorasi keamanan *database* aplikasi web terhadap serangan *SQL Injection* dalam penelitian berjudul "*Analisis Keamanan Database Aplikasi Web Dengan SQL Injection Menggunakan Penetration Tools*" Pengujian penetrasi *black box* dengan alat-alat seperti *SQLi Dumper, SQLmap, Nmap, dan Domain Whois* menyoroti rentannya aplikasi web tanpa perlindungan yang memadai[15].

Miloš Ilić, Lazar Kopanja, Dragan Zlatković, Milica Trajković, dan Dejana Ćurguz pada tahun 2021 melakukan perbandingan kinerja antara *Microsoft SQL Server* dan server Oracle dalam penelitian berjudul "*Microsoft SQL Server And Oracle: Comparative Performance Analysis*" Evaluasi ini menjadi panduan penting bagi para ahli komputer, terutama pengembang, dalam membuat keputusan saat mengembangkan aplikasi komputer, dengan mempertimbangkan perbedaan teoretis, karakteristik dasar, persyaratan sistem dan perangkat keras, keamanan, dan waktu eksekusi kueri[16].

Tabel 2.1 Penelitian Sebelumnya

Judul	Penulis dan Tahun penelitian	Pendekatan Metode	Persamaan	Perbedaan	Hasil
Protection of Information Stored in Database Means MS SQL Server (Transparent Data Encryption)	Alexander Popov dan Iryna Burachonak pada tahun 2019	*Transparent Data Encryption*	Penggunaan Teknologi *Transparent Data Encryption*	Penelitian ini hanya memberikan penjelasan terkait *Transparent Data Encryption*	Penerapan *Transparent Data Encryption* di MS *SQL Server* meningkatkan keamanan *database* melalui hierarki kunci. Meskipun dapat memengaruhi kinerja, kombinasi enkripsi simetris dan asimetris direkomendasikan. *Transparent Data Encryption* tidak mengatasi kerentanan data di tingkat jaringan.

Judul	Penulis dan Tahun penelitian	Pendekatan Metode	Persamaan	Perbedaan	Hasil
Performance Testing on Transparent Data Encryption for SQL Server's Reliability and Efficiency	Evaristus Didik Madyatmadja, Aditya Nur Hakim, dan David Jumpa Malem Sembiring pada tahun 2022	*Transparent Data Encryption*	Penggunaan Teknologi *Transparent Data Encryption*	Penelitian ini hanya berfokus ke bagian penilaian peforma dari *Transparent Data Encryption*	Hasil penelitian menunjukkan bahwa Implementasi *Transparent Data Encryption* pada sistem *database* dapat mengakibatkan penurunan kinerja sebesar 2-15%. Meskipun demikian, manfaat keamanan *Transparent Data Encryption* diakui lebih besar daripada dampak negatifnya, membuatnya tetap praktis untuk digunakan.

Judul	**Penulis dan Tahun penelitian**	**Pendekatan Metode**	**Persamaan**	**Perbedaan**	**Hasil**
Transparent Data Encryption : Comparative Analysis and Performance Evaluation of Oracle Databases	Natarajan K dan Vaheedbasha Shaik pada tahun 2020	*Transparent Data Encryption*	Penggunaan Teknologi *Transparent Data Encryption*	Penelitian ini hanya berfokus ke bagian penilaian peforma dari *Transparent Data Encryption*	Hasil penelitian menunjukkan bahwa *Transparent Data Encryption* memiliki dampak signifikan pada optimalisasi *Central Processing Unit* , IO, dan penyimpanan di versi *database* Oracle. Penelitian merekomendasikan solusi untuk meningkatkan kinerja, seperti penggunaan server terpisah, menghindari konfigurasi campuran, dan menerapkan RAM caging, atau menggunakan *Real Application Cluster (RAC).*

Judul	**Penulis dan Tahun penelitian**	**Pendekatan Metode**	**Persamaan**	**Perbedaan**	**Hasil**
What is Transparent Data Encryption ?	Matthew Mcgiffen	*Transparent Data Encryption*	Penggunaan Teknologi *Transparent Data Encryption*	Penilitan ini hanya berfokus pada apa saja yang akan di enkripisi pada teknologi *Transparent Data Encryption*	*Transparent Data Encryption* adalah bentuk enkripsi data yang berlangsung secara otomatis di latar belakang dengan melibatkan enkripsi seluruh data dalam *database*. Namun, *Transparent Data Encryption* tidak melibatkan enkripsi data di memori, data yang dikembalikan melalui jaringan, atau data yang diterima oleh klien sebagai hasil dari kueri. Kunci enkripsi *Transparent Data Encryption* disimpan dalam *database* yang terenkripsi, dengan kunci itu sendiri tetap terenkripsi oleh objek di luar *database*.

Judul	**Penulis dan Tahun penelitian**	**Pendekatan Metode**	**Persamaan**	**Perbedaan**	**Hasil**
Enkripsi Database Sekolah SMK Pembangunan Dengan Algoritma IDEA	Devlin Iskandar Saragih dan Paska Marto Hasugian pada tahun 2022	Enkripsi *Database*	Penerapan Enkripsi *Database*	Penerapan algoritma IDEA	Hasil penelitian membahas penerapan algoritma *IDEA* untuk mengamankan *database* sekolah dengan proses enkripsi dan dekripsi. Algoritma ini menggunakan kunci rahasia kompleks dan melibatkan operasi aljabar *XOR*, penjumlahan modulo, dan perkalian modulo. Proses enkripsi *IDEA* dilakukan melalui pembagian teks, transformasi, dan penggunaan subkunci.

Judul	**Penulis dan Tahun penelitian**	**Pendekatan Metode**	**Persamaan**	**Perbedaan**	**Hasil**
Enkripsi Database Sistem Informasi Helpdesk Dengan Algoritme Kriptografi AES-128 dan Vigenere Cipher	Muhammad Apit Ruswandi dan Windarto pada tahun 2023	Enkripsi *Database*	Penerapan Enkripsi *Database*	Penerapan algoritma *AES-128* dan *Vigenere Cipher*	Hasil penelitian menunjukkan implementasi algoritma *AES-128* dan *Vigenere Cipher* pada sistem informasi helpdesk bertujuan mencegah penyalahgunaan data. Meskipun memiliki antarmuka sederhana, masih perlu perbaikan terkait ukuran file setelah enkripsi dan potensi kerentanan *injeksi SQL*.

Judul	**Penulis dan Tahun penelitian**	**Pendekatan Metode**	**Persamaan**	**Perbedaan**	**Hasil**
Sistem Keamanan Database	Adelia Marwah Ujung dan Muhammad Irwan Padli Nasution pada tahun 2023		Penggunan *Database*	Pemaparan *database*	Hasil penelitian melibatkan pemahaman menyeluruh keamanan *database*, menekankan kerahasiaan, integritas, dan ketersediaan. Menyuarakan risiko dan strategi penanggulangan, serta memaparkan ancaman umum dan langkah-langkah keamanan.

Judul	**Penulis dan Tahun penelitian**	**Pendekatan Metode**	**Persamaan**	**Perbedaan**	**Hasil**
A Study of Online Database Servers: The Case of SQL - Injection, How Evil that could be?	George S. Oreku pada tahun 2022	*Black box*	Pembahasan mengenai keamanan data	Penetration testing *database*	Hasil penelitian ini menggarisbawahi prevalensi kerentanan injeksi *SQL* di *server web* di Tanzania dan pentingnya tindakan proaktif untuk mengatasi kerentanann dan meningkatkan keamanan.
Analisis Keamanan Database Aplikasi Web Dengan SQL Injection Menggunakan Penetration Tools	Dwiky Al Asyam dan Endang Wahyu Pamungkas pada tahun 2023	*Black box*	Pada penelitian membahas tentang keamanan data	Penetration testing *database*	Hasil penelitian ini menyoroti pentingnya menganalisis dan mengamankan aplikasi web dari serangan *SQL Injection* . Metode pengujian penetrasi *black box* digunakan untuk menilai kerentanan.

Judul	**Penulis dan Tahun penelitian**	**Pendekatan Metode**	**Persamaan**	**Perbedaan**	**Hasil**
Microsoft SQL Server And Oracle: Comparative Performance Analysis	Miloš Ilić, Lazar Kopanja, Dragan Zlatković, Milica Trajković, dan Dejana Ćurguz pada tahun 2021	Evaluasi system	Penggunaan *Microsoft SQL Server*	Pada penelitian ini terdapat perbandingan *SQL Server* dan Oracle	Hasil penelitian membandingkan kinerja *Microsoft SQL Server* dan Oracle menunjukkan bahwa *SQL Server* memiliki waktu eksekusi query lebih baik, meskipun Oracle unggul dalam dukungan sistem operasi dan bahasa pemrograman.

2.2. Dasar Teori

2.2.1 Data Dan Informasi

Data merupakan kumpulan deskripsi dasar atau informasi dasar mengenai suatu objek atau peristiwa yang diperoleh melalui observasi dan dapat diubah menjadi bentuk lebih kompleks seperti informasi, basis data, atau solusi dari suatu masalah tertentu[17]. Informasi merupakan hasil analisis dan penyusunan data, sehingga informasi adalah data yang telah diatur sesuai dengan kebutuhan individu tertentu[18].

Fungsi Data

- Data berfungsi sebagai panduan dalam pengambilan keputusan untuk menyelesaikan masalah.
- Data dapat menjadi dasar atau panduan dalam penelitian atau perencanaan.
- Data dapat digunakan sebagai pedoman dalam pelaksanaan kegiatan tertentu.
- Data menjadi dasar untuk mengevaluasi suatu kegiatan.

Data atau informasi yang didapat melalui media elektronik memiliki nilai tinggi, seperti data kependudukan dan demografis di Indonesia, termasuk Kartu Keluarga (KK), Nomor Induk Kependudukan (NIK), dan Kartu Tanda Penduduk (KTP). Penting untuk menjaga data ini agar tidak disalahgunakan oleh pihak yang tidak bertanggung jawab[19].

2.2.2 *Database*

Database atau disebut basis data adalah sekumpulan data mengenai suatu objek ataupun peristiwa yang saling terkait antara satu dengan yang lain. Basis data mencerminkan aspek tertentu dari dunia nyata. Basis data adalah sekumpulan data dari berbagai macam sumber yang logis memiliki makna tersirat. Basis data dibangun, dirancang, dan dikumpulkan untuk tujuan tertentu. Sistem Manajemen Basis Data (DBMS) memiliki empat jenis komponen utama yaitu:

a. Perangkat keras

b. Data

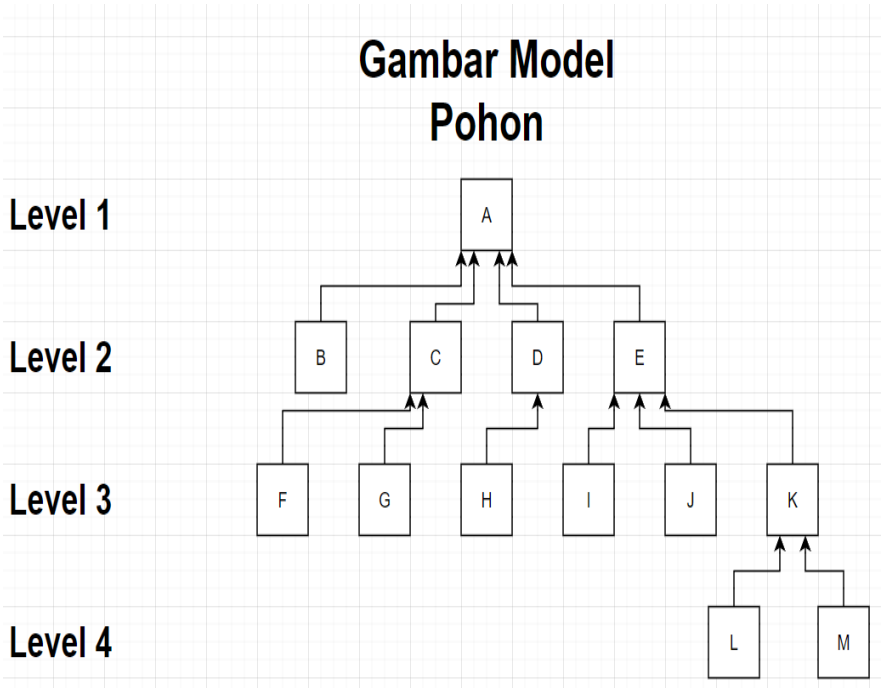
c. Perangkat lunak

d. Pengguna

Model basis data menjelaskan hubungan antara catatan-catatan yang tersimpan di dalamnya. Beberapa literatur menyebut ini sebagai struktur data logis. Berikut diantaranya :

a) Pohon

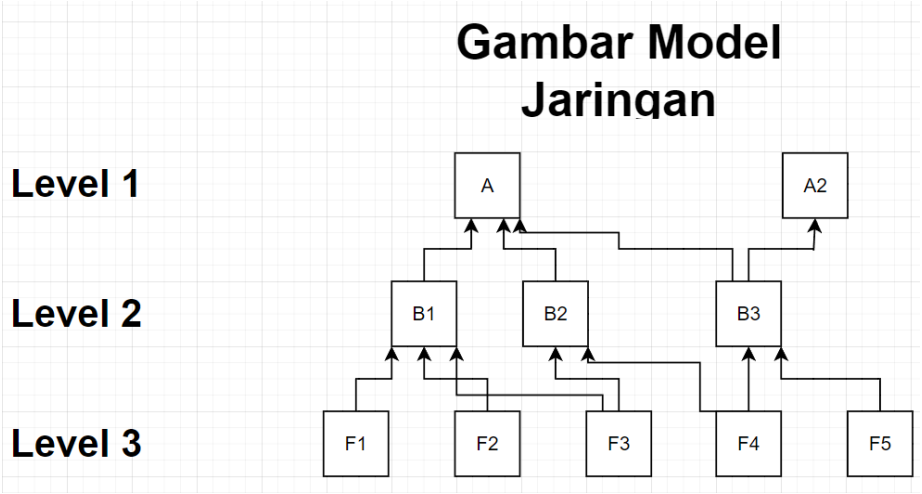
Model Pohon menggunakan struktur hierarkis untuk menunjukkan hubungan antara elemen induk dan anak. Setiap node, yang biasanya digambarkan dengan lingkaran atau kotak, merepresentasikan sekumpulan atribut. Node yang terhubung ke node lain di level bawah disebut sebagai induk. Setiap induk bisa memiliki satu hubungan 1:1 atau beberapa hubungan 1 dengan anak, namun setiap anak hanya memiliki satu induk. Node-node di bawah induk disebut anak. Node induk yang tidak memiliki induk disebut akar, sementara node yang tidak memiliki anak disebut daun. Hubungan antara anak dan induk disebut cabang. Gambar di bawah ini menunjukkan contoh dengan 4 level dan 13 node.



Gambar 2. 1 Model Pohon [18]

b) Jaringan

Model jaringan mirip dengan model hirarkis, namun dengan perbedaan bahwa sebuah node anak dapat memiliki lebih dari satu node orang tua.



Gambar 2. 2 Model Jaring [18]

c) Relasional

Model relasional dikenal karena kesederhanaannya, yang membuatnya mudah digunakan dan dipahami oleh pengguna. Model ini adalah salah satu yang paling populer saat ini. Dalam model ini, data disusun dalam bentuk tabel dua dimensi yang disebut relasi atau tabel. Setiap relasi terdiri dari baris-baris (dikenal sebagai tupel) dan kolom-kolom (dikenal sebagai atribut). Struktur relasi ini didesain untuk mengurangi redundansi data dengan menggunakan kunci asing untuk menghubungkan satu relasi dengan relasi lainnya. Dalam konteks model relasional, jumlah tupel dalam sebuah relasi disebut kardinalitas [18].

Tabel 2. 2 Tabel Pelanggan

ID_Pelanggan	Nama_Pelanggan	Alamat	Nomor_Telpon
01	Budi	Purwokerto	0812345678
02	Tuti	Purbalingga	0898765432

Tabel 2. 3 Tabel Produk

ID_Produk	Nama_Produk	Harga
P1	Sampo	5000
P2	Sabun	2000

Tabel 2. 4 Tabel Pesanan

Id_Pesanan	Tanggal_Pesanan	ID_Pelanggan
I1	05-02-2023	01
I2	15-01-2023	02

Tabel 2. 5 Detail Pesanan

ID_Detail_Pesanan	ID_Pesanan	ID_Produk	Jumlah	Subtotal
IDP 1	I1	P1	5	25000
IDP 2	I2	P2	5	10000

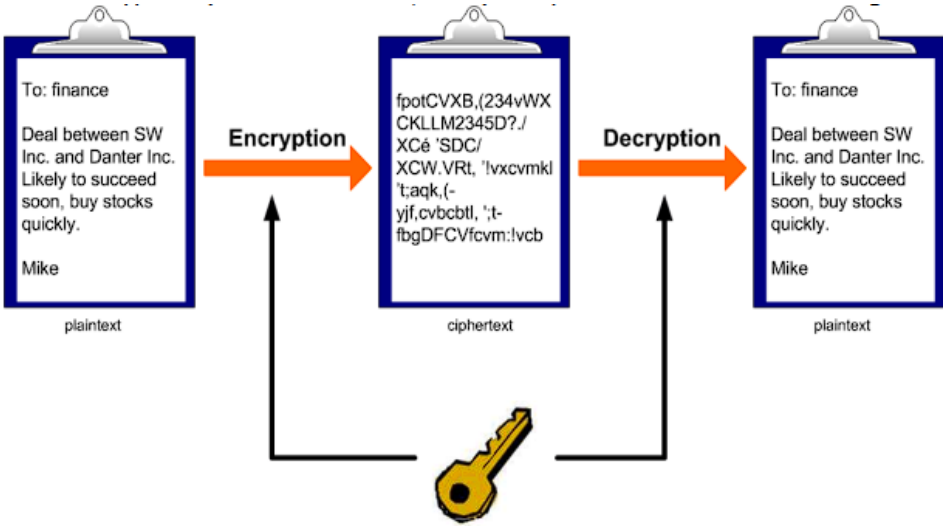
2.2.3 *SQL Server*

Microsoft SQL Server adalah sistem *database* yang terdiri dari banyak komponen, termasuk Mesin *Database*, Layanan Analisis, Layanan Pelaporan, *Database* Grafik *SQL Server*, Layanan Pembelajaran Mesin *SQL Server*, dan beberapa komponen lainnya[20]. Keamanan adalah hal yang penting dalam sistem penyimpanan dan pemrosesan data apa pun, dan *SQL Server* bangga menjadi *database* paling aman selama delapan tahun terakhir menurut *National Institute of Standards and Technology's (NIST) Comprehensive Vulnerability Database* dan menduduki posisi ke tiga sebagai database terpopuler. *SQL Server* mendukung keamanan dan kepatuhan perusahaan dengan fitur keamanan seperti *Transparent Data Encryption* , Auditing, Row-Level Security, Dynamic Data Masking, dan Always Encrypted. *SQL Server* 2019 menambahkan dukungan untuk enclave aman dalam *Always Encrypted* untuk memungkinkan perhitungan kaya pada data yang terenkripsi [21].

2.2.4 *Crypthografy*

Menurut Munir, kriptografi (*cryptography*) berasal dari kata Yunani kuno '*cryptós*' yang memiliki arti 'rahasia' dan '*gráphein*' yang memiliki arti 'menulis'. Oleh karena itu, kriptografi dapat disebut sebagai 'penulisan rahasia'[22]. Secara khusus, kriptografi mencakup pengembangan teknik-teknik enkripsi dan dekripsi untuk menjaga privasi dan integritas data.

Kriptografi digunakan untuk mengamankan pesan yang dikirim dari satu tempat ke tempat lain. Melalui teknik kriptografi, pesan asli atau disebut *plaintext* melewati proses enkripsi dengan menggunakan sebuah kunci menjadikan informasi diubah secara acak sehingga sulit dipahami, ini disebut *ciphertext*. Kunci ini hanya diketahui oleh kedua belah pihak yang melakukan pengirim dan penerima pesan, sehingga memungkinkan penerima untuk mengubah kembali *ciphertext* menjadi *plaintext*. Dengan demikian, pihak lainnya tidak dapat mengakses isi pesan tersebut dan hanya akan melihat informasi yang tidak bermakna[23].



Gambar 2. 3 Proses Enkripsi [23]

Tujuan mendasar ilmu kriptografi ada empat dan ini juga merupakan aspek yang sangat penting untuk keamanan informasi, yaitu[24]:

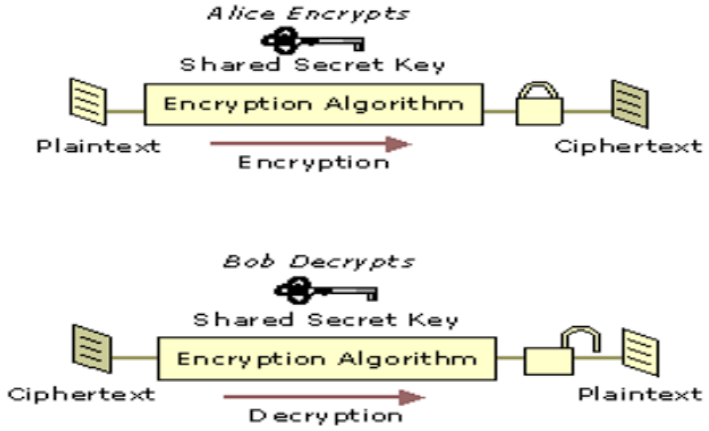
1. Kerahasiaan: Tujuan ini debentuk agar informasi tetap tersembunyi dari siapa pun kecuali pihak yang memiliki otoritas atau kunci yang tepat untuk mengaksesnya. Ini memastikan bahwa hanya individu atau sistem yang berwenang yang dapat membaca data yang telah dienkripsi.
2. Integritas Data: Tujuan ini berfokus pada pencegahan perubahan data oleh pihak-phak yang tidak memiliki wewenang. Kemampuan untuk mendeteksi setiap upaya manipulasi data, termasuk penyisipan, penghapusan, atau penggantian data yang sah dengan data yang tidak sah harus dimiliki oleh system.

3. Autentikasi: Ini terkait dengan verifikasi identitas, baik dari sistem maupun informasi yang ditransmisikan. Pihak-pihak yang berkomunikasi harus bisa bertukar identitas satu sama lain. Selain itu, informasi yang dikirim melalui saluran komunikasi, harus bisa diverifikasi keasliannya, termasuk konten dan juga waktu pengirimannya.
4. Nirpenyangkalan: Tujuan ini adalah untuk melakukan upaya pencegahan penyangkalan oleh pengirim atau pencipta informasi. Dengan kata lain, pengirim tidak bisa menyangkal telah mengirimkan pesan, dan pencipta informasi tidak bisa menyangkal telah membuatnya.

2.2.4.1 Jenis Enkripsi

a. Simetris

Kriptografi ini menggunakan satu kunci yang sama untuk melakukan proses enkripsi dan dekripsi. Kriptografi jenis ini banyak digunakan karena kinerjanya yang cepat dan efisiensi konsumsi daya komputasi yang rendah. Beberapa contoh algoritma yang digunakan dalam kriptografi simetris seperti *Advanced Encryption Standard* , Blowfish, Triple Data *Encryption* Standard (DES), dan Rivest Cipher 4 (RC4).

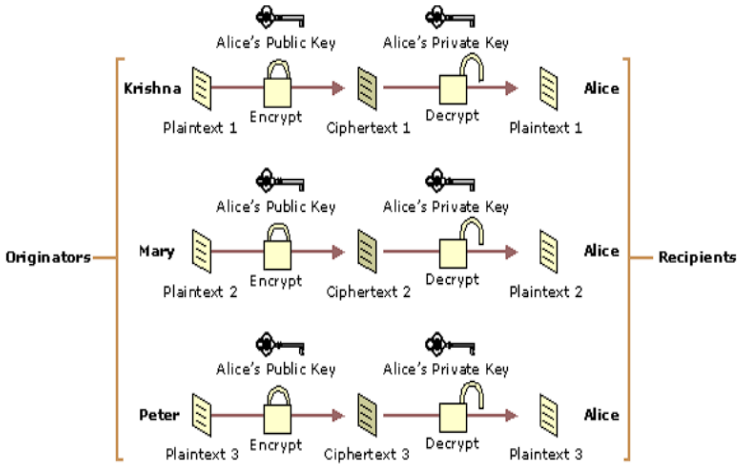


Gambar 2. 4 Enkripsi Simetris [25]

b. Asimetris

Dalam kriptografi asimetris, proses enkripsi dan dekripsi menggunakan dua kunci yang berbeda. Satu kunci bersifat publik dan dapat disebarkan, sementara kunci lainnya bersifat privat dan harus

dirahasiakan. Contoh algoritma yang menggunakan kriptografi asimetris adalah Rivest-Shamir-Adleman (RSA) dan Diffie-Hellman [25].



Gambar 2. 5 Enkripsi Asimetris [25]

2.2.5 *Transparent Data Encryption*

Transparent Data Encryption disebut sebagai bentuk enkripsi "transparan". Artinya, proses enkripsi dan dekripsi data sepenuhnya dilakukan di latar belakang. Kueri yang ditulis untuk mengakses data tidak berubah ketika *Transparent Data Encryption* diaktifkan atau tidak. Jadi, mengaktifkan *Transparent Data Encryption* tidak berdampak pada fungsionalitas aplikasi, tidak memerlukan pemfaktoran ulang kode, dan oleh karena itu relatif mudah untuk diterapkan. *Transparent Data Encryption* mengenkripsi semua data dalam *database*, jadi tidak perlu memilih item data mana yang akan dienkripsi[10].

2.2.5.1 Algoritma *Advanced Encryption Standard*

Transparent Data Encryption di *Microsoft SQL Server* memanfaatkan algoritma *Advanced Encryption Standard* untuk proses enkripsinya. *Advanced Encryption Standard* adalah algoritma kriptografi yang digunakan untuk mengamankan data. Sebagai sebuah cipher simetris berbasis blok, *Advanced Encryption Standard* mampu melakukan enkripsi dan dekripsi informasi. Enkripsi mengubah data yang dapat dibaca menjadi bentuk yang tidak dapat dibaca yang disebut *ciphertext*, sedangkan dekripsi mengembalikan *ciphertext* ke bentuk

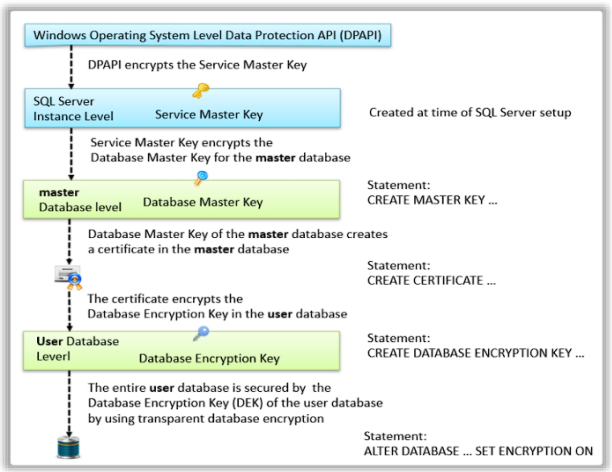
aslinya, yaitu *plaintext*. *Advanced Encryption Standard* diadopsi sebagai standar enkripsi oleh *National Institute of Standards and Technology (NIST)* dan digunakan secara luas di berbagai industri. Algoritma AES menggunakan kunci kriptografi dengan ukuran 128, 192, dan 256 bit untuk mengenkripsi dan mendekripsi data dalam blok 128 bit[26].

Tabel 2. 6 Perbandingan Algoritma *Advanced Encryption Standard*

Algoritma	Jumlah Key (NK)	Ukuran Blok (Nb)	Jumlah Putaran (NR)
AES – 128	4	4	10
AES – 192	6	4	12
AES – 256	8	4	14

Proses enkripsi dalam algoritma *Advanced Encryption Standard* melibatkan empat jenis transformasi *bytes*: *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey*. Pada awalnya, input yang telah disalin ke dalam state akan mengalami transformasi *byte AddRoundKey*. Kemudian, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang sebanyak Nr. Proses ini dalam algoritma *Advanced Encryption Standard* dikenal sebagai fungsi putaran (*round function*). Namun, pada putaran terakhir, terjadi sedikit perbedaan dimana state tidak mengalami transformasi *MixColumns*[23].

2.2.5.2 *Transparent Database Encryption Architecture*



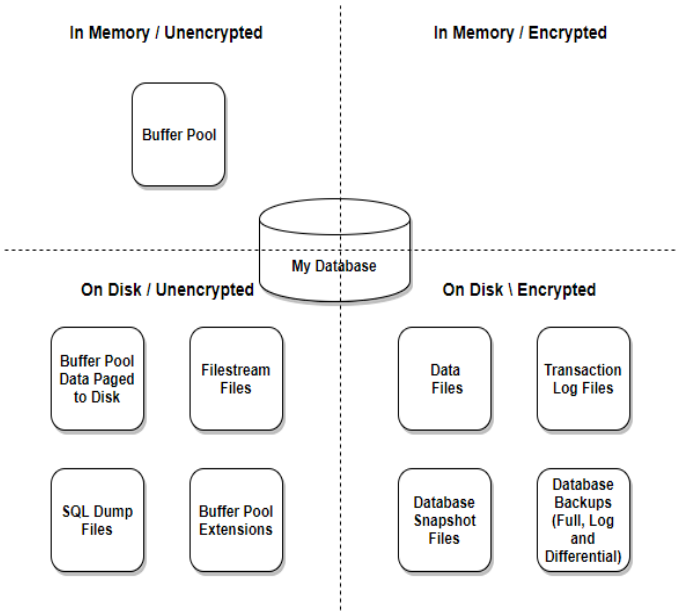
Gambar 2. 6 Arsitektur *Transparant Data Encryption* [27]

Transparent Data Encryption adalah fitur bawaan untuk melindungi data pada saat disimpan (*data at rest*). Mengenkripsi data saat disimpan (*data at rest*) dapat melindunginya dari serangan siber karena penyerang tidak akan dapat membaca data yang terenkripsi.

1. Instance *SQL Server*
   - *Service Master Key (SMK)* dibuat saat mengatur instans *SQL Server*.
2. *Database Master*
   - *Data Protection API* (*DPAPI*) mengenkripsi *Service Master Key (SMK)*. DPAPI melindungi *Service Master Key (SMK)* di *database Master*.
   - *Service Master Key (SMK)* mengenkripsi *Database Master Key* (*DMK*). Oleh karena itu, *Service Master Key (SMK)* melindungi *Database Master Key* (*DMK*).
   - *Database Master Key* (*DMK*) membuat sertifikat. Oleh karena itu, melindungi kunci pribadi sertifikat.
   - Sertifikat mengenkripsi *Database Encryption* Key (DEK). Oleh karena itu, sertifikat melindungi *Database Encryption* Key (DEK) di *database* pengguna.
3. *Database* Pengguna
   - Seluruh *database* pengguna diamankan oleh *Database Encryption* Key (DEK) menggunakan enkripsi *Transparent Data Encryption* . *Database Encryption* Key (DEK) melindungi data di *database* pengguna.
   - *Transparent Data Encryption* melakukan enkripsi tingkat halaman; halaman-halaman dienkripsi sebelum ditulis ke disk, dan didekripsi ketika pengguna ingin membacanya [27].

2.2.5.3 *Encypted* dan *Unencrypted*

Yang dienkripsi dan tidak dienkripsi oleh *Transparent Data Encryption* dirangkum dalam diagram di bawah ini:



Gambar 2. 7 Encrypted dan Unencrypted [10]

Dalam teknologi *Transparent Data Encryption* di *Microsoft SQL Server*, enkripsi diterapkan pada elemen-elemen kunci seperti *In-Memory Encrypted Data Files, Transaction Log Files, Database Snapshot Files,* dan *Database Backups on Disk Encrypted.* Ini menciptakan tingkat keamanan yang konsisten di berbagai lokasi penyimpanan, melindungi data selama penyimpanan dan perpindahan antar tempat penyimpanan. Sebagai kontrast, konsep keamanan dan enkripsi data dalam *SQL Server* mencakup elemen-elemen seperti *Unencrypted Buffer Pool in Memory*, yang meningkatkan kinerja *database* dengan menyimpan data tanpa enkripsi, dan *Unencrypted Buffer Pool Data Pages to Disk*, yang terjadi saat data dari *Buffer Pool* harus ditulis kembali ke disk tanpa enkripsi. *FileStream Files* memungkinkan penyimpanan *Binary Large Object* (*BLOB)* dengan opsi enkripsi sesuai kebijakan keamanan. *SQL Dump Files*, umumnya tidak dienkripsi, berisi informasi keadaan memori *SQL Server* dan sering digunakan untuk analisis dan *debugging*. *Buffer Pool Extensions on Disk*, memungkinkan penggunaan *Solid-state drive* (*SSD)* sebagai tambahan untuk *Buffer Pool*, dapat dienkripsi sesuai kebijakan sistem. Dalam kedua konteks ini, kebijakan keamanan dan enkripsi bervariasi sesuai kebutuhan organisasi dan konfigurasi *SQL Server*, dan implementasi enkripsi harus mempertimbangkan setiap komponen yang terlibat.

2.2.5.4 Parameter Status Enkripsi

Keberhasilan dari sebuah *database* di enkripsi adalah dengan cara melihat status dari *Encryption*_state dengan ketentuan sebagai berikut [28] :

Tabel 2. 7 Status Enkripsi

Value	Keterangan
0	*No database Encryption key present, no Encryption*
1	*Unencrypted*
2	*Encryption in progress*
3	*Encrypted*
4	*Key change in progress*
5	*Decryption in progress*
6	*Protection change in progress (The certificate or asymmetric key that is encrypting the database Encryption key is being changed)*

Indikator di atas menjadi pedoman utama dalam menilai hasil penerapan *Transparent Data Encryption* pada *database*. Evaluasi status *Encryption_state* memberikan gambaran terkait integritas dan efektivitas enkripsi, menjadi penting untuk memastikan bahwa *database* telah dienkripsi dan data terlindungi sesuai dengan standar keamanan yang diinginkan.

2.2.5.5 Perbandingan Tingkat Enkripsi

Transparent Data Encryption diterapkan di berbagai lingkungan dan sistem yang memerlukan perlindungan data tingkat tinggi, terutama untuk melindungi data yang disimpan (*data at rest*). Semua bergantung pada kebutuhan keamanan spesifik organisasi. Menerapkan *Transparent Data Encryption* dapat membantu memastikan bahwa data sensitif dilindungi dan hanya dapat diakses oleh pengguna resmi dengan kunci dekripsi yang tepat.

Berikut adalah bagan perbandingan berbagai tingkat enkripsi yang tersedia di berbagai sistem manajemen *database* relasional[29]:

Tabel 2. 8 Perbandingan Tingkat Enkripsi antara sistem manajemen *database*

Tingkat Enkripsi	*Oracle*	My*SQL*	*Microsoft SQL Server*	Postgres*SQL*
Tingkat *Cluster*	*Transparent Data Encryption*	*Pulgin InnoDB*	*Transparent Data Encryption*	T/A
Tingkat *Database*	*Transparent Data Encryption*	*Encryption at Rest*	*Transparent Data Encryption*	T/A
Tingkat Tabel	*Transparent Data Encryption*	*Encryption at Rest*	*Transparent Data Encryption*	T/A
Tingkat Kolom	*Transparent Data Encryption*	*Encryption at Rest*	*Transparent Data Encryption*	*pgcrypto*

Berdasarkan tabel diatas, dapat dilihat perbandingan metode enkripsi data pada berbagai tingkat dalam sistem manajemen basis data (DBMS) seperti *Oracle, MySQL, Microsoft SQL Server,* dan *PostgreSQL*. Di tingkat *cluster*, *Oracle* dan *Microsoft SQL Server* menggunakan *Transparent Data Encryption* , *MySQL* menggunakan *InnoDB Plugin*, dan *PostgreSQL* tidak memiliki dukungan langsung. Di tingkat *database*, *Oracle* dan *Microsoft SQL Server* menggunakan *Transparent Data Encryption* , *MySQL* menyediakan enkripsi data yang disimpan, dan *PostgreSQL* tidak memiliki dukungan langsung. Pada tingkat tabel, *Oracle* dan *Microsoft SQL Server* menggunakan *Transparent Data Encryption* , *MySQL* tetap menggunakan enkripsi data yang disimpan, sementara *PostgreSQL* tidak memiliki fitur ini. *Transparent Data Encryption* mengenkripsi seluruh file data dan log, sehingga semua data dalam tabel dan kolom yang ada di file tersebut terlindungi. Dengan demikian, *Transparent Data Encryption* juga melindungi data pada tingkat kolom. Di tingkat kolom, *Oracle* dan *Microsoft SQL Server* masih menggunakan *Transparent Data Encryption* , *MySQL* menyediakan enkripsi kolom melalui enkripsi data yang disimpan, dan *PostgreSQL* menggunakan modul *pgcrypto*. *Transparent Data Encryption* mengenkripsi data otomatis sebelum disimpan dan mendekripsi saat dibaca, sedangkan *pgcrypto* di *PostgreSQL* memberikan fleksibilitas untuk enkripsi data sensitif pada tingkat kolom.