BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian yang dilakukan oleh Agusta,S.A dan Fitri,T.A (2016) dengan judul Implementasi *Algorithma Stream Cipher* RC4 dalam Aplikasi Pendataan Alumni STMIK Amik Riau. Agusta,S.A dan Fitri,T.A menyimpulkan bahwa data yang dienkripsi dengan algoritma *stream cipher* RC4 berada dalam kondisi aman saat tersimpan pada *database* dan memiliki tingkat efisiensi yang baik dalam penyimpanan data pada *database*, karena hasil enkripsi yang dihasilkan sama jumlahnya dengan karakter aslinya.

Penelitian yang dilakukan oleh Jumrin, dkk (2016) dengan judul Aplikasi Sistem Keamanan Basis Data Dengan Teknik Kriptografi RC4 *Stream Cipher*. Jumrin, dkk, menyimpulkan bahwa untuk proses kriptografi menggunkan metode RC4 (*Rivest Code*) *stream cipher*, hasil dari proses enkripsi dan deskripsi menunjukkan *file* yang telah dienkripsi (*Ciphertext*) kembali seperti *file* basis data aslinya (*plaintext*). Serta untuk proses enkripsi dan dekripsi jauh lebih cepat karena menggunakan kunci yang sama serta memiliki tingkat keamanan yang tinggi sesuai panjang kunci.

Penelitian yang dilakukan oleh Susanto (2017) dengan judul Implementasi Keamanan Data Sistem Informasi *Inventory Stock* Barang PT. Wings Food Menggunakan Algoritma *Rivest Code* 4 (RC4). Susanto menyimpulkan bahwa data yang di simpan di dalam *database* pada sistem informasi *inventory stock*

barang pada distributor PT. Wings Food berhasil di enkripsi sehingga terjamin kerahasiaan dan keamanannya. Sistem informasi *inventory stock* barang pada pada distributor PT. Wings Food juga telah berhasil mendekripsi data yang sebenarnya pada halaman tampil data ataupun laporan.

II.2. Landasan Teori

Berikut ini adalah landasan teori yang digunakan pada penelitian ini. beberapa landasan teori berikut dikutip dari beberapa referensi yang berkaitan dengan penelitian :

II.2.1. Aplikasi

Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanankan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan maupun pemrosesan data yang diharapkan. (Andi Juansyah, 2015 : 2).

II.2.2. Kriptografi

Kriptografi (*Cryptography*) berasal dari bahasa yunani, yaitu dari kata *crypto* dan *graphia* yang berarti penulisan rahasia. Kriptografi adalah suatu ilmu yang mempelajari penulisan pesan secara rahasia. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptopology*. Teminologinya,

kriptografi merupakan ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang berbeda. (Agusta dan Fitri, 2016 : 2).

II.2.3. Keamanan Data

Keamanan data adalah perlindungan informasi dari berbagai macam ancaman agar menjamin kelanjutan usaha/bisnis, mengurangi resiko bisnis dan meningkatkan *return of investment* dan peluang bisnis. Keamanan sistem informasi merupakan suatu kegiatan perlindungan atau pencegahan terhadap gangguan penyalahgunaan informasi yang dilakukan oleh orang-orang yang tidak bertanggung jawab terhadap jalannya suatu sistem. (Adi Supriyatna, 2014).

II.2.4. Algoritma RC4

Algoritma RC4 mengenkripsi antara kombinasi *plaintext* dengan menggunakan *bit-wise* Xor *(Exclusive-or)*. RC4 menggunakan panjang kunci dari 1 sampai 256 *byte* yang digunakan untuk menginisialisasikan tabel sepanjang 256 *byte*. Tabel ini digunakan untuk generasi yang berikut dari *pseudo random* yang menggunakan XOR dengan *plaintext* untuk menghasilkan *ciphertext*. Masing - masing elemen dalam tabel saling ditukarkan minimal sekali. Proses dekripsinya dilakukan dengan cara yang sama (karena *Xor* merupakan fungsi simetris).

Untuk menghasilkan *keystream*, *cipher* menggunakan *state* internal yang meliputi dua bagian :

1. Tahap *key scheduling* dimana *state automaton* diberi nilai awal berdasar kan kunci enkripsi. *State* yang diberi nilai awal berupa *array* yang merepresentasikan suatu permutasi dengan 256 elemen, jadi hasil dari

algoritma KSA adalah permutasi awal. *Array* yang mempunyai 256 elemen ini (dengan indeks 0 sampai dengan 255) dinamakan S. Berikut adalah algoritma KSA dalam bentuk *pseudo-code* dimana *key* adalah kunci enkripsi dan *keylength* adalah besar kunci enkripsi dalam *bytes* (untuk kunci 128 bit, keylength = 16).

```
for i = 0 to 255
  S[i] := i
j := 0
for i = 0 to 255
  j := (j + S[i] + key[i mod keylength]) mod 256
  swap(S[i],S[j])
```

2. Tahap *pseudo-random generation* dimana *state automaton* beroperasi dan outputnya menghasilkan *keystream*. Setiap putaran, bagian *keystream* sebesar 1 *byte* (dengan nilai antara 0 sampai dengan 255) di *output* oleh PRGA berdasarkan *state* S. Berikut adalah algoritma PRGA dalam bentuk *pseudo-code*.

```
i := 0
j := 0
loop
  i := (i + 1) mod 256
  j := (j + S[i]) mod 256
  swap(S[i],S[j])
  output S[(S[i] + S[j]) mod 256]
```

Setelah terbentuk *keystream,* kemudian *keystream* tersebut dimasukkan dalam operasi XOR dengan *plaintext* yang ada, dengan sebelumnya dokumen dipotong-potong terlebih dahulu menjadi *byte-byte*. (Okie Setiawan, 2014 : 117).

II.2.5. *Java*

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. (Maria, 2015 : 25).

II.2.6. MySQL

MySQL merupakan *software* yang tergolong *database* server dan bersifat *open source*. MySQL juga bersifat *multiplatform*, dapat dijalankan pada berbagai sistem operasi. Pengaksesan data dalam *database* dapat dilakukan dengan mudah melalui SQL (*Structure Query Languange*). Data dalam *database* bisa diakses melalui aplikasi *non-web* (*Netbeans*) maupun aplikasi web (*PhpMyAdmin*). (Maria, 2015 : 25).

II.2.7. Data Nasabah

Data nasabah adalah data atau identitas yang wajib disediakan oleh nasabah kepada pihak tertentu sebagai contoh pihak bank dalam rangka melakukan transaksi keuangan. Data nasabah sering berubah karena sifatnya bergerak, sehingga menyebabkan perubahan nama, nomor identitas, alamat dan nomor telepon. Tingkat perubahan yang cepat dari data nasabah tersebut menimbulkan masalah pada kualitas data. (Agustiadi, 2016).

II.2.8. UML

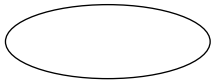
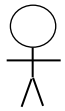
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan *system*. (Ade Hendini, 2016 : 108).

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

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*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* dapat dilihat pada tabel II.1 sebagai berikut :

Tabel II.1. *Use Case Diagram*

Simbol	Nama	Keterangan
	*Use Case*	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
	*Actor*	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*.

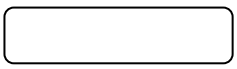
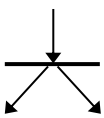
	Association	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Association	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

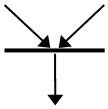
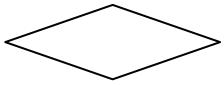
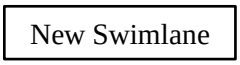
(Sumber : Ade Hendini : 2016 : 108)

2. *Activity Diagram*

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 sebagai berikut :

Tabel II.2. *Activity Diagram*

Simbol	Nama	Keterangan
	Start point	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	End point	*End point*, akhir aktivitas.
	*Activites*	*Activites*, menggambarkan suatu proses /kegiatan bisnis.
	*Fork*	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan pararel menjadi satu.

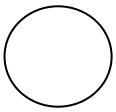
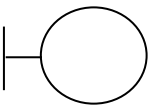
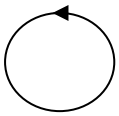
	*Join*	*Join* (penggabungan), digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true*, *false*.
	*Swimlane*	*Swimlane*, untuk menunjukkan siapa melakukan apa.

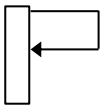
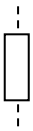
(Sumber : Ade Hendini : 2016 : 109)

3. *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* dapat dilihat pada tabel II.3 sebagai berikut :

Tabel II.3. *Sequence Diagram*

Simbol	Nama	Keterangan
	*Entity Class*	*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*	*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*	*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	*Message*, simbol mengirim pesan antar *class*.

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

(Sumber : Ade Hendini : 2016 : 110)

4. *Class Diagram*

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.

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 maksimum 4

(Sumber : Ade Hendini : 2016 : 111)