BAB II
TINJAUAN PUSTAKA

II.1 Penelitian Terkait

Berdasarkan penelitian yang dilakukan oleh Ema Utami, 2018 dengan judul Penerapan Algoritma *Blowfish* Untuk Membuat Sebuah Model Kriptosistem Algoritma Dan Menganalisis Kinerja Algoritma *Blowfish* Dengan Simulasi Data Terbatas. Berdasarkan penelitian yang dilakukan, Data yang dikirim melalui media transmisi tersebut bisa saja disadap, dirusak atau dicuri oleh para *cracker* dan pihak-pihak yang memiliki kepentingan tertentu. Masalah *cyber crime* ini masih menjadi masalah yang sulit dikendalikan hingga saat ini. Untuk mengatasi masalah keamanan data ini, penulis melakukan pendekatan teknologi enkripsi data menggunakan algoritma *Blowfish*, enkripsi yang termasuk dalam golongan *Symmetric Cryptosystem*.

Berdasarkan penelitian yang dilakukan Harry Darmawan Siregar, 2021 dengan judul *Literatur Review* Permasalahan Pengamanan pada *Database.* 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.

Berdasarkan penelitian yang dilakukan oleh Doli Hasibuan, 2019 dengan judul Implementasi Intrusion Detection System Menggunakan Algoritma Blowfish, dengan studi kasus dilakukan di Dota Café. Penelitian ini diharapkan dapat membantu dalam penanggulangan atau mencegah terjadinya kecurangan/gangguan dari pihak yang tidak bertanggung jawab.

Berdasarkan penelitian yang dilakukan oleh Anton Prafanto, 2016 dengan judul Penerapan Algoritma *Blowfish* Untuk Keamanan SMS Pada Android. Berdasarkan hasil penelitian yang telah dilakukan, Salah satunya adalah algoritma *Blowfish* yang dapat digunakan untuk mengenkripsi sebuah data. *Blowfish* sendiri merupakan algoritma kunci simetris yang memiliki kunci yang sama untuk mengenkripsi dan mendekripsi sebuah data. Algoritma *Blowfish* termasuk kedalam *chipper block* dan sampai saat ini masih dianggap aman karena belum ada *attack* yang benar-benar mematahkan algoritma *Blowfish*. Selanjutnya apa yang akan dibahas pada makalah ini nantinya meliputi bagaimana algoritma *Blowfish* mengenkripsi sebuah SMS dan juga akan membahas bagaimana cara agar algoritma *Blowfish* dapat bekerja dengan optimal.

Berdasarkan penelitian yang dilakukan oleh Harlen Gilbert Simanullang, 2018 dengan judul algoritma *blowfish* untuk meningkatkan keamanan *database* mysql. Berdasarkan hasil penelitian yang telah dilakukan, Algoritma *Blowfish* akan lebih optimal jika digunakan untuk aplikasi yang tidak sering berganti kunci, seperti

jaringan komunikasi atau enkripsi file otomatis. Selain itu, karena algoritma ini membutuhkan memori yang besar, maka algoritma ini tidak dapat diterapkan untuk aplikasi yang memiliki memori kecil seperti *smart card*. Panjang kunci yang digunakan, juga mempengaruhi keamanan penerapan algoritma ini. Dengan menggunakan algoritma *Blowfish*, keamanan informasi dalam *Database* MySQL dapat ditingkatkan.

II.2. Penerimaan Siswa Baru

Pengertian penerimaan siswa baru dapat diartikan sebagai proses pendaftaran utuk menjadi siswa baru pada sekolah lama ke sekolah yang baru dengan beberapa tahap penyeleksian yang dilakukan oleh sekolah baru. Penerimaan siswa baru selalu dilakukan di setiap tahun ajaran baru oleh semua sekolah.

Pada *website* SMK Penerbangan PBD Medan pendaftar akan di tentukan dengan nilai ijazah sekolah dasar 60% dan nilai ujian seleksi 40% dimana kuota akan ditentukan oleh panitia, siswa yang memiliki nilai dengan urutan tertinggi sesuai kuota akan di luluskan.

II.3. Kriptografi

Kriptografi berasal dari dua kata bahasa Yunani, *Cryptos* dan *Graphein*. *Cryptos* berarti *secret* (rahasia) dan *Graphia* berarti *writing* (tulisan). Jadi, kriptografi berarti *secret writing* (tulisan rahasia). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. Kriptografi adalah ilmu dan seni dalam mengamankan pesan dengan

cara mengubah pesan menjadi sesuatu yang tidak dapat dimerngerti oleh orang lain dengan teknik-teknik dan metode-metode tertentu. Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih ke arah teknik-tekniknya. (Harlen Gilbert Simanullang,2018).

Algoritma kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengolah data atau pesan yang akan disampaikan. Ada empat syarat yang perlu dipenuhi dalam ilmu kriptografi (sebagai aspek-aspek keamanan), yaitu:

1. Kerahasiaan (*confidentiality*), adalah menyediakan privasi untuk pesan dan menyimpan data dengan menyembunyikannya.
2. Integritas data (data *integrity*), berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain menyangkut penyisipan, penghapusan, dan pensubtitusian data lain ke dalam data yang sebenarnya, yang tidak dimodifikasi ketika sedang dalam proses transmisi data.
3. Autentikasi (*authentication*), beruhubunagn dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user/entity authentification*) maupun mengidentifikasi kebenaran sumber pesan (data *origin authentification*), agar penerima informasi dapat memastikan keaslian pesan, bahwa pesan itu datang dari orang yang dimintai informasi.

4. Non-repudiasi (*non-repudiation*), yang berarti dapat membuktikan bahwa dokumen memang benar datang dari orang yang dimintai informasi.

Algoritma kriptografi adalah bagian dari kriptografi yang berisi kumpulan langkah-langkah logis yang digunakan untuk melakukan enkripsi dan dekripsi. Biasanya langkah-langkah ini berupa sekumpulan fungsi matematik. Berdasarkan kuncinya, algoritma kriptografi dibedakan menjadi dua, yaitu algoritma simetris (kunci privat) dan algoritma asimetris (kunci publik). Secara umum kriptografi terdiri dari dua proses yaitu enkripsi dan dekripsi. *Plaintext* adalah pesan yang akan dirahasiakan, dinotasikan dengan m (*Message*), yang dapat berupa bit tream, file *text*, *digitized voice stream*, *digital video image* atau lebih singkatnya m adalah data *binary*. (Harlen Gilbert Simanullang,2018)

Enkripsi adalah proses pengaman data atau informasi dengan membuat informasi tersebut seolah tidak bermakna atau tidak dapat dibaca. enkripsi dinotasikan dengan E, berfungsi untuk mengubah m menjadi c, dalam matematika dinotasikan dengan:

$E(m) = c$…….(1)

Keterangan rumus:

E = enkripsi

m = *plaintext*

c = *ciphertext*

Ciphertext adalah hasil dari proses enkripsi, dinotasikan dengan c, juga berupa data *binary* yang kadang-kadang mempunyai ukuran yang sama dengan m, lebih kecil

dari m atau lebih besar dari m. Dekripsi adalah kebalikan dari enkripsi yaitu proses mengubah data menjadi bermakna. Fungsi dekripsi D, berfungsi untuk mengubah c menjadi m, dalam matematika dinotasikan dengan: (Harlen Gilbert Simanullang, 2018):

D(c) = m……(2)

Keterangan rumus:

D = dekripsi

c = *ciphertext*

m = *plaintext*

Crypanalyst adalah orang mempelajari ilmu dan seni ilmu membongkar *ciphertext*. Menurut ISO 7498-2 istilah yang lebih tepat untuk *decryption* adalah decipher. *Cipher* adalah kata lain dari algoritma yang digunakan untuk melakukan melakukan proses kriptografi. *Cipher* juga sering disebut teknik yang digunakan untuk proses enkripsi dan deskripsi. Secara umum berdasarkan kesamaan kuncinya, algoritma kriptografi dibedakan menjadi: (Harlen Gilbert Simanullang,2018):

1. Kunci-simetris / *symetric-key*, kunci yang digunakan untuk melakukan enkripsi sama dengan kunci yang digunakan untuk dekripsi

2. Kunci-asimetris / *asymetric-key*, kunci enkripsi dan kunci dekripsi tidak sama.

Berdasarkan kerahasiaan kuncinya, algoritma kriptografi dibedakan menjadi:

1. Algoritma kriptografi kunci rahasia *secret-key*

2. Algoritma kriptografi kunci publik publik-*key*

II.4. Algoritma *Blowfish*

Algoritma *Blowfish* adalah salah satu algoritma yang dapat digunakan untuk melakukan enkripsi data sehingga data asli hanya dapat dibaca oleh seseorang yang memiliki kunci enkripsi tersebut. *Blowfish* termasuk dalam enkripsi *block Cipher* 64bit dengan panjang kunci yang bervariasi antara 32bit sampai 448bit. Algoritma *Blowfish* terdiri atas dua bagian yaitu Pembangkitan subkunci (*KeyExpansion*) dan Enkripsi Data. Enkripsi Data terdiri dari iterasi fungsi sederhana (*Feistel Network*) sebanyak 16 kali putaran. Semua operasi adalah penambahan (*addition*) dan XOR pada variabel 32bit. Operasi tambahan lainnya hanyalah empat penelusuran *table* (*table lookup*) array ber*indeks* untuk setiap putana. Pada algoritma *Blowfish*, digunakan banyak subkey. Kunci-kunci ini harus dihitung atau dibangkitkan terlebih dahulu sebelum dilakukan enkripsi atau dekripsi data. Pada jaringan feistel, *Blowfish* memiliki 16 iterasi, masukannya adalah 64bit elemen data atau sebut saja "X". (Harlen Gilbert Simanullang,2018)

Proses Enkripsi dijelaskan sebagai berikut:

1. Bentuk inisial *P-array* sebanyak 18 buah (P1,P2,.....P18) masing-msing bernilai 32bit. Array P terdiri dari delapan belas kunci 32bit subkunci : P1,P2,.....,P18
2. Bentuk Sbox sebanyak 4 buah masing-masing bernilai 32bit yang memiliki masukan 255. Empat 32bit Sbox masing-masing mempunyai 255 entri :

   S1,0,S1,1,.....,S1,255

   S2,0,S2,1,.....,S2,255

S3,0,S3,1,.....,S3,255

S4,0,S4,1,.....,S4,255

3. *Plaintext* yang akan dienkripsi diasumsikan sebagai masukan, *Plaintext* tersebut diambil sebanyak 64bit, dan apabila kurang dari 64bit maka kita tambahkan bitnya, supaya dalam operasi nanti sesuai dengan datanya.
4. Hasil pengambilan tadi dibagi 2, 32bit pertama disebut XL, 32bit yang kedua disebut XR.
5. Selanjutnya lakukan operasi XL = XL xor Pi dan XR = F(XL) xor XR.
6. Hasil dari operrasi diatas ditukar XL menjadi XR dan XR menjadi XL.
7. Lakukan sebanyak 16 kali, perulangan yang ke16 lakukan lagi proses penukaran XL dan XR.
8. Pada proses ke17 lakukan operasi untuk XR = XR xor P17 dan XL = XL xor P18.
9. Proses terakhir satukan kembali XL dan XR sehingga menjadi 64bit kembali. Salah satu operator yang paling banyak digunakan pada algoritma ini adalah oerator XOR berikut adalah table kebenarannnya

Tabel II.1 *Table* kebenaran XOR

P	Q	Hasil
F	F	F
F	T	T
T	F	T
T	T	F

Lalu untuk mencari fungsi F adalah sebagai berikut :

Bagi XL, menjadi empat bagian 8bit: a,b,c dan d. F(XL)=((S1,a + S2,b mod 2^{32})xor S3,c) + S4,c mod 2^{32} Subkunci dihitung menggunakan algoritma *Blowfish*, metodenya adalah sebagai berikut. (Ema Utami,2018) :

1. Pertama-tama inilialisasi *P-array* dan kemudian empat Sbox secara berurutan dengan *string* yang tetap. String ini terdiri atas digit hexadesimal dari Pi.
2. XOR P1 dengan 32bit pertama kunci, XOR P2 dengan 32bit kedua dari kunci dan seterusnya untuk setiap bit dari kunci (sampai P18).Ulangi terhadap bit kunci sampai seluruh *P-array* di XOR dengan bit kunci.
3. Enkrip semua *string* nol dengan algoritma *Blowfish* dengan menggunakan subkunci seperti dijelaskan pada langkah (1) dan (2).
4. Ganti P1 dan P2 dengan keluaran dari langkah (3).
5. Enkrip keluaran dari langkah (3) dengan algoritma *Blowfish* dengan subkunci yang sudah dimodifikasi.
6. Ganti P3 dan P4 dengan keluaran dari langkah (5).
7. Lanjutkan proses tersebut, ganti seluruh elemen dari *P-array,* kemudian seluruh keempat Sbox berurutan, dengan keluaran yang berubah secara kontiyu dari algoritma *Blowfish*.

II.5. *Hypertext Preprocessor* (PHP)

PHP merupakan kependekan dari kata *Hypertext Preprocessor*. PHP tergolong sebagai perangkat lunak open source yang diatur dalam aturan *General Purpose*

Lincences (GPL) dan tergolong sebagai bahasa pemrograman PHP yang berbasis *server (server side scripting).* Pemrograman PHP sangat cocok dikembangkan dalam lingkungan web, karena PHP bisa diletakan pada script HTML atau sebaliknya. (Doli Hasibuan 2019)

II.6. *Hyper Text Markup Language* (HTML)

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang *user* untuk membuat dan menyusun bagian paragraf, *heading, link* atau tautan, dan *blockquote* untuk halaman *web* dan aplikasi. HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti *Microsoft Word*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui tag-tag yang ditulis dalam HTML (Fitri Pranita Nasution,2022).

II.7. MySQL

Sistem manajemen *database* SQL yang bersifat *Open Source* dan paling populer saat ini, dapat dipakai oleh siapa saja tanpa membayar dan *source code*-nya bisa diunduh oleh siapa saja. Sistem *database* MySQL mendukung beberapa fitur seperti *multithreaded, multiuser*, dan SQL *database* managemen sistem (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan. (Doli Hasibuan 2019)

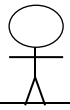
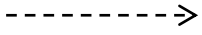
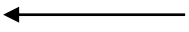
II.8. *Unified Modeling Language* (UML)

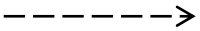
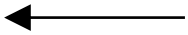
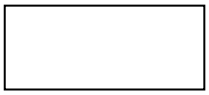
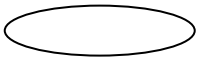
Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang di bangun dengan menggunakan teknik pemrograman berorientasi objek. *Unifed Modeling Language* (UML) muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. (Hendra Nusa Putra, S.Kom, M.Kom, 2018). UML dideskripsikan oleh beberapa diagram diantaranya:

1. *Use Case Diagram*

   Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *use case* diagram dapat digambarkan dengan sumber-sumber pada Tabel II.2.

Tabel II.2. Simbol *Use Case*

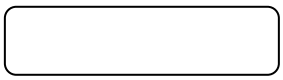
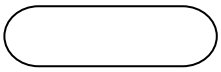
Gambar	Nama	Keterangan
	*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	*Depedency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
	*Generalization*	Hubungan dimana objek anak (descendent) berbagi perilaku dan

		struktur data dari objek yang ada di atasnya objek induk (ancestor).
	*Include*	Menspesifikasikan bahwa use case sumber secara eksplisit.
	*Extend*	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
	Association	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	*System*	Menspesifikan paket yang menampilkan sistem secara terbatas.
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

1. Diagram Aktivitas (*Activity diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

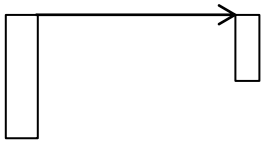
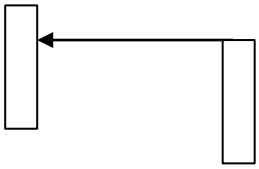
Tabel II.3. Simbol *Activity Diagram*

Gambar	Nama	Keterangan
	*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	*Initial Node*	Bagaimana objek dibentuk atau diawali.
	*Activity Final*	Bagaimana objek dibentuk dan dihancurkan
	*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

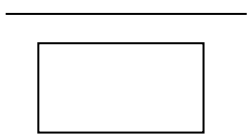
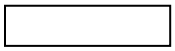
Tabel II.4. Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	*Lifeline*	Objek entity, antarmuka yang saling berinteraksi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

3. *Class Diagram* (Diagram Kelas)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.5.

Tabel II.5. *Class Diagram*

Gambar	Nama	Keterangan
	*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
	Depedency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	Assocation	Apa yang menghubungkan antara objek satu dengan objek lainnya