e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

Literatur Review Permasalahan Pengamanan pada *Database*

Harry Darmawan Siregar, Fajar Subkhi Sulaiman, Noor Falih, S.Kom., M.T

Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
email:harry@upnvj.ac.id, fajarss@upnvj.ac.id, falih@upnvj.ac.id
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia

Abstrak. *Database* atau basis data merupakan media yang digunakan oleh setiap orang yang ingin memindahkan dan meletakkan data dari cara konvensional menggunakan kertas ke arah yang lebih digital melalui komputer. Kelebihan dari basis data adalah kemudahan untuk melakukan pencarian data karena data yang ada pada *database* dikelompokan membentuk tabel-tabel dan kolom-kolom. Namun, data yang di-*input*-kan pada *database* bersifat apa adanya, sehingga data yang di-*input*-kan akan sesuai dengan data yang disimpan serta dikeluarkan *database* yang artinya jika terdapat data penting seperti nama, *password*, dsb. Maka data tersebut akan dapat dibaca oleh semua pihak, baik yang memiliki wewenang dan pihak yang tidak berwenang. Penelitian dilakukan dengan studi literatur yang me-*review* 17 jurnal mengenai pengamanan *database* dengan berbagai algoritma agar data yang ada tidak dapat dimengerti oleh pihak tidak berwenang. Penelitian ini menghasilkan sebuah daftar dari algoritma yang terbukti dapat mengamankan data pada *database* serta poin umum dan beberapa kekurangan dari algoritma yang digunakan.

Kata Kunci: *Database*, kriptografi, keamanan *database*

1 Pendahuluan

Di masa yang serba modern saat ini, seluruh data dan informasi yang ada dikumpulan dalam satu-kesatuan di dalam media penyimpanan yang digital. Media tersebut biasa dikenal dengan *database* atau basis data. Basis data adalah media penyimpanan digital yang memisahkan data menjadi tabel-tabel yang di dalamnya dikelompokan lagi menjadi kolom-kolom yang mana memudahkan proses *indexing* serta mengurangi penggunaan kertas. Namun sayangnya, dibalik kenyamanan yang disediakan oleh *database* juga terdapat berbagai kendala, salah satunya adalah *input* data pada *database* bersifat apa adanya, artinya jika terdapat informasi penting seperti nama, alamat, nomor telepon, serta *password*, maka data tersebut dapat dibaca dengan jelas. Hal ini bukanlah sebuah masalah jika memang yang mendapatkan informasi merupakan pihak yang memiliki wewenang untuk mengetahui informasi tersebut, tapi bagaimana jika yang mengetahui informasi tersebut bukanlah pihak berwenang melainkan pihak lain yang ingin menggunakan informasi yang ada untuk kegiatan yang dapat dikatakan ilegal, tentunya perlu diperhatikan lagi bagaimana cara data dimasukkan dan diambil ke dan dari *database*.

Berdasarkan data dari *Analytic Data Advertising* (ADA), kegiatan *e-commerce* yang memanfaatkan penggunaan *database* melonjak pesat hingga 400% sejak Maret 2020 dikarenakan adanya pandemi yang melanda seluruh dunia. Disisi lain, Bank Indonesia (BI) mencatat bahwa transaksi yang dilakukan melalui *e-commerce* mencapai 98.3 juta transaksi yang terjadi pada bulan Maret 2020 yang merupakan sebuah peningkatan sebanyak 18,1% jika dibandingkan dengan bulan Februari 2020. Pada 1 Mei 2020, sebanyak 91 juta data yang diinformasikan merupakan data pengguna Tokopedia yang mana data tersebut diperjual belikan seharga US$5.000 pada forum *hacker*. Tidak lama kemudian,pada 6 Mei 2020, terjadi peretasan dan pembobolan data pada *database*, dimana sebanyak 12,9 juta data pengguna Bukalapak bocor di tangan peretas dan diperjualbelikan [1].

Oleh karena melonjaknya penggunaan *database* serta terjadinya kegiatan pembobolan *database* yang terjadi, maka perlu adanya sebuah metode agar informasi dalam *database* dapat dikatakan aman sehingga informasi yang ada hanya dapat diketahui oleh pihak yang memiliki wewenang untuk mengetahui informasi tersebut. Adapun metode mengenai mengamankan data pada *database* serta algoritma apa yang akan digunakan akan dibahas berdasarkan *literatur review* dari penelitian-penelitian yang memiliki topik pengamanan *database* yang sudah dilakukan sebelumnya.

e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

2 Metode

Penelitian ini menggunakan metode studi literatur atau *literature review* mengenai metode pengamanan data pada *database* yang dapat keamanan dalam *database* yang dibatasi dengan penggunaan teknik algoritma kriptografi dengan algoritma yang berbeda-beda. Adapun literatur yang digunakan berasal dari jurnal prosiding SKANIKA, Jurnal Media Infotama, dan Jurnal E-Komtek (Elektro-Komputer-Teknik), International Journal of Engineering Technology Research & Management(Ijetrm), ProQuest, Science Direct, serta Academia.edu dengan kata kunci yang digunakan antara lain "Implementasi Kriptografi pada *Database*", "Peningkatan Keamanan pada *Database* dengan Kriptografi", "*Database Encryption*", "*Securing Database*", dan *"Database Security"*. Dari hasil pencarian dikumpulkan sebanyak 17 judul jurnal yang cocok dengan kata kunci yang digunakan. Dilakukan analisis pada setiap jurnal

3 Hasil

Penggunaan *database* sudah sangat banyak diterapkan karena perpindahan data dari media penyimpanan konvensional dengan kertas ke media penyimpanan digital dengan *database*. Pada penyimpanan *database,* data disimpan secara digital. Data pada *database* memiliki informasi penting yang tidak boleh diketahui oleh orang yang tidak berwenang. *Database* memiliki ancaman pada data yang ada pada *database,* yaitu apabila data tersebut disalahgunakan untuk tindakan ilegal. Dalam pengamanan *database,* terdapat berbagai macam serangan yang dapat mengancam keamanan isi *database* oleh karenannya pengamanan *database* diperlukan. Adapun ancaman terhadap data antara lain pencurian data, hilangnya integritas dan keaslian data, hilangnya ketersediaan data pada *database* [2].

Ancaman yang ada pada *database* biasanya berupa tindakan pembobolan *database*, dimana pihak tidak berwenang melakukan kegiatan pembobolan ini untuk mengetahui informasi dan data-data yang terdapat di dalam *database* yang nantinya data dan informasi yang didapatkan dapat disalahgunakan. Adapun target dari pembobolan *database* biasanya adalah perusahaan, organisasi, dan juga *e-commerce*.

Data-data yang dimiliki pihak tersebut merupakan data yang sangat penting, dimana jika terjadi kebocoran, data-data tersebut dapat disalahgunakan dan dimanfaatkan untuk perbuatan ilegal. Sebagai contoh menurut artikel berita [3] pada tahun 2020 Mei beberapa *e-commerce* mengalami kebocoran data seperti Tokopedia dan Bukalapak. Begitu juga menurut artikel [4] Tokopedia mengalami kebocoran data sebanyak 91 juta data pada 1 Mei yang memberikan dampak kerugian US$5.000 yang dijual pada sebuah situs *hacker*. Kemudian tidak lama kemudian pada 6 Mei Bukalapak juga mengalami kebocoran hingga 12,9 juta data. Ancaman yang menyerang dari Tokopedia menargetkan data pengguna Tokopedia. Menurut artikel [5] data pengguna Tokopedia yang berhasil dicuri merupakan data-data seperti *password*, *username*, nama, alamat, nomor telepon, dan tanggal lahir. Data-data seperti nama, *username*, alamat merupakan data sekunder yang tidak diberikan perlindungan pada *database*, maka dari itu pengamanan pada *database* sangat penting dan diperlukan untuk mencegah terjadinya kebocoran data.

Terdapat beberapa solusi yang ditawarkan untuk mengamankan *database*, salah satu solusinya yaitu dengan menggunakan algoritma kriptografi. Algoritma kriptografi akan mengamankan data pada *database* dengan cara melakukan proses enkripsi terhadap data-data yang akan dimasukkan kedalam *database* kedalam bentuk kombinasi kode, simbol, angka, dan huruf sehingga data pada *database* berubah menjadi data tidak bermakna yang biasa disebut cipherteks yang nantinya data tidak bermakna ini akan menggantikan data asli di dalam tabel pada *database* [6]. Data yang tidak memiliki makna atau cipherteks tersebut, ketika bocor dan tersebar luas, tidak dapat dimengerti maknanya oleh pihak tidak berwenang sehingga data asli menjadi aman dan tidak bisa disalahgunakan ketika data-data tersebut bocor.

Dengan adanya implementasi pengamanan pada *database*, maka seluruh bagian dan kegiatan pada organisasi dan perusahaan akan dapat berjalan dengan efektif karena data dan informasi yang penting terlindungi dari serangan pihak yang tidak berwenang dan pihak tidak berwenang tersebut tidak dapat mengetahui makna dari isi *database*-nya dan tidak dapat menyalahgunakan [7].

e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

Berdasarkan jurnal yang telah di-*review* pengamanan data pada *database* dapat diamankan dengan implementasi metode enkripsi algoritma kriptografi. Berikut daftar metode implementasi algoritma kriptografi yang dapat digunakan. Berikut daftar metode implementasi algoritma kriptografi dapat dilihat pada Tabel 1.

Tabel 1 Daftar Metode Implementasi Algoritma Kriptografi

#	Nama Metode Algoritma Kriptografi	Jurnal
1	RIVEST CODE 4 (RC4)	[8-11]
2	Vigenere Cipher	[6], [8], [12]
3	AES	[9], [12-14]
4	MD5	[15]
5	RSA	[6], [16]
6	Blowfish	[13], [17]
7	TRIANGLE CHAIN CIPHER (TCC)	[18-19]
8	Caesar Cipher	[11]
9	Elgamal	[20-21]
10	*Honey Encryption*	[22]

Dalam menganalisis solusi yang dapat diterapkan untuk permasalahan ini digunakan beberapa jurnal yang berjumlah 17 jurnal yang ditemukan dengan kata kunci "Implementasi Kriptografi pada *Database*", "Peningkatan Keamanan pada *Database* dengan Kriptografi", dan *"Database Security"*., untuk mengamankan data pada *database* terdapat berbagai macam implementasi algoritma kriptografi yang dapat digunakan untuk mengamankan *database*. Dalam implementasi algoritma kriptografi untuk meningkatkan keamanan data pada *database* dari literatur jurnal yang di-*review* terdapat dua jenis penerapan dalam mengimplementasikannya yaitu berbasis *desktop* dan berbasis *website.* Berikut daftar penerapan algoritma berbasikan *desktop* dan *website* dapat dilihat pada Tabel 2.

Tabel 2 Daftar Basis Penerapan Algoritma

#	Basis Penerapan	Digunakan di jurnal
1	*Desktop*	[6-9], [11-15], [16], [18-20]
2	*Website*	[10], [15], [17]

Dalam melakukan enkripsi data pada *database* berdasrkan jurnal artikel terdapat dua cara untuk melakukan enkripsi datanya yaitu melakukan enkripsi pada tabel *database* dan satu per satu pada isi *record.* Berikut daftar cara melakukan enkripsi pada data dapat dilihat pada Tabel 3.

Tabel 3 Cara Melakukan Enkripsi Terhadap Tabel

#	Cara Melakukan Enkripsi terhadap data	Digunakan di jurnal

1	Enkripsi per Tabel	[8], [10], [12], [16]. [18-21]
2	Enkripsi per *Record*	[6], [9], [11], [14-15]

Dari algoritma kriptografi yang diimplementasikan pada jurnal yang di-*review* dapat dibagi jenis kunci algoritmanya yaitu terdiri dari algoritma kriptografi simetris, algoritma kriptografi asimetris, dan *Hash function*. Kriptografi asimetris merupakan algoritma kriptografi yang berkerja dengan mensubsitusi plainteksnya dengan kuncinya yang mensubsitusi *binary* pada plainteksnya. [9]. Keamanan algoritma simetris ini tergantung pada kuncinya, semakin kuat kuncinya, maka akan semakin sulit dipecahkan namun apabila kuncinya ketahuan, maka cipherteks dapat dipecahkan. [13]. Algoritma simetris disebut juga algoritma kunci privat dimana hanya menggunakan satu kunci saja untuk melakukan enkripsi dan dekripsi dan tidak boleh bocor kuncinya [9]. Kemudian algoritma kriptografi asimetris merupakan jenis algoritma yang disebut dengan algoritma kunci publik, dimana algoritma tersebut menggunakan dua kunci berbeda untuk melakukan enkripsi dan dekripsi. Kunci tersebut adalah kunci publik untuk melakukan enkripsi dan kunci privat untuk melakukan dekripsi [9] [13]. Kunci publik boleh disebarkan dan tidak masalah jika diketahui orang lain, tetapi kunci privat perlu dirahasiakan karena kunci privat digunakan untuk enkripsi [13]. *Hash function* merupakan sebuah fungsi yang menghasilkan nilai untuk membuktikan integritas keaslian dari data aslinya dengan membandingkan nilai hash tersebut[13] [15]. Berikut daftar jenis algoritma yang diimplementasikan pada jurnal dapat dilihat pada Tabel 4.

Tabel 4 Jenis Kriptografi

#	Jenis Kriptogtrafi	Jurnal
1	Algortima Kriptografi Simetris	[7-15], [18-19]
2	Algoritma Kriptografi Asimetris	[6] [16], [20-21]
3	*Hash Function*	[15]

Dalam hasil proses enkripsi data yang sudah menjadi cipherteks akan mengalami kenaikan jumlah ukuran *size.* Peningkatan *size* data yang terenkripsi tergantung pada seberapa besarnya tabel pada *database* dimana semakin besar dan kompleks tabel, maka akan semakin besar *size* hasil enkripsi dan juga semakin lama waktu proses enkripsi [8] [19]. Namun pada implementasi dengan algoritma Elgamal yang merupakan algoritma asimetris hasil cipherteks enkripsi dan hasil dekripsi yang kembali menjadi plainteks *size* dari datanya sama besar namu waktu yang digunakan untuk proses enkripsi tersebut memakan waktu cukup lama[20]. Algoritma kriptografi simetris yang melakukan banyak proses iterasi subsitusi juga dapat mempengaruhi *size* dari cipherteks hasil enkripsi contohnya pada algoritma AES yang melakukan 9 iterasi dan algoritma Blowfish melakukan proses subsitusi iterasi sebanyak 16 kali pada jaringan *feistel* [7][9].

Algoritma asimetris merupakan algoritma kriptografi yang memiliki proses perhitungan yang sangat rumit dan bilangannya besar sehingga akan sangat sulit untuk dipecahkan oleh para kriptanalis [6] [21]. Algoritma kriptografi simetris juga dapat menghasilkan panjang cipherteks yang lebih panjang. Pada jurnal [6] yang merupakan implementasi dari algoritma RSA yang merupakan algoritma kriptografi asimetris, karakter hasil dari proses enkripsinya akan lebih panjang sebesar 38 % dari plainteksnya [6]. Pada algoritma kriptografi simetris penggunaan sumber daya komputer seperti *processor,* RAM, dsb. menggunakan lebih sedikit sumber daya dibandingkan dengan algoritma kriptografi asimetris. Hal tersebut dikarenakan algoritma kriptografi asimetris melakukan komputasi matematis yang lebih besar dibandingkan dengan algoritma kriptografi simetris [22].

Algoritma kriptografi simetris memiliki proses untuk enkripsi yang lebih simpel dibandingkan dengan algoritma kriptografi asimetris. Algoritma simetris merupakan algoritma yang menggunakan satu kunci saja sehingga untuk proses enkripsi hanya memerlukan masukan satu kunci saja seperti pada algoritma Blowfish dan juga ada yang tidak perlu memerlukan masukan tambahan seperti algoritma RC4 [8-11].

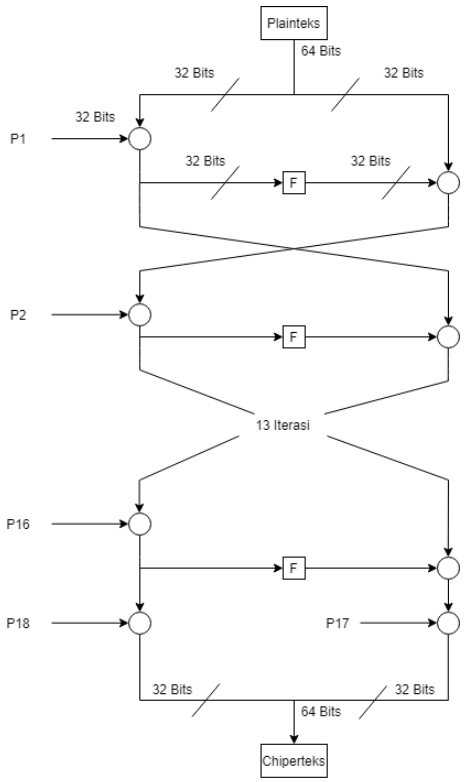
Pengamanan data pada *database* juga bisa dilakukan pada lingkungan *cloud*. Pada layanan *cloud* untuk mengamankan data yang akan disimpan pada *database cloud* dapat diimplementasikan algoritma *Honey*

Encryption. Algoritma *Honey Encryption* akan mengamankan *database* dengan cara melakukan proses otentifikasi berdasarkan *password* yang dimasukan. *Password* akan diotentifikasi dan jika *password* salah saat melakukan enkripsi, maka algoritma ini akan menhasilkan cipherteks acak yang palsu namun seakan-akan berupa cipherteks asli. Kemudian jika ingin melihat data asli yaitu melakukan dekripsi, maka algoritma tersebut akan menghasilkan plainteks acak dan sebuah *database* palsu yang akan diberikan kepada orang yang tidak berwenang tersebut yang mencoba menyerang *database*. Orang yang tidak berwenang tersebut akan mengira bahwa dia telah berhasil mendapatkan informasi asli namun yang sebenarnya terjadi adalah plainteks dan *database* yang diarahkan kepada penyerang adalah informasi palsu. Algoritma ini juga bisa digunakan untuk mengatasi serangan seperti *brute force attack* karena penyerang tidak akan bisa tahu apakah *password* yang dicoba melalui *brute force attack* apakah benar atau tidak. Setiap mencoba *password* dengan *brute force attack,* maka akan deberikan hasil plainteks dan *database* yang seolah informasi yang asli. Ketika otentifikasi berhasil, maka akan lanjut kepada proses selanjutnya, enkripsi akan dilanjutkan dengan algoritma AES dan disimpan kedalam *database* yang asli.[22]

4 Pembahasan

Pada era modern saat ini data-data penting milik perusahaan dan organisasi sudah disimpan secara digital pada *database*. Data-data penting pada *database* milik perushaan masih banyak yang belum dilakukan pengamanan , tetapi masih hanya data asli yang dimasukan kedalam *database*. Data-data yang belum diamankan tersebut sangat rentan dari serangan-serangan yang mengacam integritas, pencurian data-data penting, dan ketersediaan data [2]. Serangan seperti *brute force attack* dapat mengancam keamanan yang berbasis kepada *password*, maka dari itu untuk mengatasi serangan-serangan yang mengancam data-data penting pada *database* terdapat banyak solusi untuk mengamankan data dengan implementasi algoritma kriptografi.

Algoritma kriptografi terdiri dari tiga jenis yaitu algoritma kriptografi simetris, algoritma kriptografi asimetris, dan *hash function*. Algoritma kriptografi simetris untuk proses enkripsi dan dekripsi hanya menggunakan satu kunci saja namun kuncinya tersebut menjadi hal penting yang menentukan keamanan dari algoritma tersebut. Algoritma Blowfish menggunakan sub kunci yang akan dipermutasi dengan mengacak bit blok pada kunci yang selanjutnya akan digunakan untuk proses enkripsi[13] [17]. Kemudian pada proses enkripsinya algoritma kriptografi simetris akan melakukan subsitusi plainteksnya dengan kuncinya. Proses subsitusi tersebut akan melakukan perulangan iterasi yang cukup banyak seperti pada algoritma AES yang melakukan sembilan kali putaran dan algoritma Blowfish yang melakukan enam belas kali iterasi subsitusi kunci dan *binary* plainteks hingga hasil subsitusi akan menghasilkan cipherteksnya [9] [13] [17]. Berikut gambar ilustrasi jaringan *feistel* algoritma Blowfish dapat dilihat pada Gambar 1.



Gambar 1 Jaringan *Feistel* pada Algoritma Blowfish yang memiliki 16 iterasi pada setiap kunci.

Proses subsitusi kunci dengan *binary* plainteks menjelaskan bagaimana pentingnya kunci pada algoritma kriptografi simetris. Jika kunci tersebut berhasil diketahui maka kriptanalis dan pihak penyerang yang tidak berwenang akan dapat memecahkan keamanan dari algoritma kriptografi simetris. Seperti pada algortima Vigenere Cipher dimana merupakan algoritma kriptografi klasik yang merupakan pengembangan dari Caesar cipher. Jika panjang kuncinya sudah ketahuan dengan cara metode kasiski maka plainteks dapat dipecahkan, Dari hasil cipherteksnya terdapat beberapa potongan kata yang berulang-ulang sehingga polanya dapat diketahui [12]. Kunci pada algoritma *Triangle Chain Cipher* (TCC) memiliki kunci yang berjumlah sebanyak plainteksnya. Dari kunci tersebut plainteks akan dienkripsi dengan digeser pada setiap karakter berdasarkan kunci yang bernilai integer. Kemudian akan dilakukan proses pergeseran lagi dimana mengeser hasil dari yang sebelumnya[19]. Algoritma kriptografi simetris seperti RC4 tidak memerlukan masukan tambahan untuk proses enkripsinya [8]-[11].

Sedangkan pada algoritma kriptografi asimetris untuk proses enkripsi dan dekripsi diperlukan dua kunci yaitu kunci publik dan kunci privat. Kunci publik digunakan untuk melakukan enkripsi dan kunci privat untuk melakukan dekripsi. Pada algoritma kriptografi asimetris pada tahap sebelum melakukan enkripsi terdapat tahap pembangkitan kunci yaitu tahap untuk menentukan kunci publik dan kunci privat [16]. Pada proses pembangkitan kunci ini bilangan prima akan tentukan secara acak dan kuatnya keamanan pada algoritma RSA bergantung pada bilangan acak yang digunakan. Jika bilangan prima acaknya kecil, maka keamanan dapat dipecahkan lebih mudah [16]. Pada algotrima Elgamal bilangan prima acak yang ditentukan pada proses pembangkitan kunci akan digunakan untuk melakukan perhitungan logaritma diskrit. Angka bilangan prima akan membuat perhitungan akan menjadi sangat besar sehingga untuk memecahkannya akan memerlukan waktu yang cukup lama [21]. Setelah mendapatkan kunci plainteks, masukan akan dipotong tiap karakter dimana banyak karakter diberi variabel m dan banyak karakter pesan i. Kemudian konversi pesan m menjadi ASCII kemudian tiap karakter m yang sudah menjadi ASCII akan di-*input* nilai K dan hitung tiap karakter m dengan rumus di bawah ini.

$$\text{gamma} = g^k \bmod p \,. \quad (1)$$

$$\text{delta} = y^k \bmod p \,. \quad (2)$$

e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

Setelah hitung gamma dan delta pada tiap karakter ASCII m sebanyak i kali dan hasil gamma dan delta digabungkan dan diurutkan sehingga jadilah cipherteks. Karena dari konveris ASCII maka hasil cipherteksnya dapat menjadi lebih panjang daripada data asli plainteksnya [21].

Kemudian pada algoritma RSA untuk proses enkripsi setelah membangkitkan kunci yaitu e dan n untuk proses enkripsi RSA kemudian plainteks yang menjadi masukan akan diubah ke ASCII dalam bentuk desimal m lalu dibandingkan dengan plainteks dalam bentuk desimal tersebut dengan n ≥ m kemudian lakukan operasi perhitungan dengan rumus di bawah ini.

$$c = m^e \bmod(n) . \quad (3)$$

Sehingga dari operasi perhitungan tersebut menghasilkan cipherteks. Peningkatan jumlah karakter pada cipherteks dapat bertambah sebesar 38 % dari jumlah aslinya [6] [16].

Kemudian pada algoritma simetris yaitu algoritma Blowfish merupakan algoritma simetris yang cukup populer yang telah dilakukan analisis sejak tahun 1994. Hasil dari analisis yang dilakukan menyatakan bahwa algoritma Blowfish merupakan algoritma yang kompeten dimana pengujian dilakukan pada komputer yang memiliki setidaknya 32-bit *microprocessor* keatas dan *cache* yang besar dan belum ada serangan yang bisa menghancurkan Blowfish. Blowfish juga memiliki desain yang cepat untuk penerapannya dan bisa dijalankan pada memori yang kecil setidaknya 5 KB dan memiliki panjang kunci yang beragam bisa dari 32-bit untuk minimalnya hingga maksimalnya 448-bit [17]. Algoritma kriptografi simetris lainnya yaitu algoritma AES untuk kecepatan proses enkripsi dan dekripsi bergantung pada spesifikasi *resources* atau sumber daya pada komputer. Semakin bagus spesifikasi komputer maka semakin cepat prosesnya dan begitu juga pada algoritma RC4 [10][14]. Namun jika dibandingkan dengan algoritma kriptografi asimetris seperti algoritma RSA contohnya maka algoritma RSA memerlukan *resources* yang lebih besar dalam melakukan proses enkripsi dan dekripsi dan penggunaan energi dan efisensi pada hardware Algoritma RSA kurang baik dibandingkan dengan algoritma kriptorafi simetris tersebut [22].

Kemudian solusi algoritma kriptografi tersebut dapat diterapkan pada aplikasi berbasis *desktop* dan *website.* Penerapan implementasi algoritma kriptografi berbasis *desktop* memiliki kekurangan yaitu untuk menggunakannya tidak bisa digunakan dimana saja. Pengguna perlu melakukan instalasi aplikasi dahulu dan jika aplikasi hanya bisa terinstall pada komputer tertentu maka akan membatasi akses untuk menggunakan aplikasi tersebut [9][18][19]. Sedangkan dengan menerapkan implementasi algoritma kriptografi dengan berbasis *website*, algoritma kriptografi bisa digunakan dan diakses kapan saja dan dimana saja melalui aplikasi *browser* tergantung pada *hosting website*-nya [10][15][17]. Untuk mempermudah penggunaan aplikasi baik *desktop* maupun *website* diterapkan dilengkapi dengan *user interface* yang dapat membantu navigasi penggunaan aplikasi. *User interface* dapat memberikan layanan berupa tampilan yang dapat memudahkan proses pengoperasian antara manusia*(brainware)* dan aplikasi *(software).* Dengan adanya layanan tampilan user interface, maka proses pengamanan pada proses pertukaran informasi saat penyimpanan data dapat menjadi lebih mudah dan nyaman[16]. Pada *desktop* biasanya dibangun pada bahasa pemrograman Visual Basic. Net dan Java sedangkan *website* dibangun dengan bahasa pemrograman Java, PHP, dan Javascript [1][3-8] [10-15].

Dalam melakukan proses enkripsi, terdapat dua objek yang menjadi target enkripsi data, yaitu tabel dan tiap *record.* Enkripsi data pada *database* dengan melakukan enkripsi terhadap tabel akan memiliki waktu yang cukup lama berdsarkan ukuran *size* data pada tabel, semakin besar *size* dan semakin lama proses yang dilakukan [8]. Waktu proses enkripsi juga berpengaruh pada jumlah *field* pada tabel dan banyaknya *record* pada tabel [12-13][19-20]. Sedangkan dengan melakukan enkripsi pada tiap *record* mungkin waktu enkripsi akan lebih cepat dibandingkan dengan enkripsi pada tiap tabel, tetapi dengan melakukan enkripsi satu persatu pada setiap *records* pada tabel maka waktunya akan lebih lama juga [11].

Pengamanan pada *database* selain dapat diterapkan pada *database* lokal juga bisa diterapkan pada lingkungan atau *environment cloud.* Algoritma AES dan RSA dapat meningkatkan keamanan data pada *database desktop* dengan baik [6][13][14], namun kedua algoritma tersebut tidak efisien pada data yang berada di *environment* dengan scalability tinggi layaknya *database* pada *cloud* [17]. Adapun teknik enkripsi yang dikatakan efisien untuk data pada *database cloud* adalah teknik yang Homomorphic seperti SHE [22-23]. Penerapan algoritma Diffie-Hellman juga kurang baik jika implementasikan untuk mengamankan data pada *database* dalam *environment cloud.* Karena Diffie-Hellman memiliki kelemahan pada otentifikasi kunci rahasianya dan rentan terhadap serangan *man in the*

middle attack. Diffie-Hellman juga memiliki kekurangan pada saat pembangkitan penentuan bilangan primanya yang memerlukan komputasi yang berat [23]. Algoritma yang tepat untuk diimplementasikan pada *environment cloud* adalah *Homomorphic Encryption* karena algoritma tersebut dapat memanipulasi langsung data yang dienkripsi dan didekripsi, contohnya seperti algoritma *Honey Encryption* [22-23]. Algoritma *Honey Encryption* dapat melakukan otentifikasi melalui *password* yang dimasukan dan jika *password*-nya tidak sesuai atau salah maka algoritma tersebut akan memberikan hasil baik cipherteks enkripsi maupun plainteks derkipsi. Hasil cipherteks dan plainteks tersebut bukanlah data yang asli melainkan hasil dari algoritma *Honey Encryption* yang menghasilkan plainteks dan cipherteks acak sehingga seolah-olah cipherteks dan plainteks tersebut adalah asli [22].

Dengan melakukan implementasi algoritma-algoritma kriptografi tersebut untuk melakukan pengamanan *database,* maka data pada *database* diharapkan dapat memenuhi beberapa kriteria keamanan yaitu *Confidentiality*, *Integrity*, *Non-Repudiation*, *Authentication*, dan *Access Control* [15]. *Confidentiality* merupakan kriteria yang paling penting yaitu kerahasiaan data. Pada proses implementasi algoritma kriptografi data pada *database* akan diamankan dengan proses enkripsi sehingga kerahasiaan dari makna asli dari *database* tersebut tidak dapat diketahui oleh orang yang tidak berwenang. Kemudian integritas dimana data pada *database* yang sudah dienkripsi pada *database* yaitu berbentuk cipherteks akan dapat dikembalikan menjadi data aslinya yang maknanya dapat diketahui Kembali. Lalu *Non-Repudiation* atau penghindaran penolakan dimana data yang diamankan ketika dikembalikan menjadi data asli dengan dekripsi, memerlukan kunci yang digunakan pada saat proses enkripsi untuk mengembalikan data menjadi data asli. Hal ini dilakukan untuk membuktikan bahwa data tersebut asli dan tidak diubah. Dengan hal tersebut makan keaslian suatu data akan terjamin. Dengan implementasi algoritma kriptografi pada *database* tersebut makan setiap data pada *database* dapat dikontrol *entity* pada setiap aksesnya [15].

Pada implementasi algoritma tersebut tentu memiliki kelebihan yang dapat melindungi data di dalam *database* tetapi juga memiliki beberapa kekurangan. Algoritma kriptografi merupakan proses matematis yang menggunakan kemampuan komputasi matematis yang sangat tinggi untuk mengamankan dengan mengkodekan data asli menjadi cipherteks sandi sehingga manusia yang berusaha memecahkannya akan kesulitan [8]. Beberapa algoritma yang memiliki keamanan yang lebih tinggi seperti algoritma kriptografi asimetris, memiliki waktu untuk melakukan proses enkripsi dan dekripsi lebih lama, ini dikarenakan besarnya logaritma diskrit pada perhitungan algoritma dan juga tergantung pada besar dan banyaknya data pada *database*[20]. Waktu proses juga bergantung pada struktur *database* dan tabelnya, waktu dalam melakukan proses enkripsi dan dekripsi akan berbanding lurus dengan ukuran tabel pada *database* yang akan diproses. Semakin kecil dan simpel sebuah *database* yang diproses, maka semakin cepat proses enkripsi dan dekripsi. Semakin besar dan kompleks sebuah struktur *database* yang diproses, maka proses enkripsi dan dekripsi akan lebih lama [16]. Kemudian dengan penerapan untuk *database* yang sudah diterapkan pada *desktop* dan *website* juga dapat memberikan kemudahan untuk melihat hasil rekap data pada *database* [16]. Kemudian untuk ukuran *file*-nya juga dapat bergantung pada implementasi algoritma yang digunakan dimana pada implementasi AES-128 ukurang *file* hasil dari proses enkripsi akan lebih besar dikarenakan algoritma AES-128 pada saat proses enkripsi melakukan operasi sebanyak sembilan kali putaran [10]. Kemudian untuk implementasi dari algoritma kriptografi tersebut untuk mengamankan *database* penerapan pada *desktop* maupun *website* akan memerlukan *resources* komputer dimana dapat mempengaruhi kecepatan dalam melakukan proses pengamanan yakni enkripsi dan dekripsi.

5 Kesimpulan

Database merupakan hal yang sangat penting terutama pada sebuah perusahaan, organisasi, dan instasi lainnya seperti *e-commerce*. Data pada *database* sangat penting pada sebuah instansi karena jika datanya bocor dan disalahgunakan, maka dapat mengakibatkan masalah pada pihak pemilik data tersebut. Oleh karena itu pengamanan pada *database* diperlukan dengan implementasi algoritma kriptografi. Pengamanan *database* dengan implementasi algoritma kriptografi dapat mengamankan data pada *database* dengan mengubah data asli pada *database* menjadi penyandian cipherteks, sehingga makna dari data asli tidak dapat diketahui maknanya apabila terjadi serangan yang membocorkan *database* oleh pihak tidak berwenang.

Algoritma kriptografi yang dapat diimplementasikan ada tiga jenis yaitu algoritma kriptografi simetris, algoritma kriptografi asimetris, dan *hash function*. Algoritma kriptografi simetris seperti Blowfish, AES, RC4, TCC, Vigener Cipher, dan Caesar Cipher menggunakan satu kunci saja untuk melakukan proses enkripsi dan dekripsi. Tingkat

keamanan dari algoritma kriptografi simetris ini terletak pada kekuatan kunci. Jika kuncinya ketahuan maka keamanan dapat dipecahkan. Algoritma kriptografi simetris memilki kecepatan proses enkripsi dan dekripsi lebih cepat dibandingkan algoritma kriptografi asimetris dan juga memiliki penggunaan sumber daya atau *resources* yang lebih kecil. Sedangkan algoritma kriptografi asimetris seperti Elgamal dan RSA memiliki keamanan yang lebih baik karena menggunakan dua kunci yang berbeda yaitu kunci publik untuk enkripsi dan kunci publik untuk dekripsi. Algoritma kriptografi asimetris melakukan perhitungan komputasi matematika yang besar sehingga sangat sulit untuk dipecahkan namun dapat mengakibatkan penggunaan *resources* dan daya yang lebih tinggi. *Hash function* seperti MD5 digunakan untuk melakukan pengecekan integritas dan keaslian data pada *database*.

Lama waktu dari proses enkripsi juga berpengaruh pada besar dan kompleksitas dari tabel dan banyaknya isi data *record* pada *database*. Semakin kecil ukuran tabel *database* yang diproses, semakin cepat waktu proses enkripsi dan dekripsi. Dan sebaliknya jika semakin besar ukuran tabel *database* maka semakin lama waktu proses enkripsi dan dekripsi.Penerapan implementasi dapat diterapkan pada *desktop* dan *website*. *Website* lebih fleksibel karena bisa diakses melalui *browser* dibandingkan *desktop* harus melakukan instalasi pada komputer dahulu. Untuk melakukan proses enkripsi data pada *database* bisa melakukan pada tabel dan pada *record*. Enkripsi pada tiap *record* lebih cepat dibandingkan pada tabel namun jika melakukan enkripsi satu per satu pada tiap *record* waktu yang digunakan lebih lama dibandingkan dengan enkripsi langsung pada tabel.

Pengamanan data pada *database* pada *environment cloud* lebih baik menggunakan algoritma *Honey Encryption.* Algoritma *Honey Encryption* merupakan algoritma *Homomorphic Encryption* yang dapat langsung memanipulasi data yang dienkripsi dan didekripsi dimana dapat merespon langsung pada masukan yang salah pada sebuah serangan dimana algoritma akan memberikan cipherteks dan plainteks acak untuk mengelabuhi pihak tidak berwenang.

Referensi

[1] Nafi'ah, R. 2020. Pelanggaran Data Dan Pencurian Identitas Pada E-Commerce. *CyberSecurity dan Forensik Digital*. 3(1). p. 8. Available at: http://ejournal.uin-suka.ac.id/saintek/cybersecurity/article/download/1980/1732.
[2] Hamdani, Trio. 2020. Heboh Data Tokopedia dan Bukalapak Dibobol, e-Commerce Diserang?. Diakses pada 30 Maret 2021. https://finance.detik.com/berita-ekonomi-bisnis/d-5004198/heboh-data-tokopedia-dan-bukalapak-dibobol-e-commerce-diserang
[3] Paul, P.K, Aithal, P. S. ,"*Database* Security : An Overview and Analysis of Current Trend". International Journal of Management, Technology, and Social Sciences (IJMTS), 4(2), 53- 58. ISSN: 2581-6012, 2019
[4] Mahrus, Zuhri. 2020. OPINI : Kebocoran Data Pengguna Tokopedia, Bukalapak, dan Bhinneka: Siapa Peduli?, https://cyberthreat.id/. Available at: https://cyberthreat.id/read/6795/Kebocoran-Data-Pengguna-Tokopedia-Bukalapak-dan-Bhinneka-Siapa-Peduli (Accessed: 30 Maret 2021)
[5] Rahman, Arif. 2020. Data Sekunder Tokopedia yang Bocor Bisa Datangkan Phishing Massal. Diakses pada 30 Maret 2021 https://cyberthreat.id/read/6503/Data-Sekunder-Tokopedia-yang-Bocor-Bisa-Datangkan-Phishing-Massal
[6] A. Rahman et al., "Implementasi Keamanan *Database* menggunakan Algoritma Vigenere Cipher dan Rivest Shamir Adleman (RSA) Berbasis Desktop," Skanika, vol. 1, no. 2, pp. 801–806, 2018.
[7] Sadli, Muhammad, Painem. 2018. Implementasi Pengamanan *Database* Menggunakan Metode Blowfish Dan Aes Pada Perusahaan. 1(1). pp. 366–372.
[8] Surbakti, J.S., Subandi, "Aplikasi Pengamanan *Database* Keuangan Berbasis Dekstop Menggunakan Algoritma RC4 Dan Vigenere Cipher," vol. 1, no. 1, pp. 237–242, 2018.
[9] M. D. Wulandari et al., "Aplikasi Pengamanan *Database* Berbasis Desktop dengan Algoritma AES-128 dan Rivest Code (RC4)," Skanika, vol. 1, no. 1, pp. 373–379, 2018.
[10] A. R. Wahid and Syafrullah, "Implementasi Algoritma Rc4 Dan Kompresi Lzw Untuk Pengamanan *Database* Pada Pt . Mpp International," Skanika, vol. 1, no. 3, pp. 1045–1050, 2018..
[11] A. S. Muharyanto and T. Fatimah, "Keamanan *Database* Dengan Metode Rivest Code 4 (RC4) dan Caesar Cipher Berbasis Desktop," Skanika, vol. 1, no. 2, pp. 508–513, 2018, [Online]. Available: http://jom.fti.budiluhur.ac.id/index.php/SKANIKA/article/view/249.
[12] D. E. Erlianto and Painem, "Aplikasi Kriptografi Pengamanan *Database* Menggunakan Metode AES Dan Vigenere Berbasis Desktop Pada Pada Divisi Pencegahan Dan Penanggulangan Hiv Aids Yayasan Kapeta" Skanika, vol. 1, no. 2, pp. 772–779, 2018.
[13] M. Sadli and P. Painem, "Implementasi Pengamanan *Database* Menggunakan Metode Blowfish Dan Aes Pada Perusahaan Cv. Balerong Sakti," Skanika, vol. 1, no. 1, pp. 366–372, 2018, [Online]. Available: http://jom.fti.budiluhur.ac.id/index.php/SKANIKA/article/view/206.
[14] T. Erlangga, D. Kusumaningsih, F. T. Informasi, U. B. Luhur, P. Utara, and K. Lama, "Implementasi Algoritma

Advanced Encryption Standard -128 (Aes-128) Untuk Pengamanan *Database*," vol. 1, no. 2, pp. 565–569, 2018
[15] Khairil and P. W. Ginta, "Implementasi Pengamanan *Database* menggunakan MD5," J. Media Infotama, vol. 8, no. 1, pp. 29–44, 2012..
[16] L. Pratama and S. Subandi, "Pengamanan Tabel *Database* Menggunakan Kriptografi Algoritma Rsa," Skanika, vol. 1, no. 3, pp. 925–930, 2018, [Online]. Available: http://jom.fti.budiluhur.ac.id/index.php/SKANIKA/article/view/2507.
[17] A. Rifa'i and L. C. Sumartini, "Implementasi Kriptografi Menggunakan Metode Blowfish Dan Base64 Untuk Mengamankan *Database* Informasi Akademik Pada Kampus Akademi Telekomunikasi Bogor Berbasis Web-Based," J. E-Komtek, vol. 3, no. 2, pp. 87–96, 2019, doi: 10.37339/e-komtek.v3i2.133.
[18] D. Kusumaningsih and S. A. Pebresega, "Implementasi Algoritma Kriptografi Triangle Chain Cipher (TCC) untuk Pengamanan *Database* Berbasis Desktop pada CV.Usaha Tani," Skanika, vol. 1, no. 1, pp. 380–384, 2018.
[19] A. Sudrajat et al., "Aplikasi Keamanan*Database*Menggunakan Algoritma Kriptografi Triangle CHAIN Cipher Berbasis Desktop" Skanika, vol. 1, no. 2, 2018.
[20] G. A. Sahputra et al., "Implementasi Kriptografi dengan Mentode Algoritma Elgamal untuk Keamanan *Database* Berbasis Java Desktop pada PT. Makmur Supra Nusantara" Skanika, vol. 1, no. 1, pp. 309–315, 2018.
[21] E. pratama, "Implementasi Algoritma Elgamal dan kode HILL Untuk Keamanan *Database*," 2020, doi: 10.31219/osf.io/vq5yc.
[22] A. N. Efficient, E. Method, F. O. R. Securing, and A. To, "An Efficient, Encryption Method for Securing Access to C;oud" no. 03, pp. 1–7.2019
[23] I. Wankhede and S. Sirsat, "Securing Cloud Data Storage through Encryption," Int. J. Electron. Commun. Soft Comput. Sci. Eng., p. 21, 2015.