# BAB II

# TINJAUAN PUSTAKA

## II.1. Penelitian Terkait

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

Badrul Anwar (2020).menghasilkan sebuah sistem pengamanan dokumen penjualan tiket pesawat menggunakan metode *Advanced Encryption Standart* Pada PT. Benua Raya Jaya Tour and Travel dapat mempermudah dan mempercepat dalam pembuatan dokumen penjualan tiket pesawat dan juga memberikan keamanan dalam hal penyimpanan dokumen penjualan tiket pesawat di *database*.

Penelitian lain yang dilakukan oleh Lilik Asih Indrayani (2019) Pada penelitian ini, algoritma AES akan dimodifikasi dengan meningkatkan jumlah putaran bersamaan dengan panjang kunci menjadi 320 bit dengan 16 putaran dengan tujuan meningkatkan keamanan dari algoritma AES. Pengujian dilakukan dengan membandingkan waktu proses enkripsi dan dekripsi antara algoritma AES standar 10 putaran dengan algoritma AES modifikasi 16 putaran. File dokumen yang dapat dienkripsi hanya berupa file dengan format pdf, docx, dan txt. Hasil pengujian menunjukkan bahwa semakin besar putaran dan panjang kunci, maka semakin lama waktu yang digunakan dalam proses enkripsi maupun dekripsi. Hal ini dapat dibuktikan dengan algoritma AES modifikasi yang memiliki nilai waktu proses lebih besar dibanding algoritma AES standar sehingga dapat disimpulkan

algoritma AES modifikasi memiliki tingkat keamanan yang lebih tinggi karena berpengaruh pada waktu yang dibutuhkan seorang kriptoanalis untuk memecahkan kode enkripsi.

Penelitian lain yang dilakukan oleh Jaka Prayudha, Saniman, Ishak (2019) sebuah sistem dalam pengamanan data yang dapat melakukan penyandian dan pengacakan sebuah informasi yang berbasis Komputer. Pengamanan ini dilakukan dengan cara menerapkan sebuah algoritma kriptografi yang bertujuan untuk mengenkripsi dan deskripsi sebuah pesan text. Dan Metode yang digunakan adalah Algoritma Kriptogri Algoritma *Advanced Encryption Standard (AES)* . Hasil pengujian menunjukan bahwa system keamanan data gaji karyawan dapat mengamankan data gaji dengan sangat baik dan menghindari terjadinya penyalahgunaan atau manipulasi data oleh orang-orang yang tidak memiliki wewenang atas data tersebut.

Penelitian lain yang dilakukan oleh Asri Prameshwari (2018) menghasilkan suatu implementasi kriptografi algoritma *AES-128* untuk enkripsi dan deskripsi data yang berupa file *dokumen (PDF, DOC, TXT). Algoritma Advanced Encryption Standard (AES)* dipilih karena memiliki suatu tingkat keamanan pertukaran informasi yang cukup bagus dan pada penelitian ini diuji coba file *dokumen* untuk melihat waktu yang dibutuhkan selama proses enkripsi dan deskripsi. Metode yang digunakan dalam penelitian ini adalah Algoritma *Advanced Encryption Standard (AES )*

Penelitian lain yang dilakukan oleh Aris (2017) Dengan Metode Kriptografi Algoritma AES adalah merupakan algoritma cryptographic yang dapat

digunakan mengaman data. Algoritma AES merupakan blok chipertext simetrik yang dapat mengenkripsi (enchiper) dan deskripsi (decipher) informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut chipertext, sebaliknya deskripsi adalah merubah chipertext data menjadi bentuk semula yang kita kenal sebagai plaintext serta dan menggunakan Algoritma Kompresi Huffman yaitu merupakan algoritma yang paling terkenal untuk mengompres teks Dengan menggunakan Pemogramanan PHP maka dapat dibuat sebuah Aplikasi untuk mengamankan Data.

## II.2. Landasan Teori

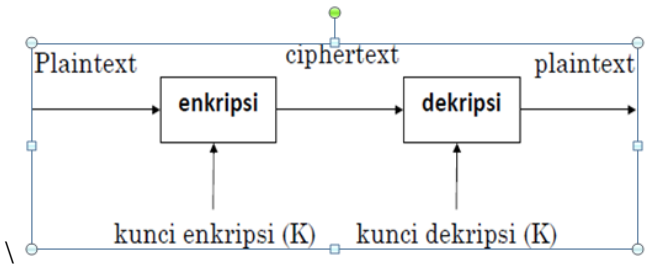
### II.2.1. Definisi Kriptografi

Kriptografi berasal dari Bahasa Yunani, yaitu kripto dan graphia. Menurut bahasa kripto berarti rahasia (secret) dan graphia berarti tulisan (writting). Menurut terminology, Kriptografi adalah ilmu seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain. Secara keseluruhan kriptografi dapat disimpulkan sebagai ilmu yang mempelajari tentang pengacakan pesan dengan fungsi perhitungan 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. Selain itu kriptografi juga dapat dibagi berdasarkan jenis kunci yaitu algoritma simetris dan asimetris. Agoritma simetris adalah algortima yang menggunakan kunci yang sama untuk melakukan enkripsi dan dekripsi, sedangkan algoritma asimetris adalah algoritma yang menggunakan kunci berbeda

untuk melakukan enkripsi dan dekripsi, untuk eknripsi menggunakan kunci public dan dekripsi menggunakan kunci private.

### II.2.2 Kriptografi Simetris

Algoritma simetris (symmetric algorithm) adalah algortima yang menggunakan kunci yang sama untuk melakukan enkripsi dan dekripsi




**Gambar II.1 Proses algoritma kriptografi simetris**
(Sumber : Kristoforus,Aditya ; 2014)

### II.2.3 Kriptografi Simetri

AES Merupakan standar algoritma kriptografi terbaru yang dipublikasikan oleh NIST ( *National Institute of Standard and Technology )* sebagai pengganti algoritma DES ( *Data Encryption Standard )* yang sudah berakhir masa penggunaanya. Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendeskripsi data dengan panjang kunci 128 bit, 192 bit, dan 256 bit. Pada Algoritma AES (Yuniarti dkk, 2009)

Pada algoritma AES, jumlah blok input, blok output, dan state adalah 128 bit. Dengan besar data 128 bit, berarti Nb = 4 yang menunjukkan panjang data 22 tiap baris adalah 4 byte. Dengan blok input atau blok data sebesar 128 bit, key yang digunakan pada algoritma AES tidak harus mempunyai besar yang sama

dengan blok input. Cipher key pada algoritma AES bisa menggunakan kunci dengan panjang 128 bit, 192 bit atau 256 bit. Perbedaan panjang kunci akan mempengaruhi jumlah round yang akan di implementasikan pada algoritma AES ini (Gilang Gumira ; 2016 : 280).

### II.2.4. *PHP*

PHP adalah bahasa yang dirancang secara khusus untuk penggunaan pada *Web*. PHP adalah *tool* untuk pembuatan halaman web dinamis. Pada awalnya PHP merupakan kependekan dari Personal *Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data form dari *web*. Saat ini PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: *Hypertext Preprocessor* (Ahmad Lutfi, 2017 ; 105).

### II.2.5. *MySQL*

*MySQL* adalah salah satu aplikasi DBMS (*Database Management System*) yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Dalam sistem database tak relasional, semua informasi disimpan pada satu bidang luas, yang kadangkala data di dalamnya sangat sulit dan melelahkan untuk diakses. Tetapi *MySQL* merupakan sebuah sistem database relasional,

sehingga dapat mengelompokkan informasi ke dalam tabel-tabel atau grup-grup informasi yang berkaitan. Setiap tabel memuat bidang-bidang yang terpisah, yang mempresentasikan setiap bit informasi. MySQL menggunakan indeks untuk mempercepat proses pencarian terhadap baris informasi tertentu. MySQL memerlukan sedikitnya satu indeks pada tiap tabel. Biasanya akan menggunakan suatu *primary key* atau pengenal unik untuk membantu penjejakan data (Ahmad Lutfi, 2017; 106).

### II.2.6. *Database*

Database ialah suatu wadah untuk menampung sebuah data yang ada pada sebuah system. Database juga bisa diartikan sebagai kumpulan data. Database juga bisa dikenal formal dan tegas. Database juga bisa diartikan dengan kumpulan data yang terintegrasi yang dapat dimanipulasi, diambil dan dicari secara tepat. ( Hesananda et aL 2017).

## II.7. *Unified Modeling Language* (UML)

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah satu alat bantu yang sangat handal didunia pengembnagan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam membentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif

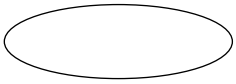
untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar ; 2018 : 49).

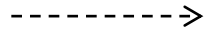
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan tipikal interaksi antara (pengguna) sebuah *system*dengan sistemnya sendiri melalui sebuah cerita bagaima sebuah sistem yang dipakai (Munawar ; 2018 : 89).

**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 aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *Use Case*. |
|  | 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 *control* 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 aliran 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 : Munawar ; 2018)**

2. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggng jawab entitas yang menentukan perilaku sistem.

*Class diagram* merupakan diagram statis dari suatu aplikasi. Class Diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi *(executable code)* dari aplikasi perangkat lunak (Munawar ; 2018).

**Tabel II.2. Simbol *Class Diagram***

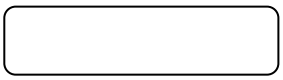
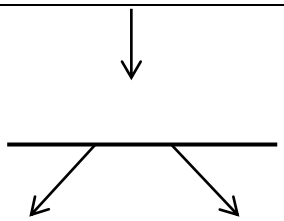
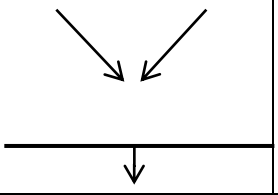
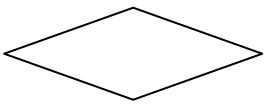
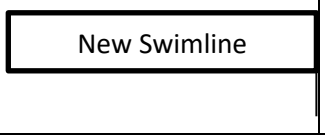
| ***Multiplicity*** | **Penjelasan** |
|---|---|
| 1 | Satu dan hanya satu |
| 0..* | Boleh tidak ada atau 1 atau lebih |
| 1..* | 1 atau lebih |
| 0..1 | Boleh tidak ada, maksimal 1 |
| n..n | Batasan antara. Contoh 2..4<br>mempunyai arti minimal 2 maksimal 4 |

***(Sumber :* Munawar ; 2018*)***

3. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram* (Munawar ; 2018).

**Tabel II.3. Simbol Diagram Aktivitas**

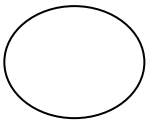
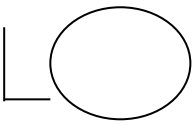
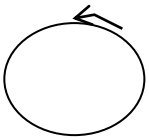
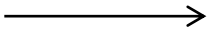
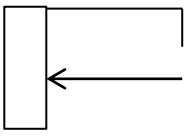
| Gambar | Keterangan |
|---|---|
|  | *Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas. |
|  | *End point*, akhir aktifitas. |
|  | *Activites*, menggambarkan suatu proses/kegiatan bisnis. |
|  | *Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu. |
|  | *Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi. |
|  | *Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true*, *false*. |
|  | *Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa. |

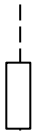
***(*Sumber : Munawar ; 2018*)***

4. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* adalah salah satu *interaction diagram*. Karena sequence diagram mengacu kepada obyek, maka sbelum membuat diagram ini class diagram sudah harus teridentifikasi (Munawar ; 2018).

**Tabel II.4. Simbol Diagram Urutan**

| Gambar | Keterangan |
|---|---|
|  | *EntityClass*, 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 *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan *formentry* 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. |

| | |
|---|---|
|  | *Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi. |
|  | *Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*. |

***(Sumber : Munawar ; 2018)***