



BAB II

TINJAUAN PUSTAKA

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan:

1. Nurmalina, (2019) Dengan Judul “Perancangan Aplikasi Keamanan Pesan Teks Dengan Menggunakan Algoritma *Triple DES*” pada Teknik Informatika Kaputama (JTIK). Masalah yang terjadi pada penelitian ini adalah pesan teks yang sangat rentan terhadap pencurian atau penyadapan oleh orang yang ingin mendapatkan informasi secara ilegal. Penelitian ini menghasilkan sebuah aplikasi yang dapat mengamankan pesan teks dengan algoritma *Triple DES* menggunakan 3 kunci dan dengan adanya pengamanan maka pesan teks atau informasi dapat terjaga dan terjamin keasliannya.
2. Joko, dkk, (2016) dengan judul “Aplikasi Enkripsi Dan Dekripsi Untuk Keamanan Dokumen Menggunakan *Triple DES* Dengan Memanfaatkan Usb *Flash Drive*” pada jurnal *Coding*, Sistem Komputer Untan. Masalah yang terjadi pada penelitian ini adalah pengamanan data berbentuk dokumen yang ingin diamankan dengan menggunakan *Triple Des* agar pengamanan dokumen menjadi lebih terjaga. Penelitian ini menghasilkan sebuah *software* yang dapat

mengenkripsi dan mendekripsi dokumen dan memanfaatkan *USB Flash Drive* sebagai salah satu kunci dalam penggunaan aplikasi *3DES*.

3. Suroso, (2018) dengan judul “Studi Perbandingan Kriptografi Menggunakan Metode *DES*, *Triple DES* Dan *Rsa*” pada Jurnal Teknologi Pelita Bangsa. Yang menjadi masalah dalam penelitian ini adalah data-data atau informasi yang terlalu mudah untuk diakses siapa saja yang mengakibatkan data itu sendiri menjadi rusak atau sudah tidak asli lagi. Penelitian ini menghasilkan sebuah perbandingan dalam pengamanan data dengan metode *DES*, *Triple DES*, dan *RSA*, perbandingan ini dilakukan dalam hal lama proses dekripsi antara algoritma *RSA* dan *Triple DES*.
4. Hidayat dan Setiana, (2018) dengan judul “Perbandingan Waktu Dan Kecepatan Proses Enkripsi Dan Dekripsi Data Teks.Txt Menggunakan Algoritma *DES* Dan *3DES*” pada Jurnal Siliwangi. Masalah yang terdapat dalam penelitian ini adalah perbandingan Algoritma *Des* dan *Triple Des* yang menghasilkan sebuah perbandingan bahwa penggunaan algoritma *3DES* dianggap lebih aman dibandingkan dengan algoritma *DES* karena tingkat kerahasiaan algoritma *3DES* terletak pada panjangnya kunci yang digunakan.
5. Syafmi, dkk, (2019) dengan judul “Analisa Keamanan *Url* Yang Menggunakan Algoritma *3DES*” pada KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer). Masalah yang terdapat pada penelitian ini adalah jika resource atau value dari parameter dari suatu URL terlihat jelas dan disadari oleh pengguna yang berniat untuk mencuri atau merubah data, maka itu dapat menjadi suatu masalah keamanan website. Masalah dari keamanan url tersebut

menjadi celah yang paling sering digunakan peretas untuk melakukan tindakan kejahatan terhadap *Website*. Penelitian ini menghasilkan algoritma *3DES address bar* akan berupa hasil enkripsi yaitu ciphertext. Dengan menerapkan algoritma *3DES* pada pembangkitan *URL* akan memberikan keamanan pada *URL* tersebut, karena *value* dari parameter *URL* tidak bisa dimengerti oleh pengguna.

II.2. Pengertian Rancang Bangun

Menurut Devy (2018), rancang bangun adalah suatu kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada. Sedangkan menurut Andi, dkk (2018) rancang bangun adalah program yang menentukan aktifitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai atau pengguna komputer.

Dengan demikian, penulis menyimpulkan pengertian dari rancang bangun adalah suatu kegiatan menciptakan sistem ataupun memperbaiki sistem yang sudah ada untuk menyelesaikan tugas-tugas khusus dari pemakai atau pengguna.

II.3. Aplikasi

Menurut Helmi, dkk (2018) aplikasi merupakan penerapan, menyimpan suatu informasi, data, permasalahan pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk diterapkan menjadi sebuah bentuk yang baru. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya. Dengan demikian, penulis menyimpulkan pengertian aplikasi adalah suatu perangkat lunak komputer yang

menyimpan suatu informasi, data, permasalahan pekerjaan yang kemudian diterapkan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

II.4. Keamanan Data

Secara umum data dikategorikan menjadi dua, yaitu data yang bersifat rahasia dan data yang tidak bersifat rahasia. Data yang tidak bersifat rahasia biasanya tidak akan terlalu diperhatikan. Yang sangat perlu diperhatikan adalah data yang bersifat rahasia, dimana setiap informasi yang ada didalamnya akan sangat berharga bagi pihak yang membutuhkan karena data tersebut dapat dengan mudah digandakan. Untuk mendapatkan informasi didalamnya, biasanya dilakukan berbagai cara yang tidak sah (Angga dan Desi, 2018).

II.5. Algoritma *Triple-DES*

Menurut Elvara, dkk (2018) Algoritma *Triple-DES (Triple Data Encryption Standard)* adalah salah satu *system* enkripsi berlapis tiga dari sistem yang sebelumnya sudah ada, yaitu *Data Encryption Standard (DES)*. *3DES* lebih aman dari *DES*, karena mengalami enkripsi tiga kali. Pada awalnya, ukuran kunci sandi *DES* yaitu 56bit sudah mencukupi pada saat algoritma ini dibuat. Namun, dengan meningkatnya kemampuan komputasi, serangan brutal telah mungkin terjadi. *3DES* menyediakan metode yang sederhana dengan menambah ukuran kunci *DES* untuk mencegah serangan tersebut, tanpa memerlukan sandi blok (*block cipher*) yang sama sekali baru.

II.6. Microsoft Visual Studio

Microsoft Visual Studio adalah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi

yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Microsoft Visual Studio* (yang sering juga disebut dengan *VB.Net*) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis *windows* (Yudhi dan Gilang, 2018).

II.7. *My SQL*

MySQL adalah sistem manajemen *Database SQL* yang bersifat *Open Source* dan paling populer saat ini. Sistem Database *MySQL* mendukung beberapa fitur seperti *multithreaded*, multiuser dan *SQL Database managemen system (DBMS)* (Fitri dan Nia, (2018).

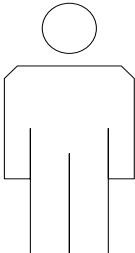
II.8. *Unified Modeling Language (UML)*

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

II.8.1 *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-Simbol *Use Case Diagram*

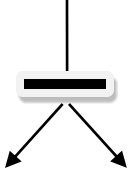
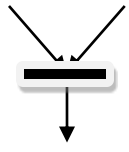
Simbol	Dekripsi
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja
	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi 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 <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

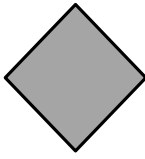
Sumber: (Hendini, 2016)

II.8.2 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.2 Simbol-Simbol Activity Diagram

Simbol	Dekripsi
	Start Point, diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	End Point, akhir aktivitas
	Activities, menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Join (penggabungan) atau <i>merge</i> , digunakan untuk menunjukkan adanya dekomposisi.

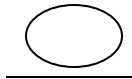
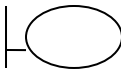
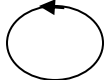
	<i>Decision Points</i>, menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>
---	--

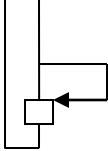
Sumber: (Hendini, 2016)

II.8.3 Sequence Diagram

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

Tabel II.3 Simbol-Simbol *Sequence Diagram*

Simbol	Dekripsi
	<i>Entity Class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas yang membentuk gambaran awal sistem dan menjadi landasan menyusun basis data.
	<i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>.
	<i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>, simbol mengirim pesan antar <i>class</i>.

	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

Sumber: (Hendini, 2016)

II.8.4 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 *Associations*, *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 Simbol-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 mempunyai arti minimal 2 maksimal 4

Sumber: (Hendini, 2016)