BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terkait

Penelitian yang dilakukan oleh Madya Satria, Rika Rosnelly, Nurhayati (2019) yang berjudul **"Perancangan Aplikasi Keamanan Data Dokumen *Word* dengan Menggunakan *Algoritma Triple DES*".** Dalam perancangan aplikasi ini salah satu metode yang dianggap tangguh dalam melakukan pengamanan ini adalah metode *algoritma triple des* yang merupakan penyandian dengan cara mengubah letak dari huruf-huruf pada pesan yang akan disandikan. Dan untuk membaca pesan aslinya kembali, cukup dengan mengembalikan letak dari huruf-huruf pada pesan tersebut berdasarkan kunci dan algoritma enkripsi data *file doc* yang telah disepakati pihak pengirim dan penerima. Hasil dari uji coba yang dilakukan terhadap berbagai jenis file menunjukkan bahwa aplikasi yang digunakan untuk tipe *file doc* tanpa ada kesalahan.

Penelitian yang dilakukan oleh Arief Rilo Pambudi, dkk (2018) yang berjudul **"Implementasi Kriptografi Pada Email Menggunakan Algoritma Rivest Code 4 (RC4) Dan Data Encryption Standart (DES) Berbasis Java Dekstop Pada PT VEPRO NUSA PERSADA".** Dalam perancangan aplikasi ini menggunakan metode pengembangan system dengan model waterfall. Aplikasi menggunakan algoritma Rivest Code 4 (RC4) dan Data Encryption Standart (DES) dengan bahasa pemrograman java, dimana kunci enkripsi dan dekripsi

diatur oleh pengguna, hal ini dilakukan untuk meminimalisir kesalahan dari sistem. Algoritma Rivest Code 4 (RC4) dan Data Encryption Standart (DES) diharapkan mampu memproses enkripsi dan dekripsi data dengan waktu yang lebih cepat dan pengamanan yang lebih aman. Adapaun file yang dapat diproses oleh aplikasi hanya file berekstensi .docx,.xlxs, .pptx, dan .pdf.

Penelitian lain yang dilakukan oleh Sabar Hanadwiputra (2018) dengan judul **"*Implementasi Enkripsi Dalam Pengamanan File Data Karyawan Dengan Metode Algoritma DES (Data Encryption Standard) Pada CV. Sinergi Informasi Global"*.** Penelitian ini menghasilkan sebuah aplikasi *kriptografi* pengamanan file yang berfungsi untuk melakukan enkripsi dan deskripsi data dengan metode DES pada jenis file tertentu sehingga data tidak mudah diketahui orang. Yang membedakan dari penelitian yang akan dibuat dalam penelitian ini adalah data yang diamankan hanya berupa data File kontainerdengan menggunakan *Algoritma DES* berbasis *web*.

Penelitian lain yang dilakukan oleh Dhimas Dwiki Ramadhan Wicaksono, dkk (2019) dengan judul **"Rancang Bangun *Server File Transfer Protocol* (FTP) Dengan Metode Keamanan Jaringan Enkripsi Berbasis *Data Encryptiom Standard* (DES) Dan *Advanced Encryption Standard* (AES)".** Dari hasil pengujian dapat disimpulkan bahwa jaringan FTP dapat sniffing dengan menggunakan perangkat wireshark. File yang telah dienkripsi, tidak dapat terbaca keseluruhan oleh *wireshark*, ada beberapa karakter yang tidak dapat ditampilkan di TCP *stream*. Panjang teks yang akan dienkripsi, berpengaruh pada hasil enkripsi DES, sedangkan pada metode AES, hasil enkripsi memiliki jumlah

karakter yang sama dengan file aslinya. Waktu yang diperlukan *bruteforce* untuk memecahkan enkripsi DES adalah 0.075 detik. Sedangkan enkrpsi AES memerlukan waktu 0.125 detik untuk pemecahannya.

Berdasarkan hasil penelitian yang terdahulu, maka dibuatlah kesimpulan untuk merancang sebuah aplikasi untuk mengimplementasikan sebuah algoritma yang belum pernah digunakan sebelumnya untuk melakukan enkripsi dan dekripsi terhadap data *file* dokumen kontainer yaitu *Algoritma DES*. Sehingga pada penulisan skripsi ini dibuatlah sebuah judul **"*Perancangan Pengamanan Data File Kontainer Pada PT. Menggunakan Metode DES Berbasis Web*"**. Berdasarkan judul tersebut nantinya akan dihasilkan sebuah aplikasi untuk melakukan enkripsi dan dekripsi terhadap data File dokumen kontainer.

II.2. Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk membantu meayani kebutuhan seseorang untuk melakukan aktivitas. Dengan adanya aplikasi maka kebutuhan akan pelayanan sebuah aktivitas menjadi lebih baik. Dimana setiap pekerjaan dapat dilakukan dengan mudah kalau menggunakan sebuah aplikasi (Agusdi Syafrizal: 2018).

II.2.1. *Kriptografi*

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa

mengalami gangguan dari pihak ketiga. *Kriptografi* adalah ilmu pengetahuan dan seni menjaga *message* agar tetap aman. Tujuan penerapan *kriptografi* adalah untuk membuat sesuatu yang tersembunyi, dapat suatu pesan rahasia berupa teks, suara, gambar dan video. Di dalam sistem *kriptografi* terdapat 5 bagian yaitu

1. *Plaintext* adalah pesan atau data dalam bentuk aslinya teks yang dapat terbaca. *Plaintext* adalah masukan bagi algoritma enkripsi.

2. *Secret Key* adalah masukan bagi algoritma enkripsi merupakan nilai yang bebas terhadap teks asli dan menentukan hasil keluaran algoritma enkripsi.

3. *Chipertext* adalah keluaran algoritma enkripsi. *Ciphertext* dapat dianggap sebagai pesan tersembunyi yang akan terlihat acak.

4. *Algoritma Enkripsi* memiliki 2 masukan teks asli dan kunci rahasia. *Algoritma enkripsi* melakukan transpormasi terhadap teks asli sehingga menghasilkan teks sandi.

5. *Algoritma Dekripsi* memiliki 2 masukan yaitu teks sandi dan kunci rahasia. *Algoritma dekripsi* memulihkan kembali teks sandi menjadi teks asli bila kunci rahasia *algoritma enkripsi* sama dengan *algoritma dekripsi*(Guntur Tri Wibowo, dkk: 2015).

II.2.2. *Algoritma DES*

Algoritma DES (*Data Encryption Standard*) merupakan algoritma cipher blok yang popular karena dijadikan standar algoritma enkripsi kuncisimetri. Secara umum DES terbagi menjadi tiga kelompok yaitu pemrosesan kunci,

enkripsi data 64 bit dan deskripsi data 64 bit yang mana satu kelompok saling berintegrasi satu sama lain (Sabar Hanadwiputra: 2018).

II.2.3. Enkripsi dan Dekripsi

Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yangtidak berhak.

a. Enkripsi

Encrypt atau enkripsi merupakan sebuah teknik yang dilakukan mengacak data asli menjadi kode rahasia sehingga menyulitkan orang yang tidak berkepentingan untuk mengakses dan mengetahui data yang asli. (Arisantoso,dkk; 2017).

b. Dekripsi

Decryption atau dekripsi adalah kebalikan dari enkripsi, dimana berfungsi untuk mendeskripsikan data yang telah dienkripsi sehingga data yang telah menjadi kode rahasia diubah kembali menjadi data biasa atau aslinya (Arisantoso,dkk; 2017).

II.2.4. Pengertian Data

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masihmemerlukan adanya suatu pengolahan. Data bisa berwujud suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan,obyek, kejadian ataupun suatu konsep (Eka Iswandy; 2015).

II.2.5. *PHP*

PHP adalah salah satu *server* side yang dirancang khusus untuk aplikasi web. PHP disisipkan diantara bahasa HTML dan karena bahasa *server side*, maka bahasa PHP akan dieksekusi di *server*, sehingga yang dikirimkan ke *browser* adalah hasil jadi dalam bentuk HTML, dan kode PHP tidak akan terlihat.PHP termasuk *OpenSource Product*. Jadi, dapat diubah *source code* dan mendistribusikanya secara bebas (Priyo S; 2016 : 25).

II.2.6. *Web Server*

Menurut Fathansyah (2012:466) menerangkan bahwa pengertian *web server* adalah "*Server Web* (*Web Server*) merujuk pada perangkat keras (*server*) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai". *(*Agus Prayitno, dkk; 2015 : 2).

II.2.7. *MySQL Server*

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.2.8. *Database*

Pengertian database menurut Jogiyanto dalam buku analisis & desain (2005:217) adalah "merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasiinya. Database juga merupakan salah satu komponen yang penting di sistem infomasi, karena berfungsi sebagai penyedia informasi bagi para pemakainya (Leonard Tambunan, dkk; 2018: 132).

II.2.8.1. Normalisasi

Normalisasi adalah proses pengelompokkan atribut data yang membentuk entittas sederhana, fleksibel, dan mudah beradaptasi, sehingga dapat dipastikan bahwa database yang dibuat berkualitas baik.

Adapun bentuk-bentuk normalisasi sebagai berikut :

1. **Bentuk normal tahap pertama (1" Normal Form)**

   Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok bahan

bakar minyak berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri(Janner Simarmata ; 2010 : 77).

Contoh normalisasi 1NF adalah seperti pada table berikut :

Tabel I.1. Tabel Bentuk Normal Pertama (1NF)

p#	Status	kota	b#	Qty
p1	20	Batam	b1	300
p1	20	Batam	b2	200
p1	20	Batam	b3	400
p1	20	Batam	b4	200
p1	20	Batam	b5	100
p1	20	Batam	b6	100
p2	10	Medan	b1	300
p2	10	Medan	b2	400
p3	10	Medan	b2	200
p4	20	Batam	b2	200
p4	20	Batam	b4	300
p4	20	Batam	b5	400

2. Bentuk normal tahap kedua (2nd normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk

normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama(Janner Simarmata ; 2010 : 77).

Tabel II.2. Tabel Bentuk Normal Kedua (2NF)

Pemasok2

p#	Status	Kota
P1	20	Batam
P2	10	Medan
P3	10	Medan
P4	20	Batam
P5	30	Bandung

Barang

p#	b#	Qty
p1	b1	300
p1	b2	200
p1	b3	400
p1	b4	200
p1	b5	100
p1	b6	100
p2	b1	300
p2	b2	400
p3	b2	200
p4	b2	200
p4	b4	300
p4	b5	400

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap

kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya(Janner Simarmata ; 2010 : 77).

Tabel II.3. Tabel Bentuk Normal Ketiga (3NF)

Pemasok Kota

p#	Kota
P1	Batam
P2	Medan
P3	Medan
P4	Batam
P5	Bandung

Kota Status

Kota	Status
Batam	20
Medan	10
Batam	20
Bandung	30

4. *Boyce Code Normal Form* (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF(Janner Simarmata ; 2010 : 78).

5. Bentuk Normal Keempat (4NF)

Sebuah tabel rasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalue merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan *multivalue* (MVD). Sebuah ketergantungan *multivalue*

tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda(Janner Simarmata ; 2010 : 78).

Tabel II.4. Tabel Bentuk Normal Keempat (4NF)

Pegawai Proyek

peg#	Pry#
1211	P1
1211	P3

Pegawai Ahli

Peg#	Ahli
1211	Analisis
1211	Perancangan
1211	Pemrograman

6. Bentuk Normal Kelima

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 78).

Tabel II.5. Tabel Bentuk Normal Kelima (5NF)

peg#	Pry#	Ahli
1211	11	Perancangan
1211	28	Pemrograman

II.2.9. UML *(Unified Modeling language)*

Unified Modeling Language (*UML*), adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini

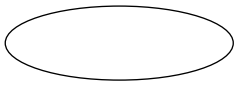
disebabkan karena *UML* menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi *(sharing)* dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).Pemodelan *visual* adalah salah satu cara berpikir tentang persoalan menggunakan model-model yang diorganisasikan seputar dunia nyata yang berguna untuk memahami persoalan, mengkomunikasikan dengan orang-orang yang terlibat dalam proyek (*costumer*, ahli dibidangnya, analisis, desainer dan lain-lain). Serta didefenisikan sebagai proses pemodelan sistem informasi menggunakan pengaturan standar elemen grafik dan objek-objek dalam sistem dan antar sistem itu sendiri. (Munawar; 2018: 49).

1. *Use Case* Diagram

   Use Case diagram merupakan pemodelan untuk kelakuan (*behavior)* sistem informasi yang akan dibuat. *Use Case*mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *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.6. 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.

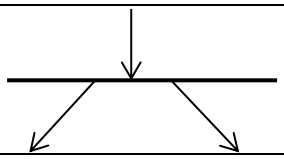
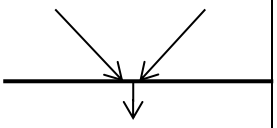
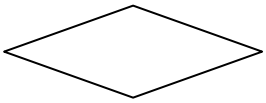
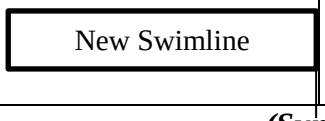
(Sumber : Munawar; 2018: 89)

2. Diagram Aktivitas (*ActivityDiagram*)

   Activity diagram menggambarkan *Workflow* (Aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *Activitydiagram*, yaitu :

Tabel II.7. Simbol Diagram Aktivitas

Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.

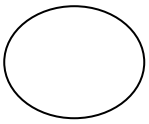
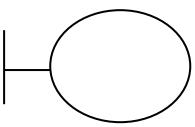
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true*, *false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

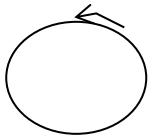
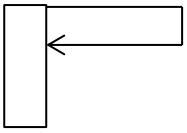
(Sumber : Munawar ; 2018 : 137)

3. Diagram Urutan *(Sequence Diagram)*

Diagram urutan menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam diagram urutan, yaitu

Tabel II.8. 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 : 186)