BAB II

TINJAUAN PUSTAKA

II.1. Keaslian Penelitian

Penelitian yang telah dilakukan oleh Ahmad Muazim Abidin,Fitria Hardianti.P,Indra Nur Setiani (2016) yang berjudul Analisa Dan Implementasi Proses Kriptografi *Encryption-Decryption* Dengan *Algoritma Advanced Encryption Standard* (Aes-128), dari penelitian tersebut diperoleh hasil : AES Rijndael merupakan algoritma yang cukup sulit untuk dipecahkan saat ini, karena belum ada serangan atau pemecahan yang benar-benar mampu secara analisis matematis dengan efektif dan efisien dengan alasan pola yang dibentuk cukup acak. Keacakan pola tersebut didapat dari sebagian teknik AES sebagai kekuatan yang dimiliki algoritma ini, yaitu SubBytes() dan MixColoums()yang dibangun secara nonlinier sehingga menjadi tantangan kriptoanalisis linier. Besar file ciphertext yang dihasilkan dari proses enkripsi dan besar file plaintext akhir dari proses dekripsi untuk algoritma AES: Rijndael memiliki besar file yang berbeda.

Penelitian kedua yang dilakukan oleh Asri Prameshwari, Nyoman Putra Sastra (2018), yang berjudul Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen, menghasilkan aplikasi untuk melakukan enkripsi dan dekripsi dengan penerapan algoritma Aes128 yang dibangun dengan *Visual Studio Net*, dan menghasilkan kesimpulan sebagai berikut bahwa algoritma AES-128 dapat dijadikan salah satu alternatif untuk proses keamanan data

dalam hal ini enkripsi dan dekripsi file dokumen dan Ukuran file merupakan salah satu variabel yang cukup penting karena berpengaruh terhadap waktu proses enkripsi dan dekripsi. Pada variabel hasil, waktu merupakan tolak ukur dari proses, apakah terhitung cepat atau lambat dari ukuran file yang harus diproses.

Penelitian lainnya yang dilakukan peleh Muhammad Taufiqur Rahman, Aryo Pinandito, Eko Sakti Pramukantoro (2017) dengan judul Perbandingan Performansi Algoritme Kriptografi Advanced Encryption Standard (AES) dan Blowfish pada Text di Platform Android, menghasilkan kesimpulan : Algoritma Blowfish memiliki performa kecepatan algoritme terhadap panjang data sedikit lebih baik daripada algoritme AES secara keseluruhan dengan keunggulan 0.2 ms saat enkripsi dan 0.12 ms saat dekripsi, Algoritma Blowfish memiliki performa kecepatan algoritma terhadap panjang kunci sedikit lebih baik daripada algoritme AES secara keseluruhan dengan keunggulan 0.18 saat enkripsi dan 0.26 saat dekripsi.

Penelitian lainnya yang dilakukan oleh Ade Handini (2016) dengan judul Permodalan Sistem Informasi Monitoring Penjualan Dan Stok Barang , menghasilkan kesimpulan dengan adanya sistem informasi monitoring penjualan dan stok barang ini, mempermudah pelaku usaha dalam memantau atau mengetahui penjualan dan stok barang ditiap cabang.

Penelitian lainnya yang dilakukan oleh Ibnu Akil dengan judul Rekayasa Perangkat Lunak Dengan Model *UNIFIED PROCESS,* menghasilkan bahwasannya *Unified process* sebagai kerangka kerja proses rekayasa perangkat lunak terasa cukup ringan dan tidak terlalu membebani pengembang dengan proses-proses yang tidak terlalu penting.

Penelitian lainnya yang dilakukan oleh Sari, R, N, & Hayati, R, S, (2018, August) dengan judul *Beaufort Cipher Algorithm Analysis Based on the Power Lock-Blum Blum Shub in Securing Data. In 2018 6th International Conference On Cyber and IT Service Management (CITSM) (pp. 1-4). IEEE,* menghasilkan Kriptografi nama adalah kombinasi dari *cryptos* yunani (tersembunyi) dan *logo* (studi, sains) oleh karena itu, kata kriptografi secara harfiah menyiratkan ilmu menyembunyikan. Kriptografi mencakup teknik seperti microdot, menggabungkan kata-kata dengan gambar, dan cara lain untuk menyembunyikan informasi dalam penyimpanan.

II.2. Definisi Kriptografi

Kriptografi berasal dari Bahasa Yunani, yaitu kripto dan graphia. Menurut bahasa kripto berarti rahasia (*secret*) dan graphia berarti tulisan (*writing*). Menurut terminologi, kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan ketika pesan dikirim darisuatu tempat ke tempat yang lain. Secara keseluruhan kriptografi dapat disimpulkan sebagai ilmu yang mempelajari tentang pengacakan pesan dengan fungsi matematika agar tidak bisa dibaca oleh pihak yang tidak berwenang. Kriptografi merupakan studi matematika yang mempunyai hubungan dengan aspek keamanan informasi seperti integritas data dan keaslian data. Dalam penerapannya, kriptografi merupakan suatu metode enkripsi atau penyandian data yang hanya diketahui atau berarti oleh suatu kelompok pengguna tertentu. Metoda ini telah dikenal sejak lama, salah satu contoh penggunaannya pada masa keKaisaran Romawi Kuno. Pada waktu itu Julius Caesar tidak menginginkan berita atau pesan yang

dibawa oleh kurir-kurirnya jatuh kepada pihak lawan. Oleh karena itu, beliau menggunakan sistem substitusi sederhana, yang kini disebut dengan Caesar Cipher.

Dalam kriptografi ada beberapa istilah yang seringdigunakan, antara lain sebagai berikut :

1. Plaintext adalah informasi asli sebelum dienkripsi atau teks terang.
2. Enkripsi adalah proses kriptografi dari plaintext menjadi ciphertext.
3. Ciphertext adalah informasi acak yang berasal dari plaintext yang telah dimasukkan ke dalam fungsi kriptografi atau dienkripsi.
4. Dekripsi adalah proses pengubahan ciphertextmenjadi plaintext.
5. Kriptoan analisis adalah studi yang mempelajari teknik matematika untuk memecahkan teknik kriptografi.
6. Kriptoanalis adalah orang yang melakukan kriptonalisis.
7. Kriptologi adalah ilmu tentang kriptografi dan kriptonalisis.

(Ahmad Muazim Abidin, 2016 : 3)

II.2.1. Metode AES256

Algoritma AES adalah blok *ciphertext* simetrik yang dapat mengenkripsi *(enchiper)* dan dekripsi *(dechiper)* informasi. Algoritma AES merubah folder asli kedalam bilangan biner yang tidak dapat dibaca atau dibuka oleh siapapun dan ketika kita ingin melihat *file* tersebut kita harus mendekripsi *file* tersebut dengan memasukkan kunci yang sudah kita tentukan untuk mendekripsi *file.*

II.2.2. Tujuan Kriptografi

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

1. Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas data, adalah 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 penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
3. Autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
4. Non-repudiasi., atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

(*Sumber : Ahmad Muazim Abidin, 2016 : 3*)

II.3. UML (*Unified Modelling Language*)

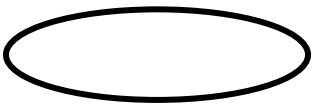
Menurut Ade Hendini (2016) *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

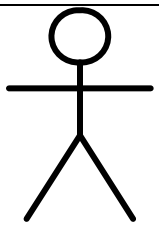
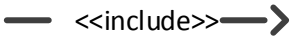
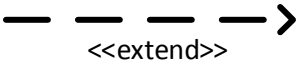
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

a. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuakn (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel II.1 Simbol *Use Case Diagram*

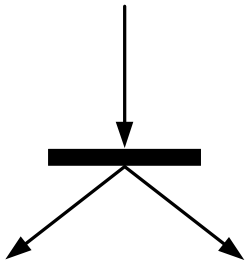
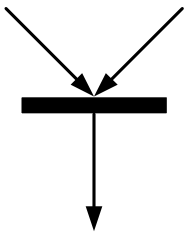
Gambar	Keterangan
	*Use Case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif yang dinyatakan dengan menggunakan kata kerja.
	Actor atau Aktor adalah *Abstraction* dari orang atau sistem yang

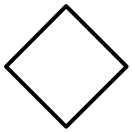
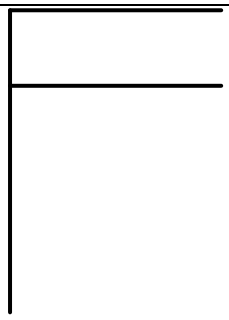
	lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifika sikan aktir, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *Use Case*, 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 didalam *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

b. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

Tabel II.1 Simbol *Activity Diagram*

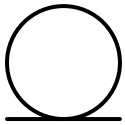
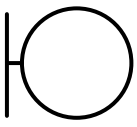
Gambar	Keterangan
	*Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	*End Point*, akhir aktivitas
	*Activities,* menggambarkan suatu proses/kegiatan bisnis
	*Fork* / percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	*Join* (penggabungan) atau *rake*, digunakan untuk menunjukkan adanya dekomposisi

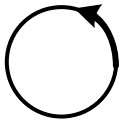
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *tru* atau *false*
	*Swimlane*, pembagian *activity diagram* untuk menunjukkan siapa melakukan apa

c. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel II.3 Simbol *Sequence Diagram*

Gambar	Keterangan
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interfaces* atau interaksi antara satu atau lebih aktor dengan

	sistem, seperti tampilan form entry dan form cetak
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	*Message*, simbol mengirim pesan antar *class*
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *tru* atau *false*
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*

d. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class Diagram secara khas meliputi : Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, attribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau attribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality.*

Tabel II.4 ***Multiplicity Class Diagram***

Multiplicity	**Penjelasan**
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4