BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu :

1. (Rohby Visdya Harris, 2019) dengan judul "Analisis Performa Proses Enkripsi dan Dekripsi Menggunakan Algoritme *AES* 128 Pada Berbagai Format *File*" pengujian algoritme *AES* yang dilakukan terhadap berbagai format *file*, didapatkan hasil enkripsi dan dekripsi yang berbeda-beda pada setiap ekstensi file yang diujikan. Pada format dokumen menunjukkan bahwa hasil enkripsi file terhadap ekstensi .docx lebih baik dibandingkan dengan .txt hanya saat *file* berukuran 5000 KB. Sedangkan hasil dekripsi menunjukkan hasil yang bermacam-macam pada setiap ukuran *file* yang diujikan dengan rata-rata hasil ekstensi .txt lebih kecil dibandingkan ekstensi .docx, dengan nilai masing-masing yaitu 0,031194 dan 0,031916. Hasil enkripsi menunjukkan hasil yang bermacam-macam pada setiap ukuran *file* yang diujikan dengan rata-rata hasil ekstensi .jpeg lebih kecil dibandingkan ekstensi .png, dengan nilai masing-masing yaitu 22,509 dan 23,043. Sedangkan hasil dekripsi *file* yang diperoleh menunjukkan bahwa *file* dengan ekstensi .png lebih baik dari pada .jpeg
2. (Widiartha, 2016) Dengan Judul "Implementasi Algoritma Kriptografi *AES* dan Metode *Steganografi LSB* pada Gambar *Bitmap*" hasil dan pembahasan

yang telah didapatkan maka dapat ditarik kesimpulan bahwa algoritma *kriptografi AES* 256 dan metode *steganografi LSB* baik untuk diimplementasikan dalam mengamankan *file* gambar yang kerahasiaannya sangat dijaga. Dari hasil proses enkripsi didapatkan gambar chiper yang tidak dapat dimengerti. Dengan semakin bertambahnya ukuran *file* semakin besar waktu eksekusi dari proses enkripsi. Seperti yang terlihat pada tabel percobaan yang mendapatkan hasil dari waktu eksekusi yang bertambah seiring dengan bertambahnya ukuran *file* gambar.

3. (Ibrahim, Perancangan Keamanan Data Menggunakan Algoritma AES (Advanced Encyption Standard), 2017) dengan judul " Perancangan Pengamanan Data Menggunakan Algoritma *AES (Advanced Encryption Standard)*" Seiring dengan kemajuan teknologi informasi yang semakin pesat maka sangat diperlukan sebuah keamanan data terhadap kerahasiaan informasi. Sangat banyak informasi yang sensitif dan berharga tersebut diakses oleh orang yang tidak bertangung jawab, kemungkinan besar akan merugikan bahkan membahayakan orang yang akan mengirim pesan, maupun penerimanya. Informasi yang terkandung di dalamnya pun bisa saja berubah sehingga menyebabkan salah penafsiran oleh penerima pesan. Selain itu data yang dibajak kemungkinan rusak atau hilang yang menimbulkan kerugian material yang besar. Oleh karena itu untuk menghindari agar hal tersebut tidak terjadi, penulis menggunakan algoritma kriptografi *AES (Advanced Encryption Standard)* untuk proses enkripsi dan deskripsi data.

4. (Aditya Hermawan, 2021) Dengan Judul “Implementasi Enkripsi Data Menggunakan Kombinasi *AES* dan *RSA*” Penelitian implementasi enkripsi data menggunakan kombinasi *AES* dan *RSA* yang memiliki tipe enkripsi yang berbeda, yaitu simetris dan asimetris dapat diterapkan untuk mengenkripsi dan mendekripsi data dengan aman. Hal ini dikarenakan diperlukan dua kali proses dekripsi dan menggunakan kunci pribadi untuk mendapatkan informasi pesan rahasia tersebut. Pembangkitan kunci yang ideal digunakan yaitu 2048 bit dengan waktu pembuatan yaitu 495,56 ms. Proses dekripsi membutuhkan waktu lebih lama dibandingkan proses enkripsi. Proses enkripsi menggunakan *RSA* membutuhkan waktu rata-rata 7,428 ms dan proses enkripsi menggunakan *AES* membutuhkan waktu ratarata 4,365 ms. Sedangkan proses dekripsi menggunakan *RSA* membutuhkan waktu rata-rata 54,068 ms dan proses dekripsi menggunakan *AES* membutuhkan waktu rata-rata 2,753 ms.
5. (Lutfa Ibtihaji Ilham, 2021) dengan judul “Pengembangan Aplikasi Pesan Instan Terenkripsi Menggunakan Algoritma Kriptografi AES (Advanced Encryption Standard)” Aplikasi dapat mengirim dan menerima pesan antar pengguna secara instan atau *real-time*. Metode dan proses enkripsi data pesan dapat menjaga keamanan data pesan oleh pihak yang tidak berhak atau tidak bertanggung jawab. Aplikasi dapat dijalankan pada berbagai platform *Android*, mulai dari versi 4.1 hingga terkini. Aplikasi memiliki autentikasi berupa *email* dan *password* sehingga memungkinkan penggunaan secara global atau umum. Aplikasi memiliki dashboard pesan terbaru untuk mengakses pesan kepada pengguna lain sehingga akses pesan

menjadi lebih mudah dan cepat. Berdasarkan hasil pengujian dengan teknik *black box*, sistem aplikasi yang dikembangkan mampu memperoleh tingkat kelayakan hingga mencapai 96%.

II.2. Landasan Teori

II.2.1. Kriptografi

Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan *(Cryptography is the art and science of keeping messages secure)* selain itu ada pengertian tentang *kriptografi* yaitu *kriptografi* merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata "seni" di dalam definisi di atas maksudnya adalah mempunyai cara yang unik untuk merahasiakan pesan. Kata *"graphy"* di dalam *"cryptography"* itu sendiri sudah menyiratkan sebuah seni. (Ibrahim, 2017).

Untuk dapat menjalakan dengan baik pada proses kriptografi haruslah terdapat emapat elemen utama didalamnya, yang paling berkait satu sama lain, yaitu :

1. *Plain Text* Merupakan sebagai pesan awal atau pesan asli yang dikirim pada proses komunikasi. *Plain Text* inilah yang kemudian dienkripsi dan dideskripsi.
2. *Cipher Text* Merupakan pesan yang tersembunyi, yaitu pesan asli *(Plain Text*) yang telah dienkripsi pada proses kriptografi. Cipher Text ini dapat diubah kembali kebentuk aslinya (*Plain Text*) memanfaatkan *Key* yang telah disediakan.

3. *Cryptography Key* Merupakan kunci yang di gunakan untuk melakukan enkripsi dan deskripsi pada proses kriptografi. Tanpa adanya kunci (*key*) yang sama maka proses enkripsi dan deskripsi tidaka dapat dilakukan dengan baik. Kunci (*key*) merupakan informasi yang padat menjadi kendali terhadap proses terjadinya *kriptografi*
4. *Encryption Decryption* Algorithm Komponen terakhir yang juga sama pentingnya dalam proses kriptografi adalah algoritma yang di gunakan untuk enkripsi dan dekripsi.

II.2.2. Gambar / Image

Gambar merupakan sebuah sarana yang segala sesuatunya diwujudkan dengan mengilustrasikan kedalam bentuk dua dimensi sebagai curahan ataupun pemikiran yang bentuknya bermacam-macam seperti slide, potret, lukisan, film, opaque projector, ataupun strip (Hamalik, 2017). Gambar adalah media yang sering digunakan, gambar bisa disebut Bahasa yang umum, yang bisa dimengerti dan dinikmati dimanapun (Sadiman, 2017).

Dari pengertian diatas dapat disimpulkan bahwa gambar adalah hasil pengaktualan dari curahan pikiran atau imajinasi yang divisualisasikan ke dalam bentuk dua dimensi agar dapat dipahami dan dimengerti oleh orang lain.

II.2.3. Algoritma AES

Advanced Encryption Standard (AES) merupakan algoritma *cryptographic* yang dapat digunkan untuk mengamankan data. Algoritma *AES* adalah blok *chipertext* simetrik yang dapat mengenkripsi *(encipher)* dan dekripsi *(decipher)* informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut *ciphertext*,

sebaliknya dekripsi adalah merubah *ciphertext* data menjadi bentuk semula yang kita kenal sebagai *plaintext.*

Algoritma *AES* menggunakan kunci *kriptografi* 128, 192, dan 256 bits untuk mengenkrip dan dekripsi data. Seiring dengan kemajuan teknologi informasi yang semakin pesat maka sangat diperlukan sebuah keamanan data terhadap kerahasiaan informasi terutama berupa gambar. Sangat banyak informasi yang sensitif dan berharga tersebut diakses oleh orang yang tidak bertangung jawab, kemungkinan besar akan merugikan bahkan membahayakan orang yang akan mengirim gambar, maupun penerimanya. Makusd gambar yang terkandung di dalamnya pun bisa saja berubah sehingga menyebabkan salah penafsiran oleh penerima gambar. Selain itu data yang dibajak kemungkinan rusak atau hilang yang menimbulkan kerugian material yang besar. Oleh karena itu untuk menghindari agar hal tersebut tidak terjadi, penulis menggunakan algoritma kriptografi *AES (Advanced Encryption Standard)* untuk proses enkripsi dan deskripsi sebuah gambar. (Ibrahim, 2017).

Adapun tiga buah versi *AES* yang dapat dilihat pada Tabel II.1.

Tabel II. 1. Tiga Buah Versi AES

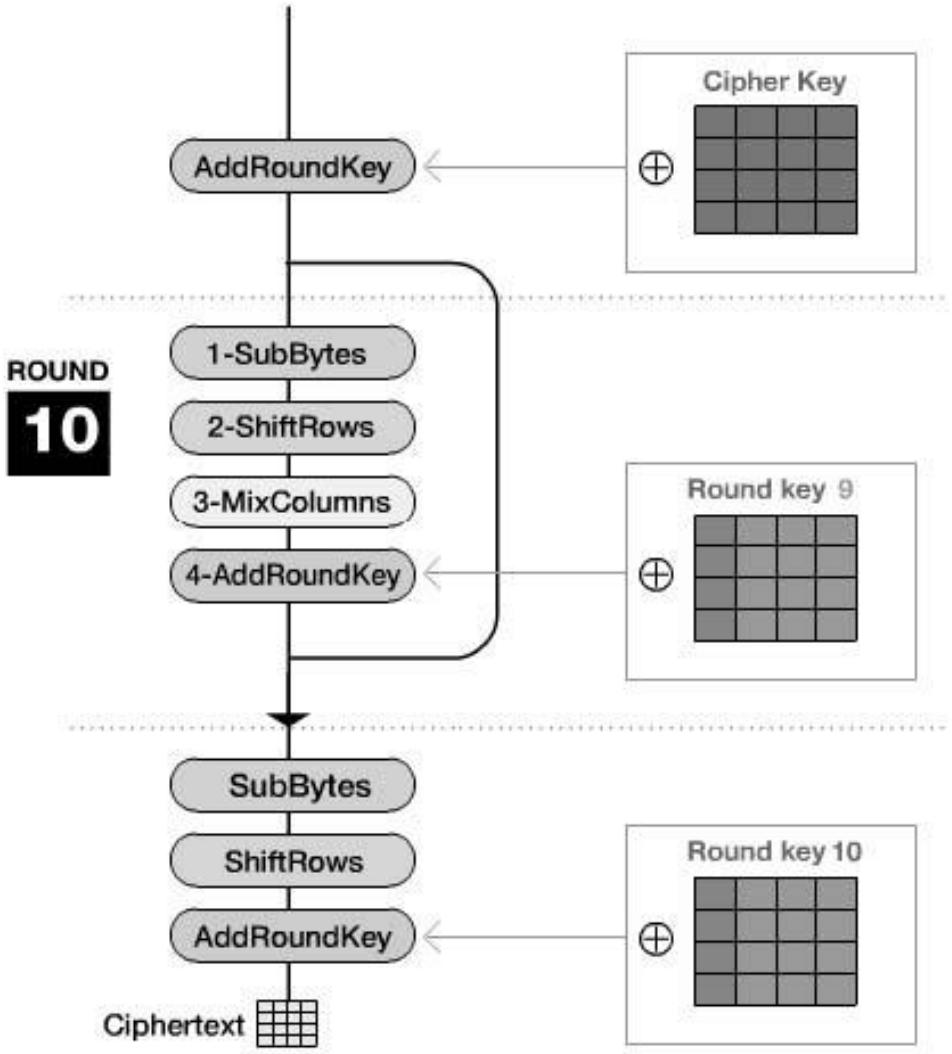
Tipe Kunci	Panjang Kunci	Ukuran Blok	Jumlah Putaran
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

(Apianto, Yoga, n.d, 2018)

Pada garis besar algoritma *rijndael* yang beroperasi pada blok 128-bit dengan kunci 128-bit adalah sebagai berikut (di luar proses pembangkitan *round key*) adalah sebagai berikut :

1. AddRoundKey, adalah melakukan *XOR* antara *state* awal (plainteks) dengan *cipher key*, tahap ini disebut juga *initial round.*
2. Putaran sebanyak Nr-1 kali. Proses yang dilakukan pada setiap putaran :
   a. *SubBytes* adalah subsitusi *byte* dengan menggunakan tabel subtitusi (S-box).
   b. *ShiftRows* adalah pergeseran baris-baris *array state* secara *wrapping*.
   c. *MixColumns* adalah mengacak data di masing-masing kolom *array state*.
   d. AddRoundKey adalah melakukan *XOR* antara *state* sekarang *round key*.

3. Final *round* adalah proses untuk putaran terakhir :
   a. *SubBytes*
   b. *ShiftRows*
   c. AddRoundKey



Gambar II. 1. Diagram Proses Enkripsi AES (Advanced Encryption Standard)

II.2.3. Hypertext Preprocessor (PHP)

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. (Sugijanto, dkk, 2020 : 2).

PHP merupakan singkatan dari "*Hypertext Preprocessor*". PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrogramn C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang Spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa file dalam 1 file dengan cara di include dan require. PHP itu sendiri sudah dapat berinteraksi dengan beberpa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle. (Rahmasari, 2019 : 414).

II.2.4. My Structure Query Language (MySQL)

Definisi MySQL merupakan *software* RDMS (*Relational Database Management System*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak pengguna dan dapat melakukan suatu proses secara sinkron atau bersamaan. (Rahmasari, 2019 : 414).

MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database* pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread, multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. (Sugijanto, dkk, 2020 : 2).

II.2.4. Unfield Modeling Language (UML)

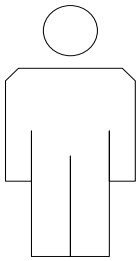
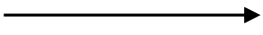
UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

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

Simbol	Deskripsi
	*Use Case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja
	*Actor* atau Aktor adalah *Abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki kontrol terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi

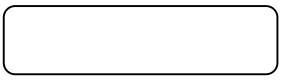
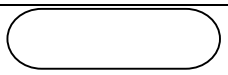
(Sumber : Hendini, 2016)

2. 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

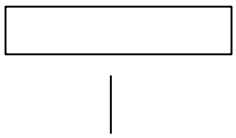
Gambar	Nama	Keterangan
	Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	Initial Node	Bagaimana objek dibentuk atau diawali.
	Activity Final	Bagaimana objek dibentuk dan dihancurkan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

(Sumber : Andikos, 2019 : 39)

3. 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 da ri komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

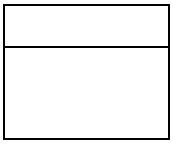
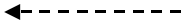
(Sumber : Andikos, 2019 : 39)

4. *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 lainlain. Class diagram dapat digambarkan dengan simbol-simbol seperti pada Tabel II.5.

Tabel II. 5. Simbol 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.

Tabel II. 5. Simbol Class Diagram Lanjutan

	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

(Sumber : Andikos, 2019 : 39)