BAB II

LANDASAN TEORI

2.1 Teori yang Digunakan

2.1.1 *Database*

Database adalah sebuah kumpulan data atau informasi yang disimpan ke dalam sebuah sistem komputer secara sistematik yang membentuk sebuah *file* [22]. *Database* juga dapat diartikan sebagai suatu wadah berkumpulnya berkas yang terkomputerisasi, atau juga dapat diartikan sebagai sebuah sistem terkomputerisasi yang berfungsi sebagai pemelihara informasi yang dapat menyediakan data saat dibutuhkan [22, 23]. *Database* secara sederhana juga dapat diartikan sebagai sebuah bentuk kumpulan data dan informasi yang bermanfaat yang disimpan secara terorganisir dalam sebuah tata cara khusus, yang dapat digunakan dan dimanfaatkan sesuai kebutuhan dari suatu organisasi [22, 24]. Dari pengertian-pengertian tersebut, *database* dapat diartikan sebagai sebuah bentuk representasi dari sebuah kumpulan data dan informasi yang disimpan dalam suatu kesatuan tanpa adanya redudansi yang tidak perlu, yang dapat memenuhi berbagai kebutuhan. Dalam konteks ini, *database* dapat didefinisikan sebagai sebuah kumpulan data dan informasi yang saling berkaitan pada sebuah subjek khusus dengan tujuan khusus, yang berbentuk sebuah susunan *record* data dari sebuah organisasi atau perusahaan, yang disusun dengan tata cara khusus yang terorganisir dan terintegrasi dengan menggunakan metode tertentu dalam sebuah sistem komputer yang mampu untuk memenuhi kebutuhan informasi secara optimal saat dibutuhkan oleh sebuah organisasi atau perusahaan [25]. *Database* menjadi kebutuhan utama sebuah organisasi atau perusahaan untuk dapat menyimpan dan mengorganisir data penting yang mereka miliki.

2.1.2 *Database Management System*

Database management system (DBMS) adalah sebuah perangkat lunak (*software*) yang dibuat dengan tujuan untuk mengatur atau mengorganisir akses-akses di dalam sebuah *database*, yang dapat melayani dan menyediakan data sesuai dengan kebutuhan pengguna yang akan melakukan akses ke dalam sebuah *database*. Sistem *database management system* dapat diartikan sebagai sebuah sistem yang berfungsi sebagai sebuah sistem proses manajemen pada *database* [26]. *Database management system* dapat juga diartikan sebagai sebuah sistem perangkat lunak yang memungkinkan sebuah organisasi atau perusahaan untuk dapat mendefinisikan akses, membuat atau memelihara akses yang terkontrol ke dalam *database* organisasi atau perusahaan tersebut [27]. Penggunaan *database management system* yang baik dapat memberikan beberapa keuntungan, seperti berkurangnya redudansi data. Keberadaan redudansi atau pengulangan data yang sama pada dalam sebuah *database* akan memboroskan ruang dalam *database* dan akan menghambat efisiensi dari kinerja *database*. Dengan menggunakan sistem DBMS yang baik, redudansi data dapat diminimalisir. Sistem DBMS yang baik juga dapat meningkatkan integritas data dalam *database*. Berkurangnya redudansi data mengartikan bahwa integritas data dalam *database* juga membaik. Sistem DBMS juga dapat meningkatkan keamanan sebuah *database.* Dengan menggunakan DBMS, akses ke data dapat dibatasi, sehingga tidak sembarangan orang bisa mengakses informasi penting yang ada di dalam *database*. Penggunaan sistem DBMS juga menawarkan prosedur-prosedur standar seperti penambahan, *edit*, atau penghapusan data, juga validasi data yang dapat meningkatkan kemudahan pemeliharaan data di dalam *database* [26, 27]. Penggunaan DBMS juga memungkinkan penyimpanan dan pengambilan data yang efisien karena dapat mengurangi kompleksitas akses data.

2.1.3 Proteksi Data & *Database*

Proteksi data merupakan sebuah bentuk perlindungan data suatu individu berkaitan dengan data pribadi yang tersimpan di dalam sebuah sistem komputer [29]. Proteksi data adalah sebuah bentuk perlindungan yang mencakup pengumpulan data, penyebaran data, dan teknologi. Proteksi data dikenal sebagai privasi informasi. Proteksi data harus diterapkan ke dalam semua bentuk data, baik dalam bentuk data individu ataupun data perusahaan. Proteksi data juga memiliki kaitan dengan integritas data, dimana data hanya boleh diakses oleh orang yang memiliki hak akses. Proteksi data secara umum berarti melindungi data dari akses yang tidak sah, untuk menghindari data jatuh ke tangan orang lain selain pemiliknya [30]. *Database* sebagai tempat penyimpanan data juga harus diproteksi untuk menghindari risiko kebocoran atau pencurian data dari akses yang tidak sah. Proteksi *database* berkaitan erat dengan ancaman yang disengaja maupun yang tidak disengaja. Risiko-risiko dan ancaman-ancaman terhadap *database* dapat mengurangi ataupun menghilangkan tujuan dari keamanan *database*, seperti integritas data (*integrity*), ketersediaan data (*availability*), dan kerahasiaan data (*privacy*) [30]. Proteksi *database* tidak hanya berbicara mengenai data yang terdapat di dalam *database* saja, tetapi juga keamanan pada bagian lain yang berpengaruh pada keamanan *database*, seperti jaringan, sistem operasi, bangunan tempat fisik *database*, dan orang orang yang memiliki peluang akses ke dalam sistem [31]. Dalam organisasi seperti perusahaan, keamanan data sangat diperhatikan, kerahasiaan data-data yang dimiliki oleh perusahaan harus diproteksi dari risiko-risiko dan ancaman-ancaman yang ada terhadap keamanan data. Risiko-risiko seperti kebocoran data dan pencurian data bisa dihindari dengan memberikan proteksi pada data dan *database* yang dimiliki, salah satunya bisa menggunakan cara enkripsi [32].

2.1.4 Enkripsi

Enkripsi merupakan sebuah proses teknikal yang mengubah sebuah data atau informasi menjadi sebuah kode, sehingga data atau informasi tidak bisa dibaca secara langsung. Enkripsi secara sederhana juga bisa diartikan sebagai proses mengacak data, sebelum proses dekripsi dilakukan untuk membaca data atau informasi yang dienkripsi. Data atau informasi yang tidak terenkripsi dikenal dengan "*plaintext*", sedangkan data atau informasi yang telah dienkripsi dikenal sebagai "*ciphertext*" [33]. Dari pengertian tersebut, enkripsi juga dapat diartikan sebagai proses konversi suatu "*plaintext*" menjadi sebuah "*ciphertext*". Enkripsi juga merupakan bagian dari kriptografi, dan menjadi sebuah elemen penting dalam menjaga kerahasiaan data atau informasi [34]. Enkripsi bisa dilakukan dalam level data juga dalam level *database*. Enkripsi *database* bisa menjadi solusi untuk melindungi data-data yang ada di dalam suatu *database*. Bagi organisasi khususnya perusahaan, enkripsi *database* menjadi solusi yang tepat untuk melindungi data-data penting dari kebocoran dan pencurian data. Enkripsi *database* dapat diartikan sebagai sebuah proses yang menggunakan sebuah algoritma tertentu dengan tujuan memodifikasi data yang ada di dalam suatu *database* menjadi sebuah kode rahasia yang tidak dapat dibaca tanpa proses dekripsi [35]. Dengan melakukan enkripsi *database*, ancaman akan peretasan *database* juga akan berkurang karena data-data yang ada di dalam *database* sudah terenkripsi [35, 36]. Proses enkripsi *database* akan memastikan bahwa data-data yang ada di dalam *database* tersebut tidak dapat dibaca oleh pihak-pihak yang mungkin memiliki niat jahat untuk meretas *database* atau mencuri data-data yang ada di dalam *database*. Data yang tidak bisa dibaca ini tidak memiliki *value*, sehingga akan mengurangi niat orang untuk meretas *database* atau mencuri data yang ada di dalamnya [37].

2.1.5 Enkripsi Kolom

Enkripsi kolom (*column-level encryption*) adalah salah satu teknik enkripsi yang banyak digunakan dalam melakukan proteksi *database* untuk melindungi data-data dan informasi-informasi sensitif yang ada pada kolom-kolom tertentu dalam tabel di dalam *database* [38]. Dalam MySQL, enkripsi kolom dapat diimplementasikan dengan menggunakan kunci simetris ataupun kunci asimetris. Penggunaan kunci simetris adalah metode yang paling umum digunakan dalam melakukan enkripsi kolom, dimana setiap kolom yang akan dienkripsi akan memiliki kunci yang unik, dan data pada kolom tersebut akan memiliki kunci yang sama dalam melakukan enkripsi dan dekripsi [38]. Diperlukan manajemen kunci yang baik agar kunci enkripsi dan dekripsi hanya dapat dimiliki oleh pihak-pihak yang berwenang. Dalam melakukan proteksi data menggunakan enkripsi kolom, MySQL memiliki *built-in functions* seperti ENCRYPT() dan DECRYPT() yang dapat digunakan untuk melakukan enkripsi kolom untuk *database* MySQL [38]. ENCRYPT() adalah *built-in function* yang berfungsi untuk melakukan enkripsi data sebelum data disimpan ke dalam kolom, sedangkan DECRYPT() adalah *built-in function* yang berfungsi untuk melakukan dekripsi data ketika ada keperluan untuk menampilkan data atau ada keperluan untuk memproses data [38]. MySQL juga menyediakan fitur / *tools* keamanan yang bisa digunakan untuk melakukan enkripsi kolom seperti MySQL TDE (Transparent Data Encryption) yang dapat digunakan di dalam MySQL Enterprise Edition untuk melindungi data [39]. Dengan menggunakan enkripsi kolom, data-data sensitif yang tersimpan akan dapat terjaga kerahasiaan dan integritasnya, bahkan ketika pihak yang tidak memiliki wewenang berhasil mendapatkan akses ke dalam *database*, mereka tidak akan dapat memahami atau membaca data-data sensitif pada kolom yang sudah dienkripsi.

2.1.6 Enkripsi Tabel

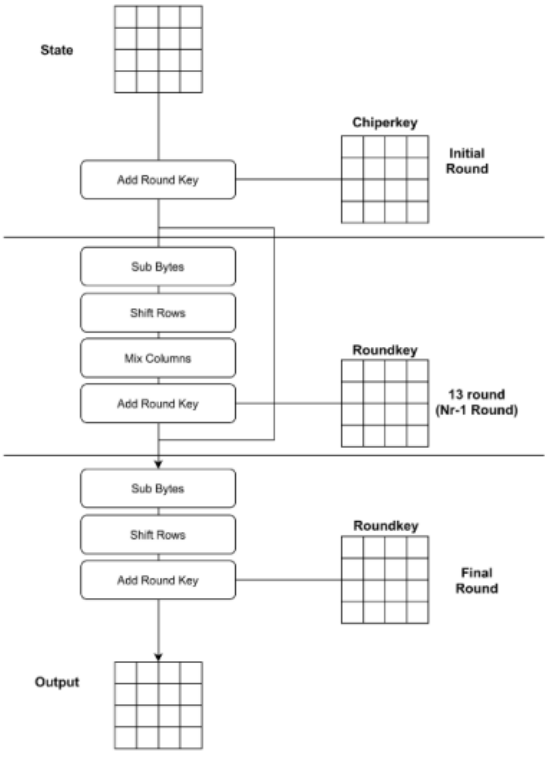
Enkripsi tabel adalah salah satu metode enkripsi yang bisa digunakan untuk memproteksi data dan informasi di dalam sebuah *database*. Dalam metode ini, data-data yang disimpan dalam tabel pada suatu *database* akan dienkripsi secara menyeluruh [40]. Dalam metode ini, data sensitif yang tersimpan di dalam tabel sebuah *database* akan dienkripsi sehingga tidak dapat dibaca oleh pihak yang tidak memiliki hak akses. Enkripsi menggunakan metode ini dilakukan kepada tabel tertentu secara menyeluruh yang diinginkan tanpa harus melakukan enkripsi untuk keseluruhan *database* [40, 41]. Dalam konteks *database* MySQL, banyak fitur yang yang disediakan oleh MySQL untuk melakukan implementasi enkripsi tabel [39, 40]. Salah satu fitur / *tool* yang disediakan oleh MySQL untuk melakukan enkripsi tabel adalah MySQL TDE (Transparent Data Encryption), yang bisa digunakan dalam MySQL Enterprise Edition [39]. Dengan menggunakan MySQL TDE, data-data yang ada di tabel di dalam *database* akan dienkripsi secara otomatis saat data-data tersebut disimpan ke dalam *database*, dengan menggunakan algoritma AES yang digunakan untuk memastikan keamanan data yang dienkripsi [39]. Penggunaan enkripsi tabel untuk *database* MySQL dapat memberikan banyak keuntungan. Proteksi data menggunakan enkripsi tabel dapat melindungi data-data sensitif yang dari kebocoran data atau pencurian data. Dengan menggunakan enkripsi tabel, ketika ada pihak tidak bertanggung jawab berhasil mengakses *database* dan mencuri data dari *database*, pihak-pihak tersebut tidak akan dapat membaca dan memahami data yang telah mereka curi, karena tabel yang menyimpan data-data tersebut sudah dienkripsi [40, 41]. Implementasi enkripsi tabel juga dapat membantu organisasi atau perusahaan untuk memenuhi regulasi privasi dan kerahasiaan data. Dengan memastikan keamanan data yang mereka miliki melalui enkripsi, organisasi dapat menjamin keamanan data mereka [42].

2.1.7 Enkripsi *Database*

Enkripsi *database* adalah salah satu metode enkripsi yang bisa digunakan untuk melindungi data-data sensitif yang disimpan di dalam *database*. Dalam metode ini, semua elemen yang ada di dalam *database*, seperti kolom, tabel, dan data-data yang tersimpan di dalamnya, diproteksi dengan sebuah algoritma enkripsi [43]. Dalam enkripsi tingkat *database*, *table-space* atau *database* yang digunakan akan dienkripsi secara keseluruhan, kemudian hanya akan didekripsi dan dibaca oleh pengguna yang memiliki hak akses [43, 44]. Penggunaan enkripsi dalam tingkat *database* akan dapat memberikan proteksi ekstra untuk *database* terhadap potensi-potensi ancaman keamanan data seperti kebocoran data atau pencurian data [44]. Dengan melakukan enkripsi pada tingkat *database* ini, bahkan ketika *database* mengalami kebocoran data atau pencurian data, data-data yang bocor atau dicuri tersebut tidak akan dapat dibaca dan dipahami oleh pihak yang tidak sah karena semua data telah dienkripsi [43, 44]. Salah satu algoritma enkripsi yang sering digunakan untuk melakukan enkripsi *database* adalah algoritma AES (Advanced Encryption Standard), yang dapat memberikan perlindungan dan proteksi yang efektif dan efisien untuk *database,* yang dapat memberikan tingkat keamanan yang tinggi dan telah terbukti karena sudah banyak digunakan dan diadopsi dalam industri teknologi informasi dan komunikasi [45]. Semua data yang disimpan di dalam *database* dalam bentuk *backup file* juga akan dienkripsi, dengan begitu keamanan data dapat dijamin untuk semua proses yang melibatkan *database* [44]. Implementasi enkripsi dalam tingkat *database* untuk *database* MySQL dapat menggunakan fitur / *tools security* yang dimiliki oleh MySQL, misalnya MySQL TDE yang sudah menggunakan algoritma AES-256 sebagai algoritma enkripsi yang digunakan untuk melakukan proteksi *database* [39, 43, 46, 47].

2.1.8 Enkripsi AES

Advanced Encryption Standard (AES) adalah salah satu metode enkripsi yang mengubah data atau informasi menjadi sebuah kode dalam 4 langkah, yaitu langkah nonlinear (*SubBytes*), langkah disperi (*ShitRows*), langkah difusi (*MixColumns*), dan penambahan kunci (*AddRoundKey*) [48]. Pada awal proses enkripsi dengan algoritma ini, *input* yang telah dimasukkan dan disalin ke dalam *state* akan menjalankan tahap *AddRoundKey*, yaitu penambahan kunci. Setelah itu, *state* akan menjalankan 4 langkah yang telah disebutkan sebelumnya secara berulang-ulang sebanyak N. Proses pengulangan pada algoritma AES ini dikenal dengan *round function*. Pada putaran terakhir, semua langkah akan dilakukan, kecuali langkah *MixCollumns* [49]. Gambar berikut menunjukkan proses dari algoritma AES.



Gambar 2.1 Proses Enkripsi AES
Sumber: [49]

Enkripsi dengan menggunakan algoritma AES merupakan algoritma simetris, dimana kunci yang digunakan untuk proses enkripsi dan dekripsi adalah kunci yang sama [50]. Algoritma AES ini merupakan salah satu algoritma *chipper* yang banyak digunakan untuk melakukan proteksi data atau informasi, terutama untuk data-data atau informasi-informasi yang bersifat rahasia. Algoritma AES pertama kali dipublikasikan oleh National Institute of Standard and Technology (NIST) di tahun 2001 yang waktu itu digunakan sebagai pengganti dari algoritma *Data Encryption Standard* (DES) yang dirasa sudah tertinggal zaman dan memiliki banyak kelemahan. Dalam algoritma AES, *input* dan *output* yang akan diproses adalah sebuah urutan data yang disebut sebagai blok data / *plaintext* yang nantinya akan dienkripsi *ciphertext*. Panjang kunci pada algoritma ini terdiri dari panjang kunci 128 bit, 192 bit, dan 256 bit. Panjang kunci inilah yang akan menentukan jumlah putaran pada algoritma ini [51].

2.1.9 Benchmarking

Benchmarking adalah sebuah metode yang biasa digunakan untuk melakukan pengukuran kinerja dan kemampuan sebuah sistem. Dalam konteks *database*, *benchmarking* dilakukan untuk mengukur kinerja dan kemampuan dari *database* dalam mengelola dan menyimpan data [52]. Tujuan dari *benchmarking* untuk *database* digunakan untuk mencari informasi tentang kecepatan dan efisiensi dari suatu sistem *database* dalam menangani *workloads* yang diberikan [53]. *Benchmarking* biasanya dilakukan dengan membuat serangkaian tes atau skenario yang diberikan kepada sistem yang diuji. Pembuatan *workload* yang mewakili skenario penggunaan nyata *database*, seperti membaca *database*, memasukkan data ke *database*, atau melakukan *query* pencarian data [54]. *Benchmarking* yang dilakukan kemudian akan menampilkan beberapa indeks seperti waktu respon, latensi, penggunaan sumber daya sistem, dan lain-lain yang dapat diukur [63, 64]. *Benchmarking* terbagi ke dalam beberapa

jenis, diantaranya *benchmarking* fungsional dan *benchmarking* kinerja. *Benchmarking* fungsional berfokus kepada pengujian terhadap fungsionalitas dari sistem yang diuji [62, 63]. Sementara itu, *benchmarking* kinerja berfokus kepada pengujian terhadap performa *database* dalam menangani *workload* yang diberikan dari segi kecepatan dan kinerja [62, 63]. *Benchmarking*, terutama untuk *database*, memiliki peran penting dalam melakukan pengembangan dan pemeliharaan sistem *database*, karena *benchmarking* dapat membantu organisasi atau perusahaan membuat keputusan yang lebih baik dalam pemilihan infrastruktur *hardware* dan *software* yang digunakan untuk sistem *database*, juga dalam melakukan identifikasi hal apa saja yang bisa dilakukan untuk meningkatkan performa *database* [63, 64]. Salah satu *tools* yang bisa digunakan untuk melakukan *benchmarking* untuk *database* adalah JMeter.

2.2 ISO (*International Organization for Standardization*)

2.2.1 ISO 9001

ISO 9001 adalah sebuah standar internasional yang membahas dan mengatur mengenai *quality management system* suatu organisasi atau perusahaan [55]. ISO 9001 adalah sebuah standar yang dirancang secara khusus untuk membantu sebuah organisasi atau perusahaan dalam meningkatkan kualitas dari produk atau layanan yang mereka tawarkan kepada para konsumennya, yang bisa meningkatkan tingkat kepuasan pelanggan mereka [59, 60]. ISO 9001 memberikan sebuah struktur dan kerangka kerja yang bisa membantu sebuah organisasi atau perusahaan untuk mendapatkan kinerja bisnis yang lebih baik, sehingga bisa mendapatkan keunggulan operasional, dengan memiliki prinsip-prinsip yang baik mengenai *quality management* [55]. Sertifikasi ISO 9001 bisa didapatkan oleh organisasi atau perusahaan dengan memenuhi persyaratan dasar yang telah ditetapkan, terkait dengan merancang, implementasi, dan pemeliharaan sistem *quality management* yang mereka miliki [56]. Standar dari ISO 9001

membahas sistem *quality management* dari elemen-elemen penting yang ada dalam sebuah perusahaan, seperti komitmen dari pemimpin organisasi atau perusahaan, keterlibatan dari para pekerja / karyawan, dan *risk management* [59, 60]. Implementasi ISO 9001 dapat membantu organisasi atau perusahaan untuk membangun fondasi yang kuat, dalam sistem *quality management* yang mereka miliki menjadi lebih efektif dan efisien [56]. ISO 9001 membantu perusahaan dalam memastikan bahwa produk atau layanan yang mereka sediakan memenuhi harapan dari para konsumen / pelanggannya [56]. Dengan menggunakan pendekatan yang berdasarkan proses, ISO 9001 akan memungkinkan organisasi atau perusahaan untuk melakukan identifikasi, pengelolaan, dan optimalisasi proses-proses yang *critical* dalam meningkatkan kualitas dan efisiensi operasional mereka [59, 60].

2.2.2 ISO 27001

ISO 27001 adalah standar internasional yang membahas dan mengatur mengenai pengelolaan dari keamanan informasi suatu organisasi atau perusahaan [57]. Standar ISO 27001, dirancang secara khusus untuk membantu sebuah organisasi atau perusahaan untuk melakukan identifikasi, pencegahan, dan pengurangan risiko keamanan yang berkaitan dengan informasi atau aset-aset yang dimiliki oleh organisasi atau perusahaan [57]. Standar ISO 27001 yang diimplementasikan ke sebuah organisasi atau perusahaan memberikan struktur dan kerangka kerja yang sudah disusun untuk membantu organisasi atau perusahaan untuk memiliki pengelolaan keamanan informasi yang efektif dan efisien [57]. Standar ISO 27001 didasarkan pada tiga prinsip keamanan informasi yang dikenal dengan CIA Triad, yaitu c*onfidentiality, integrity,* dan *availability* [57]. Dari ketiga prinsip tersebut, ISO 27001 mendorong organisasi atau perusahaan untuk bisa melakukan identifikasi risiko dan mengurangi risiko yang dapat mempengaruhi kerahasiaan, integritas, dan

ketersediaan data dan informasi yang dimiliki oleh organisasi atau perusahaan. Standar ISO 27001 memastikan bahwa organisasi atau perusahaan dapat menjaga integritas data dan informasi yang mereka miliki, dimana data tidak akan dapat dirusak, diubah, atau dimanipulasi tanpa adanya proses autentikasi dan otorisasi [57]. ISO 27001 juga menjadi sebuah pedoman bagi organisasi atau perusahaan untuk melindungi data dan informasi dari akses yang tidak sah untuk menjaga kerahasiaan. ISO 27001 memberikan langkah-langkah yang bisa diikuti untuk memastikan bahwa informasi yang sensitif tetap terjaga kerahasiaannya [57]. Terakhir, ISO 27001 juga memastikan bahwa organisasi atau perusahaan memiliki ketersediaan data atau informasi yang baik. Standar ISO 27001 mendorong organisasi atau perusahaan untuk melakukan implementasi *backup* dan *recovery*, sehingga informasi tetap tersedia dalam situasi yang tak terduga [57].

2.3 *Tools* yang Digunakan

2.3.1 MySQL

MySQL adalah salah satu *database management system* (DBMS) yang dijalankan dengan *structured query language* (SQL) yang banyak digunakan oleh pengembang sistem dalam mengembangkan sistem. MySQL berada pada *rank* 2 *Database Management Systems DB-Engines Ranking* berdasarkan popularitas, hanya kalah dari Oracle [52, 53]. MySQL juga termasuk sebagai *Relational Database Management System* (RDBMS) yang proses pengambilan datanya menggunakan metode *relational database* [58]. MySQL memiliki banyak fitur, seperti fitur keamanan, integritas data, *backup and restore*, bahkan replikasi *database*. Dalam pengembangan sistem, MySQL biasa digunakan sebagai *database* untuk membangun aplikasi web, *desktop*, dan sistem lainnya. MySQL juga dapat diintegrasikan dengan berbagai bahasa pemrograman seperti Python, PHP, Java, dll [60].

2.3.2 MySQL Enterprise Edition

MySQL *Enterprise Edition* (EE) merupakan versi komersial dari MySQL yang merupakan bagian dari Oracle Corporation. MySQL EE menyediakan bermacam-macam fitur tambahan yang tidak tersedia di versi *community edition*, seperti fitur manajemen, keamanan, replikasi, dan dukungan teknis dari Oracle. Dalam konteks dukungan teknis, MySQL EE memberikan layanan dukungan premium dengan memberikan akses ke tim dukungan MySQL yang terlatih dan berpengalaman. Layanan ini memastikan bahwa pengguna MySQL EE selalu memiliki akses ke dukungan teknis yang dapat diandalkan [46]. Dari penjelasan tersebut, MySQL EE dapat diartikan sebagai versi komersial dari MySQL yang memberikan banyak fitur tambahan untuk kepentingan bisnis yang lebih kompleks dibandingkan dengan versi gratisnya. Beberapa contoh fitur yang ditawarkan dalam MySQL EE adalah MySQL *Enterprise Backup,* MySQL *Enterprise Monitor*, MySQL *Enterprise Security*, MySQL *Transparent Data Encryption*, MySQL *Enterprise Audit*, dan MySQL *Enterprise Scalability*. MySQL *Enterprise Backup* dan MySQL *Enterprise Monitor* memberikan solusi untuk organisasi bisa melakukan *backup database* dan *monitoring database* yang lebih handal. MySQL *Enterprise Security* dan MySQL *Enterprise Audit* membantu untuk meningkatkan tingkat keamanan dari *database* dalam melindungi data dari ancaman keamanan dan membantu untuk melacak akses dan aktivitas pengguna. Adapun MySQL *Enterprise Scalability* dapat membantu meningkatkan kinerja aplikasi dengan skalabilitas yang baik [47]. Di sisi lain, fitur yang cukup menonjol adalah MySQL *Transparent Data Encryption* (TDE) yang menyediakan mekanisme enkripsi data secara transparan untuk melindungi data yang disimpan di dalam *database* MySQL. Fitur MySQL TDE ini menggunakan algoritma AES yang sudah terbukti kuat dan aman untuk melakukan proses enkripsi data [39].

2.3.3 MySQL Transparent Data Encryption

MySQL Transparent Data Encryption (TDE) merupakan salah satu fitur keamanan yang disediakan di dalam MySQL *Enterprise Edition* (EE), yang memungkinkan pengguna untuk melakukan enkripsi data yang ada di dalam *database* secara transparan. MySQL TDE ini menggunakan algoritma *Advanced Encryption Standard* (AES) yang sudah terbukti kuat dan aman untuk melakukan enkripsi data [39]. MySQL TDE memungkinkan pengguna MySQL EE untuk meningkatkan keamanan data yang mereka miliki, karena data yang disimpan di dalam *database* dienkripsi menggunakan algoritma AES. MySQL TDE juga memungkinkan pengguna untuk melakukan enkripsi terhadap kolom atau tabel di dalam *database* secara individual, sehingga pengguna dapat memilih data apa saja yang ingin dienkripsi, selain itu, MySQL TDE juga mendukung integrasi dengan MySQL *Enterprise Backup* yang memastikan *backup* data juga terenkripsi dengan aman [61]. MySQL TDE memanfaatkan *caching database* sehingga dapat mencapai kinerja tinggi [39]. MySQL TDE menjadi solusi yang tepat untuk suatu perusahaan dapat meningkatkan keamanan data di dalam *database* MySQL. Dengan menggunakan MySQL TDE, pengguna dapat memastikan bahwa data-data yang ada di dalam *database* aman dari ancaman kebocoran ataupun pencurian data [62]. MySQL TDE menawarkan beberapa keuntungan yang berkaitan dengan keamanan dan proteksi data. Proteksi data yang diberikan MySQL TDE memberikan perlindungan data yang tersimpan di dalam *database* dari akses yang tidak sah dari pihak internal maupun eksternal. Dengan enkripsi yang baik, data yang dicuri atau diakses secara ilegal oleh pihak-pihak yang tidak bertanggung jawab tidak akan dapat dibaca dan digunakan. MySQL TDE juga memungkinkan pengguna untuk memilih tabel dan kolom apa saja yang mau dienkripsi, sehingga kinerja sistem dapat terjaga dengan baik [61].

2.3.4 MySQL InnoDB Cluster

MySQL InnoDB Cluster adalah salah satu fitur yang disediakan oleh MySQL untuk *hight availability* sebuah *database*. MySQL InnoDB Cluster menjadi sebuah pilihan yang dapat diandalkan untuk melakukan manajemen replikasi (*clustering*) dalam sebuah *database environment* [63]. Arsitektur yang mendukung MySQL InnoDB Cluster adalah MySQL Router, MySQL Shell, dan Group Replication [63]. MySQL InnoDB Cluster memiliki sebuah kemampuan untuk membuat *clusters* replikasi yang terdiri beberapa anggota, di mana anggota *cluster* dapat saling berkomunikasi dan melakukan pertukaran data secara *real-time*, sehingga replikasi data yang dilakukan dalam *cluster* menjadi lebih cepat dan konsisten [63]. Dengan kemampuan ini, ketika ada satu anggota yang mengalami kerusakan atau mengalami gangguan, anggota lain akan dapat mengambil alih dan melanjutkan operasional sistem untuk menjaga ketersediaan *database* [63]. MySQL InnoDB Cluster juga menyediakan sebuah sistem pendeteksi dan penanganan konflik yang canggih. Saat terjadi sebuah konflik dalam proses replikasi data, sistem ini akan secara otomatis mendeteksi dan menyelesaikan konflik tersebut, dengan tujuan untuk mengurangi intervensi dari manusia yang memiliki potensi melakukan kesalahan [63]. Penggunaan dan implementasi MySQL InnoDB Cluster pada *database* organisasi atau perusahaan dapat meningkatkan ketersediaan (*availability*) dan keandalan sistem dari *database* MySQL yang digunakan. Dengan kemampuan untuk melakukan *clustering multi-master* dan manajemen konflik otomatis, MySQL InnoDB Cluster dapat memastikan bahwa layanan *database* akan dapat tetap tersedia untuk digunakan walaupun terjadi gangguan pada salah satu atau pada beberapa anggota dalam *cluster* [63]. Secara umum, MySQL InnoDB Cluster menjadi solusi yang baik untuk memastikan *high availability* dari suatu *database* [63].

2.3.5 JMeter

JMeter adalah sebuah *software* yang bisa digunakan sebagai *tools* untuk melakukan pengujian atau *benchmarking* pada sistem berbasis web. *Software* ini dikembangkan oleh Apache Software Foundation, di mana *software* ini menyediakan sebuah *environment* yang dapat digunakan untuk melakukan *benchmarking* kepada sistem, termasuk *database* [64]. Salah satu fitur penting yang dimiliki oleh JMeter adalah kemampuan dari JMeter untuk melakukan simulasi *workload* dengan mengirimkan HTTP *request* kepada sistem yang ingin diuji [64]. Dengan adanya kemampuan ini, pengguna dapat melakukan pengujian untuk mengetahui seberapa baik sistem yang diuji dapat menangani *workload* yang diberikan, dan bagaimana kinerja dari sistem yang diuji mengalami perubahan ketika ada perubahan yang dilakukan pada sistem, misalnya enkripsi *database*. JMeter tidak hanya mendukung protokol HTTP, tetapi juga mendukung protokol-protokol lainnya seperti FTP, JDBC, LDAP, TCP, dan banyak lagi [64]. Dengan dukungan ini, pengguna dapat melakukan pengujian terhadap berbagai jenis aplikasi dan sistem, seperti aplikasi *website*, aplikasi *desktop*, bahkan server *database*. Bukan hanya itu, JMeter juga menyediakan fitur pengujian lainnya, seperti pengaturan waktu respons, pengujian *workload*, pengujian keandalan, pengujian skalabilitas, dan masih banyak lagi. Dengan fitur-fitur yang disediakan, pengguna dapat melakukan pengujian dan memahami bagaimana sistem yang diuji berperilaku di dalam kondisi dan keadaan yang berbeda-beda [64]. Sebagai *software* yang populer digunakan untuk melakukan *benchmarking*, JMeter telah digunakan secara luas oleh *tester* di seluruh dunia, di mana dukungan yang diberikan oleh komunitas-komunitas yang menggunakan JMeter dan banyaknya dokumentasi-dokumentasi yang dimiliki oleh JMeter membuat JMeter menjadi pilihan yang tepat untuk melakukan *benchmarking* [64].

2.4 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

1	Judul	Keamanan Dokumen Menggunakan Algoritma *Advanced Encryption Standard* (AES) [65]
	Penulis	Joko Handoyo, Yulieo Muchti Subakti
	Tahun	2018
	Latar Belakang	Penggunaan algoritma enkripsi yang baik diperlukan untuk menjaga kerahasiaan dokumen penting.
	Metode	Implementasi enkripsi AES pada dokumen.
	Tools	AES, Java
	Hasil	Algoritma AES dapat diterapkan dengan efektif pada dokumen, dan dapat mengamankan dokumen dengan baik.
2	Judul	*Evaluation of DES and AES Cryptographic Algorithms* [66]
	Penulis	Navneet Kaur, N Venkat Sai, G Manikanta Kumar
	Tahun	2019
	Latar Belakang	Evaluasi perbandingan antara algoritma DES dan AES dalam aspek keamanan data.

	Metode	Analisis perbandingan kinerja dan keamanan DES dan AES.
	Tools	MATLAB
	Hasil	AES lebih aman dan efektif dalam melindungi data daripada DES.
3	Judul	*A Cloud Database based on AES 256 GCM Encryption Through Devolving Web application of Accounting Information System* [67]
	Penulis	Alameen Eltoum Mohamed Abdalrahman
	Tahun	2021
	Latar Belakang	Keamanan data dan privasi menjadi perhatian penting dalam sistem informasi akuntansi berbasis web.
	Metode	Pengembangan aplikasi web dengan enkripsi AES 256 GCM.
	Tools	AES 256 GCM, PHP, MySQL
	Hasil	Sistem basis data awan dapat diimplementasikan dengan aman dan terenkripsi melalui aplikasi web akuntansi menggunakan AES 256 GCM.
4	Judul	*A Comparative Study of MongoDB and Document-Based MySQL for Big Data Application Data Management* [68]

	Penulis	Cornelia A. Győrödi, Diana V. Dumşe-Burescu, Doina R. Zmaranda and Robert Ş. Győrödi
	Tahun	2020
	Latar Belakang	Manajemen data aplikasi *big data* memerlukan basis data yang dapat mengelola jumlah data yang besar.
	Metode	Studi perbandingan antara MongoDB dan MySQL berbasis dokumen.
	Tools	MongoDB, MySQL
	Hasil	Kedua *database* tersebut cocok untuk aplikasi Big Data yang melibatkan volume data yang besar, serta *database* yang sangat kompleks, dengan waktu respons yang sangat singkat terlepas dari kerumitan kueri dan jumlah data. MySQL berbasis dokumen yang memiliki metode yang mengambil parameter bagian seperti kode dengan cara yang mirip dengan MySQL relasional, dan MongoDB didasarkan pada format BSON. Dengan demikian MySQL berbasis dokumen jauh lebih mudah digunakan, terutama dalam hubungannya dengan *database* relasional.

5	Judul	*System of End-To-End Symmetric Database Encryption* [69]
	Penulis	V V Galushka, A R Aydinyan, O L Tsvetkova, V A Fathi and D V Fathi
	Tahun	2018
	Latar Belakang	Keamanan data pada *database* merupakan hal yang sangat penting, terutama pada data yang bersifat rahasia atau penting. Teknik enkripsi bisa digunakan untuk melakukan proteksi data.
	Metode	Implementasi teknik enkripsi *end-to-end* simetris untuk melindungi *database* dari kebocoran informasi.
	Tools	Microsoft SQL Server
	Hasil	Teknik enkripsi *end-to-end* simetris dapat digunakan untuk melindungi *database* dari kebocoran informasi. Namun, implementasi teknik ini memerlukan manajemen kunci yang baik dan dapat mempengaruhi kinerja *database*. Adapun enkripsi dapat digunakan untuk melindungi data dalam *database* dan membahas tantangan dalam menerapkan teknik enkripsi pada *database*. Implementasi teknik enkripsi juga mempengaruhi kinerja *database.*

6	Judul	*Performance Testing on Transparent Data Encryption for SQL Server's Reliability and Efficiency* [17]
	Penulis	Evaristus Didik Madyatmadja, Aditya Nur Hakim, & David Jumpa Malem Sembiring
	Tahun	2021
	Latar Belakang	Keamanan data menjadi salah satu aspek yang paling penting untuk difokuskan pada sistem. Namun, menggunakan fitur keamanan untuk meningkatkan keamanan data mungkin mempengaruhi kinerja sistem.
	Metode	*Performance testing* yang dibagi menjadi *Load Testing*, *Stress Testing*, dan *Backup Testing*
	Tools	HammerDB, SQL Server
	Hasil	Untuk Efisiensi, Transparent Data Encryption dapat menurunkan kinerja sistem hingga 15% dari CPU, Memori, dan Durasi Pencadangan. Namun, manfaat menggunakan Enkripsi Data Transparan untuk pengukuran keamanan dianggap berguna dan menambahkan lapisan keamanan lain untuk data sistem.

7	Judul	*Transparent Data Encryption: Comparative Analysis and Performance Evaluation of Oracle Databases* [70]
	Penulis	Natarajan K, Vaheedbasha Shaik
	Tahun	2020
	Latar Belakang	Transparent Data Encryption (TDE) bisa memberikan manfaat yang sangat besar untuk *Database* Relasional di aspek Keamanan Data, Enkripsi Kriptografi, dan Kepatuhan
	Metode	Pendekatan infrastruktur dan analisis pengumpulan hasil
	Tools	VM Server, Oracle Cloud
	Hasil	TDE adalah salah satu fitur terbaik dalam keamanan data pada *database* Penelitian membuktikan bahwa fitur TDE telah mengalami peningkatan yang signifikan dari versi Oracle yang lama ke versi baru. Pada rilis terbaru, Oracle mengklaim adanya optimisasi CPU dan penyimpanan, dan hasilnya telah terbukti. Namun, tidak ada detail yang signifikan dan spesifik tentang kinerja komponen IO (*Input/Output*) dan RAM (*Random Access Memory*).

8	Judul	Kajian Literatur Terstruktur terhadap Kebocoran Data Pribadi dan Regulasi Perlindungan Data Pribadi [71]
	Penulis	T. Rahmat Kautsar
	Tahun	2022
	Latar Belakang	Dalam era yang berkembang pesat seperti sekarang, akses terhadap berbagai hal menjadi lebih mudah melalui internet. Dengan kebebasan yang ada saat ini dan dukungan dari perangkat elektronik seperti komputer, telepon pintar, serta koneksi jaringan yang baik, orang dapat dengan mudah mengakses berbagai informasi termasuk data pribadi orang lain.
	Metode	Kajian Literatur
	Hasil	Kasus kebocoran data pribadi di Indonesia terjadi akibat 3 faktor yaitu : Standar Operasional Prosedur (SOP), Sumber Daya Manusia (SDM), dan Teknologi. Adapun dalam semua proses pengelolaan data pribadi, termasuk perolehan, pengolahan, penyimpanan, transfer, dan penghapusan data, dapat dipastikan bahwa teknologi digunakan.

Tabel penelitian terdahulu tersebut membahas topik keamanan dokumen dengan algoritma AES, perbandingan dari algoritma AES dan DES, *cloud database* dengan enkripsi AES-256, perbandingan MongoDB dan MySQL, enkripsi *end-to-end symmetric*, *testing* terhadap performa dari Transparent Data Encryption untuk SQL Server, evaluasi performa Transparent Data Encryption untuk Oracle, dan kajian mengenai faktor yang menyebabkan terjadinya kebocoran data [17], [65-71]. Berbagai literatur yang termuat pada tabel tersebut telah menghasilkan penemuan-penemuan penting dalam mengatasi masalah keamanan dan manajemen data [17], [65-71]. Beberapa literatur membahas mengenai penggunaan algoritma AES untuk melakukan proteksi data dengan menggunakan teknik enkripsi [65-67]. Dari literatur-literatur yang membahas mengenai penggunaan algoritma AES, menunjukkan bagaimana algoritma AES dapat diterapkan dalam teknik enkripsi untuk secara efektif dan efisien memproteksi data di dalam *database* [65-67]. Dilakukan juga perbandingan dengan algoritma DES yang menunjukkan bahwa algoritma AES memiliki keunggulan dalam efektivitas dan efisiensi dibandingkan dengan algoritma DES [66]. Selain itu, ada juga literatur yang melakukan perbandingan antara *database* MongoDB dan MySQL berbasis dokumen di dalam manajemen data aplikasi Big Data [68]. Hasil studi dalam literatur ini menunjukkan bahwa kedua *database* tersebut cocok untuk aplikasi Big Data, tetapi MySQL berbasis dokumen jauh lebih mudah digunakan, terutama dalam hubungannya dengan *database* relasional [68]. Adapun penelitian lainnya membahas mengenai *end-to-end symmetric encryption* [69]. Dalam literatur ini, ditemukan bahwa implementasi teknik enkripsi memerlukan manajemen kunci yang baik dan dapat mempengaruhi kinerja *database* [69]. Literatur lain yang dilakukan membahas mengenai Transparent Data Encryption (TDE) [17], [70]. Dimana dalam literatur-literatur ini dilakukan evaluasi terhadap performa TDE, di mana TDE dapat menurunkan kinerja sistem, tetapi di lain sisi dapat memberikan proteksi yang baik [17], [70]. Terakhir, salah satu literatur membahas mengenai faktor-faktor yang menjadi penyebab terjadinya kebocoran data [71]. Ditemukan bahwa ada 3 faktor yang menjadi penyebab kebocoran data, yaitu: standar operasional prosdur, sumber daya manusia, dan teknologi [71].

Secara keseluruhan, literatur-literatur tersebut memberikan wawasan penting dalam masalah keamanan dan manajemen data pada sistem informasi, dan memberikan solusi-solusi yang dapat diterapkan secara efektif dalam menjaga keamanan data dan pengaruhnya terhadap kinerja sistem secara keseluruhan [17], [65-71]. Dari literatur-literatur tersebut dapat diambil beberapa hal yang dapat menjadi pembaharuan atau inspirasi untuk penelitian ini. Pertama, terdapat solusi-solusi dari teknologi enkripsi yang dinilai efektif dalam memproteksi data pada *database* untuk menjaga data-data dan informasi-informasi penting sebuah organisasi. Dari penelitian dalam literatur-literatur tersebut, dapat diketahui bahwa algoritma AES terbukti efektif dan aman untuk digunakan dalam melakukan enkripsi *database* [65-67]. Dimana algoritma AES juga terbukti lebih baik dibandingkan dengan algoritma DES [66]. Dengan hasil tersebut, dalam penelitian ini akan dilakukan penelitian proteksi data dengan menggunakan *tool* yang menerapkan algoritma AES. Dalam penelitian ini akan dilakukan proteksi data di dalam *database* MySQL. Berdasarkan penelitian yang pernah dilakukan, MySQL memiliki *usability* yang baik [68]. Dengan *usability* yang baik ini, MySQL cocok untuk digunakan, terutama untuk *database* relasional [68]. Di penelitian lain, teknik enkripsi *end-to-end* simetris dapat digunakan untuk melindungi *database* dari kebocoran informasi. Namun, implementasi teknik enkripsi memerlukan sebuah manajemen kunci yang baik. Adapun implementasi dari teknik enkripsi dapat mempengaruhi kinerja *database* [69]. Oleh karena itu, di penelitian ini akan digunakan *tool* yang memiliki manajemen kunci yang baik dalam melakukan enkripsi untuk memproteksi data dalam *database*. Di dalam penelitian ini dilakukan proteksi data yang bertujuan untuk melindungi *database* dari ancaman keamanan seperti kebocoran data, dan dari literatur yang ada ditemukan bahwa teknologi adalah salah satu faktor dari kemungkinan bocornya data [71]. Penelitian ini akan berfokus untuk melindungi data dari kebocoran data dengan pemanfaatan teknologi.