PENGEMBANGAN PESAN TEXT MENGGUNAKAN KRIPTOGRAFI UNTUK KEAMANAN DATA KONSUMEN PADA SHOWROOM MOBIL MITSHUBISHI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Menempuh Ujian Akhir
Memperoleh Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH :

NAMA	: ARIE AGUSTIONO
NPM	: 1414370227
PROGRAM STUDI	: SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

Abstrak

Pengembangan Pesan Text Menggunakan Kriptografi Untuk Keamanan Data Konsumen Pada Showroom Mobil Mitshubishi

Proses kombinasi algoritma Vigenere Cipher dan *One Time Pad* ini mendapatkan hasil penyandian pesan yang lebih baik. Sehingga dapat menjadi media pembelajaran bagi masyarakat atau akademika. Proses penyandian dalam penelitian ini dilakukan dengan empat tahapan, tahapan pertama yang mana pesan ditulis, tahap dua proses pembangkit kunci menggunakan LCG, tahap tiga proses enkripsi, dan tahapan terakhir adalah deskripsi menggunakan pembangkit kunci LCG yang digunakan pada saat enkripsi. menggunakan kombinasi kriptografi vigenere cipher dan *one time pad* agar dapat mengenkripsi dan dekripsi data teks yang akan di gunakan secara rahasia dan lebih mudah digunakan oleh *user* nantinya. Pemrograman desktop yang digunakan dalam penelitian ini mensimulasikan proses enkripsi dan deskripsi database pada data konsumen pada showroom mobil mitshubishi berjalan sesuai dengan tahapan penelitian yang dirancang.

Kata Kunci : *Kriptogarafi OneTimePad, Viginere Cipher, Linear Congruential Generator, OTP, LCG.*

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Keamanan data dan nformasi merupakan hal yang sangat penting di era informasi saat ini. Umumnya, setiap institusi memiliki dokumen-dokumen penting dan bersifat rahasia yang hanya boleh diakses oleh orang tertentu.Sistem informasi yang dikembangkan harus menjamin keamanan dan kerahasian dokumen-dokumen tersebut.Namun kendalanya bahwa media-media yang digunakan sering kali dapat disadap olehp ihak lain. Oleh karena itu, diperlukan metode untuk mengamankannya, salah satunya dengan menggunakan metode kriptografi.

Dalam penelitian ini, proses penyandian pesan masih berbasis dekstop ada di komputer kita.Diharapkan pada proses kombinasi kedua algoritma ini mendapatkan hasil penyandian pesan yang lebih baik. Sehingga dapat menjadi media pembelajaran bagi masyarakat atau akademika. Proses penyandian dilakukan empat tahap, tahap satu tuliskan pesan yang akan dikirim, tahap dua proses pembangkit kunci menggunakan LCG, tahap tiga proses enkripsi, dan tahapan terakhir adalah deskripsi menggunakan pembangkit kunci LCG yang digunakan pada saat enkripsi.

Aplikasi yang akan dibuat oleh penulis adalah dengan menggunakan *Microsoft Visul Studio* 2008 dengan menggunakan kriptografi *one time pad* agar dapat mengenkripsi dan dekripsi data teks yang akan di gunakan secara rahasia dan

lebih mudah digunakan oleh *use r*nantinya. Berdasarkan latar belakang di atas makapenulis tertarik untuk memilih judul "**Pengembangan Pesan Text Menggunakan Kriptografi Untuk Keamanan Data Konsumen Pada Showroom Mobil Mitshubishi**"**.**

1.2 Perumusan Masalah

Sesuai dengan latar belakang yang dipaparkan, maka masalah yang akan dibahas adalah sebagai berikut:

1. Bagaimana mengkombinasi kedua algoritma *Vigener cipher* dengan *Dynamic Key Linear Congruential Generator (LCG)* yang di implementasikan pada *Three-pass protocol*?
2. Bagaimana menerapkan algoritma *Vigener cipher* dengan *Dynamic Key Linear Congruential Generator (LCG)*?

1.3 Batasan Masalah

Adapun batasan masalah yang dibatasi dari penulisan skripsi ini adalah sebagai berikut:

1. Metode yang di gunakan pada perancangan aplikasi pengamanan informasi ini menggunakan metode *one time pad* untuk ekripsi dan dekripsi *text*.
2. Bahasa program yang digunakan dalam perancangan aplikasi kriptografi *one time pad* ini adalah *Microsoft Visual Studio 2008* dan *database MySQL*.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai penulis dalam perancangan aplikasi kriptografi *one time pad* ini adalah :

1. Untuk mengubah pengiriman data.
2. Membuat suatu aplikasi kriptografi yang mengimplementasikan algoritma *One Time Pad* sehingga dapat mengatasi masalah keamanan informasi serta menjaga kerahasiaan data.

1.5 Manfaat Penelitian

Perancangan aplikasi kriptografi *on time pad* ini bermanfaat bagi masyarakat luas antara lain :

1. Memberikan pengetahuan terhadap proses pembuatan *One Time Pad* dengan menggunakan algoritma *Dynamic Key Linear Congruential Generator (LCG)* dengan *Vigenere cipher.*
2. Memberikan penjelasan tentang *One Time Pad* dalam implementasi pada aplikasi dekstop.

BAB II

LANDASAN TEORI

2.1 Keamanan Data

Secara umum data dikategorikan menjadi data yang bersifat rahasia dan data tidak bersifat rahasia. Data tidak bersifat rahasia biasanya tidak terlalu penting. Data yang penting adalah data yang bersifat rahasia, dimana setiap informasi yang ada didalamnya sangat berharga bagi pihak yang membutuhkan karena data tersebut dapat dengan mudah digandakan. Untuk mendapatkan informasi didalamnya, biasa dilakukan berbagai cara yang tidak sah sehingga dibutuhkan suatu sistem yang dapat menjamin keamanan data(Harahap, 2019).

Keamanan suatu data terkait dengan hal-hal sebagai berikut :

a. Fisik. Dalam hal ini pihak yang tidak berwenang terhadap data berusaha untuk mendapatkan data dengan melakukan sabotase atau penghancuran tempat penyimpanan data.
b. Organisasi. Dalam hal ini pihak yang tidak berwenang terhadap data berusaha untuk mendapatkan data melalui kelalaian atau kebocoran anggota yang menangani data.
c. Ancaman dari luar. Dalam hal ini pihak yang tidak berwenang terhadap data berusaha untuk mendapatkan data melalui media komunikasi dan melakukan pencurian data yang tersimpan didalam komputer.

Adapun fungsi keamanan dalam komputer adalah menjaga tiga karakteristik, yaitu:

a. *Secrecy*, adalah isi program komputer yang hanya dapat diakses oleh orang yang berhak. Termasuk *reading*, *viewing*, *printing*, atau mengetahui keberadaan sebuah objek.
b. *Integrity*, adalah isi komputer yang dapat dimodifikasi oleh orang yang berhak. Termasuk *writing*, *changing* status, *deleting*, dan *creating*.
c. *Availability*, adalah isi komputer yang tersedia untuk beberapa kelompok yang diberi hak.

2.2 Kriptografi

Kriptografi telah menjadi bagian penting dalam dunia teknologi informasi, terutama dalam bidang komputer.Hampir semua penerapan teknologi informasi menggunakan kriptografi sebagai alat untuk menjamin keamanan dan kerahasiaan data atau informasi. Karena itu kriptografi menjadi suatu ilmu yang berkembang pesat dan dalam waktu singkat banyak muncul algoritma-algoritma baru yang dianggap lebih unggul dari pada algoritma pendahulunya(Pratiwi, Marwati, & Yusnitha, n.d.).

Hingga zaman modern seperti saat ini, kriptografi semata-mata dianggap sebagai enkripsi, yaitu proses mengubah data yang tidak biasa dan tidak dapat dibaca menjadi suatu informasi yang jelas dan dapat dibaca. Sedangkan dekripsi adalah proses sebaliknya. *Chipertext* tersebut adalah suatu pasangan algoritma yang melakukan enkripsi dan membalikkan dekripsi. Informasi detail dari *chipertext* dikontrol oleh algoritma tersebut, dengan kata lain dengan suatu kunci. Hal tersebut merupakan parameter rahasia untuk membaca pesan rahasia tersebut dan biasanya hanya pengirim dan yang dikirim yang mengetahui kunci tersebut.

Kunci tersebut amatlah penting karena tanpa kunci itu, pesan tersebut akan mudah terbongkar dan menjadi tidak berarti lagi. Berdasarkan sejarahnya, chipertext kadang kalau digunakan langsung untuk mengenkripsi atau dekripsi tanpa prosedur tambahan seperti pengesahan dan pengecekan kepribadian(Penulis & Rickson, n.d.).

Dalam bahasa sehari-hari, kode biasanya digunakan untuk mengartikan suatu metode enkripsi atau menyembunyikan suatu makna.Tetapi, dalam kriptografi, kode memiliki arti spesifik lebih, berarti suatu pergantian dari suatu unit dari suatu informasi dengan kata kode (sebagai contoh, apple pie diganti dengan *attack at dawn*). Kode tidak digunakan lagi dalam kriptografi yang sesungguhnya kecuali tidak sengaja seperti proses desain suatu unit (contoh '*Bronco Flight*' atau *Operation Overlord*) sejak *chipertext* yang dipilih lebih praktis dan lebih aman dari biasanya, serta lebih mudah disesuaikan dengan komputer. Beberapa penggunaan kriptografi dan kriptologi dapat saling bertukar tempat dalam bahasa Inggris, ketika penggunaan kriptografi yang lain mengarah ke penggunaan dan praktek dari teknik kriptografi, dan kriptologilebih mengarah ke *subjek* sebagai studi lapangan. Kriptografi di Indonesia disebut persandian yaitu secara singkat dapat berarti seni melindungi data dan informasi dari pihak-pihak yang tidak dikehendaki baik saat ditransmisikan maupun saat disimpan. Sedangkan ilmu persandiannya disebut kriptologi yaitu ilmu yang mempelajari tentang bagaimana tehnik melindungi data dan informasi tersebut beserta seluruh ikutannya(Firdaus, Marwati, & Sispiyati, n.d.).

2.2.1 Definisi Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani yang terdiri dari dua suku kata, yaitu *cryptós* yang berarti rahasia dan *gráphein* yang berarti kata tulisan. Karena itu secara umum kriptografi diartikan sebagai tulisan rahasia(Applications, 2018).

Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan (Cryptography is the art and science of keeping messages secure) selain itu ada pengertian tentang kriptografi yaitu kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata "seni" di dalam definisi di atas maksudnya adalah mempunyai cara yang unik untuk merahasiakan pesan. Kata "graphy" di dalam "cryptography" itu sendiri sudah menyiratkan sebuah seni(Nasution, Teknik, Sekolah, Teknik, & Medan, n.d.).

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Atau dalam defenisi yang lainnya kriptografi adalah seni dan ilmu dalam mengamankan pesan.Dalam arti lain, Kriptografi adalah ilmu dan seni yang mempelajari tentang merahasiakan pesan atau informasi kedalam suatu bentuk yang tidak dapat dimengerti sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak berhak(Diani & Widhiyasana, 2018).

Terdapat beberapa definisi kriptografi dalam berbagai literatur. Definisi pada tahun 80-an menyatakan kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Kata seni dalam definisi ini berasal dari fakta sejarah bahwa

pada awal sejarah kriptografi, setiap orang mempunyai cara yang unik untuk merahasiakan pesan(Diani & Widhiyasana, 2018).

Sedangkan definisi dalam buku-buku terbaru menyatakan kriptografi merupakan ilmu mengenai metode untuk mengirimkan pesan secara rahasia sehingga hanya penerima yang dimaksud yang dapat menghapus dan membaca pesan tersebut atau memahaminya. Pengertian lain kriptografi yaitu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata *graphy* dalam kata *cryptography* itu sendiri sudah menyiratkan sebuah seni(Nasution et al., n.d.).

Jadi, kriptografi adalah suatu ilmu sekaligus seni yang bertujuan untuk menjaga keamanan suatu pesan (*cryptography is the art and science of keeping messages secure*). Secara umum, kriptografi adalah teknik pengamanan informasi dimana informasi diubah dengan kunci tertentu melalui enkripsi sehingga menjadi informasi baru yang tidak dapat dimengerti oleh orang yang tidak berhak menerimanya, dan informasi tersebut hanya dapat diubah kembali oleh orang yang berhak menerimanya melalui dekripsi(Applications, 2018).

Untuk dapat menjalankan dengan baik pada proses kriptografi haruslah terdapat emapat elemen utama didalamnya, yang paling berkait satu sama lain. Yaitu (Firdaus et al., n.d.):

1. *Plain Text*

Merupakan sebagai pesan awal atau pesan asli yang di kirim pada proses komunikasi. Plain Text inilah yang kemudian di enkripsi dan di deskripsi.

2. *Cipher Text*

   Merupaka pesan yang tersembunyi, yaitu pesan asli (*Pain Text*) yang telah di enkripsi dapa proses kriptografi. *Cipher Text* ini dapat diubah kembali kebentuk aslinya (*Pain Text*) memanfaatkan *Key* yang telah di sediakan.

3. *Cryptography Key*

   Merupakan kunci yang di gunakan untuk melakukan enkripsi dan deskripsi pada proses kriptografi. Tanpa adanya kunci (*key*) yang sama maka proses enkripsi dan deskripsi tidak dapat dilakukan dengan baik. Kunci (*key*) merupakan informasi yang padat menjadi kendali terhadap proses terjadinya kriptografi.

Jenis serangan berdasarkan cara dan posisi seseorang untuk mendapatkan pesan-pesan dalam jaringan, yaitu (Penulis & Rickson, n.d.):

1. *Sniffing*

   Sniffing berarti ‘mengendus’, dalam hal ini yang diendus merupakan pesan (baik yang belum ataupun yang sudah di enkripsi) dalam suatu saluran komunikasi. Hal ini umum terjadi pada saluran publik yang tidak aman yang mengakbatkan sang pengendus dapat merekam pembicaraan yang terjadi.

2. *Replay Attack*

Replay Attack adalah serangan jaringan dimana penyerang menyadap percakapan antara pengirim dan penerima, serta mengambil informasi autentik dengan berbagi kunci.Penyerang kemudian menghubungi penerima dengan kunci itu sebagai bukti identitas dan keaslian untuk menipu penerima.Misalkan Yudhit mencuri informasi yang dikirimkan Bagas ke Salsa selanjutnya mengubah pesan tersebut sebelum dikirim kembali ke Salsa seolah pesan tersebut asli dari Bagas.

3. *Spoofing*

   Spoofing adalah teknik untuk memperoleh akses yang tidak sah ke suatu komputer atau informasi dimana penyerang berhubungan dengan pengguna dengan berpura-pura sebagai *host* yang dapat dipercaya.Spoofing biasanya dilakukan oleh seorang *hakcer*/*craker*.Sebagai contoh, Seorang penyerang (misalnya Yudhit) bisa menyamar menjadi Bagas.Semua orang dibuat percaya bahwa Yudhit adalah Bagas.Penyerang berusaha meyakinkan kepada pihak-pihak lain bahwa tidak ada yang salah dengan komunikasi yang dilakukan.Padahal komunikasi itu dilakukan dengan penipu/penyerang.

4. *Man-in-the-middle*

   Jika *spoofing* kadang hanya menipu satu pihak, dalam skenario ini saat Bagas hendak berkomunikasi dengan Salsa, Yudhit dimata Bagas seolah-olah menjadi Salsa. Yudhit juga dapat menipu Salsasehingga ia seolah-olah adalah Bagas. Yudhit dapat berkuasa penuh atas jalur komunikasi dan bisa membuat berita fitnah.

Kriptografi bertujuan untuk memberi layanan keamanan, yang dinamakan aspek - aspek keamanan yaitu(Pratiwi et al., n.d.):

1. Kerahasiaan adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak – pihak yang tidak berhak.
2. Integritas data adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.
3. Otentikasi adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak – pihak yang berkomunikasi.
4. Non-repudiation adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan.

2.2.2 Sejarah Kriptografi

Kriptografi dimulai pertama sekali dengan metode pertukaran posisi untuk mengenkripsi suatu pesan tertentu. Dalam perkembangannya, dikatakan bahwa Julius Caesar dalam mengirim pesan selalu mengacak pesan sebelum diberikan kepada para kurir. Karena itu ada pendapat bahwa yang dilakukan Julius Caesar dianggap sebagai awal mula dari penggunaan kriptografi. Namun sesungguhnya kriptografi telah digunakan untuk pertama kalinya oleh bangsa Mesir pada 4000 tahun lalu dan masih digunakan hingga kini(Harahap, 2019).

Saat ini kriptografi masih diperbincangkan secara luas karena kriptografi dapat digunakan sebagai suatu alat untuk melindungi kerahasiaan dan strategi negara. Sejarah kriptografi sebagian besar merupakan kriptografi klasik, yaitu metode enkripsi yang menggunakan kertas dan pensil atau dengan bantuan alat mekanik sederhana(Pratiwi et al., n.d.).

Secara umum algoritma kriptografi klasik dikelompokkan dalam dua kategori, yaitu *transposition cipher* dan *substitution cipher*. *Transposition cipher* mengubah susunan huruf-huruf yang ada dalam pesan, sedangkan *substitution cipher* mengganti setiap huruf atau kelompok huruf yang ada dalam pesan dengan huruf atau kelompok huruf lain(Pratiwi et al., n.d.).

Kriptografi klasik mencatat penggunaan algoritma transposition cipher oleh tentara Sparta di Yunani pada awal tahun 400 SM saat mereka menggunakan suatu alat bernama scytale yang terdiri dari sebuah kertas panjang dari daun papyrus yang dililitkan pada sebuah selinder berdiameter tertentu yang menyatakan kunci penyandian pesan. Pesan kemudian ditulis secara horizontal, baris per baris. Bila pita dilepaskan, huruf-huruf yang ada didalamnya telah tersusun secara acak membentuk pesan rahasia. Untuk membaca pesan, penerima pesan harus melilitkan kembali kertas tersebut pada selinder berdiameter sama dengan diameter selinder pengirim (Pratiwi et al., n.d.).

Sedangkan penggunaan substitution cipher yang paling awal dan paling sederhana adalah Caesar Cipher yang digunakan raja Yunani kuno, yaitu Julius Caesar. Caranya dengan mengganti setiap karakter dalam alphabet dengan karakter yang terletak pada tiga posisi berikutnya dalam susunan alphabet yang digunakan(Penulis & Rickson, n.d.).

2.2.3 Terminologi Dalam Kriptografi

Dalam kriptografi akan sering dijumpai beberapa istilah atau terminology sebagai berikut:

1. Pesan, *plaintext* dan *ciphertext*

Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext* atau teks jelas (*cleartext*). Agar pesan tidak dimengerti oleh pihak lain yang tidak berkepentingan, maka pesan perlu disandikan menjadi bentuk lain yang tidak dapat dipahami. Bentuk pesan tersandi disebut *ciphertext* atau *cryptogram*. Cipherteks harus dapat ditransformasi kembali menjadi plainteks. Sebagai contoh plainteks, uang disimpan di balik buku X, maka cipherteksnya adalah j&kloP#d$gkh*7h^”tn%6^klp..t@(Firdaus et al., n.d.).

2. Pengirim dan penerima

   Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan. Entitas dapat berupa orang, mesin, kartu kredit dan sebagainya.

3. Enkripsi dan Dekripsi

   Proses menyandikan *plainteks* menjadi *chiperteks* disebut enkripsi (*encryption*). Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* disebut dekripsi (*decryption*).

4. Algoritma Kriptografi dan Kunci

   Algoritma kriptografi disebut juga *chiper* yaitu aturan untuk *enchipering* dan *dechipering*, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Kriptografi modern mengatasi masalah keamanan algoritma kriptografi dengan penggunaan kunci. Kunci (*key*) adalah parameter yang

digunakan untuk transformasi *enchipering* dan *dechipering*. Kunci biasanya berupa string atau deretan bilangan.

5. Sistem Kriptografi

   Kriptografi membentuk sebuah sistem yang dinamakan sistem kriptografi. Menurut Schneier dan Munir, system kriptografi (*cryptosystem*) adalah kumpulan yang terdiri atas algoritma kriptografi, semua *plainteks* dan *chiperteks* yang mungkin serta kunci.

6. Kriptanalisis dan Kriptografi

   Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan *chiperteks* menjadi *plainteks* tanpa mengetahui kunci yang diberikan. Pelakunya disebut kriptanalis. Kriptologi (*cryptology*) adalah studi mengenai kriptografi dan kriptanalisis.

2.3 Vigenere Chiper

Pada saat itu belum dapat menyelesaikan algoritma vigenere chiper sampai akhirnya dipecahkan oleh Friedman dan Kasiski sekitar tahun 1917. Salah satu kelemahan dari algoritma ini adalah jika panjang kuncidimasukkan oleh pengguna lebih kecil dari plaintext, kunci berikutnya adalah pengulangan kunci pengguna, iniakan memungkinkan untuk histogram, Kasiski menemukan cara untuk mendapatkan panjang kunci pengguna awal darihistogram(Series & Science, 2018).

Modifikasi dari Vigenere cipher dimungkinkan untuk membuat algoritma ini lebih baik, tetapi efektivitas modifikasi yang dilakukan tidak selalu lebih baik dari pada

Vigenere standar, dalam penelitian meninjau beberapa modifikasi yang telah dibuat dan menyimpulkan bahwaentropi dan Indeks kebetulan modifikasi yang diusulkan tidak meningkat secara signifikan,bahkan ada subset dari vigenere standar itu sendiri.

Pada tahun 2010 dan 2011 penelitian yang melibatkan sandi Vigenere telah dilakukan dan dikelolamemperoleh efek longsoran jauh lebih baik daripada vigenere standar, tetapi dalam studi tersebutkombinasi dari banyak algoritma klasik dan modern, penelitian ini mungkin efektif tetapi tidak efisien,karena kesan Vigenere sederhana hilang(Series & Science, 2018).

Vigenere Cipher adalah algoritma klasik dan tentu saja algoritma simetris. Dalam proses enkripsi dan dekripsi menggunakan *tabula recta*, sebuah matriks 26 x 26 yang berisi huruf alphabet(Gambar 2.1), di mana *ciphertext* adalah huruf alfabet perpotongan antara *Plaintext*alfabet dan alfabet kunci.

		-- PLAINTEXT --					
		A	B	C	D	...	Z
K	A	A	B	C	D	...	Z
	B	B	C	D	E	...	A
E	C	C	D	E	F	...	B
	D	D	E	F	G	...	C
Y	:	:	:	:	:	...	:
	Z	Z	A	B	C	...	Y

Gambar 2.1. VigenereTabula *recta* 26 x 26
Sumber :(Handoyo & Teknik, 2013)

Secara matematis, proses enkripsi dan dekripsi dapat dilihat sebagai berikut persamaan:

$Ci = (Pi + Ki)\ mod\ 26$

$Pi = (Ci - Ki)\ mod\ 26$

Di mana C adalah teks sandi yang dihasilkan dari proses Enkripsi E dengan menambahkan indeks alfabet *Plaintext* P dengan Key K dimodulasi dengan 26, dan sebaliknya *Plaintext* P dihasilkan dari proses Dekripsi D dengan mengurangi abjad *Cipher teks*indeks dengan kunci K dan juga dimodulasi dengan 26.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 2.2 Tabel Vigenere Chiper
Sumber : (Series & Science, 2018)

Untuk melakukan enkripsi perlu menentukan terlebih dahulu kunci yang akan digunakan. Panjang kunci ini dapat bervariasi tergantung dari kesepakatan anatara kedua belah pihak yang saling berkomunikasi.Setelah menentukan kunci yang digunakan, maka substitusi dilakukan menggunakan Tabel Vigenere Cipher 2.2. Pada setiap step enkripsi dari alphabet, *cipher* yang dihasilkan menggunakan indeks alphabet yang berbeda-beda sesuai kunci. Bila kunci < panjang plainteks, maka kunci tersebut akan digunakan berulangulang. Proses dekripsi dilakukan merujuk pada baris kunci ke-i, kemudian cari pada kolom apa cipher ke-i tersebut muncul. Indeks kolom tersebut merupakan plainteks ke-i. Sebagai contoh proses enkripsi-dekripsi, misalkan plainteks yang akan dienkripsi adalah MAKALAH KRIPTOGRAFI dengan kunci yang telah ditetapkan bersama, yaitu VIGENERE. Kunci dibuat berulang-ulang sesuai dengan plainteks terlebih dahulu seperti berikut: Plainteks : MAKALAH KRIPTOGRAFI Key : VIGENER EVIGENEREVI Dari susunan tersebut, maka kita akan melakukan operasi penjumlahan sebagai berikut :

$$C[i] = (P[i] + K[i])\ mod\ 26$$
$$\text{dengan } i \leq length(P)$$

C: cipher teks
P: plainteks
K: kunci

Sehingga didapatkan cipher seperti berikut:

Chiper : HIQEYEY OMQVXBKIEAQ

$$P[i] = (C[i] - K[i])\ mod\ 26$$
$$\text{dengan } i \leq length(P)$$

2.4 One Time Pad

Algoritma *One Time Pad* (OTP) adalah algoritma kriptografi kunci simetris dengan enkripsi danproses dekripsi menggunakan kunci yang sama. Dalam penelitian ini, kami menggunakan 256 karakter dari tabel ASCII jadiakan melakukan proses enkripsi dan dekripsi di mod 256.

Rumus enkripsi OTP adalah

$C_i = (P_i + K_i) \bmod 256$

Rumus dekripsi AndOTP adalah

$P_i = (C_i - K_i) \bmod 256$

Keterangan :

C_i = Ciphertext

P_i = Plaintext

K_i = Key

Mod 255 yaitu total dari ASCII

2.5 *Linear Congruential Generator Algorithm*

Penelitian tentang keamanan data sangat menarik.Studi ini juga selalu dibahas untuk meningkatkan kenyamanan menggunakan data rahasia. Kriptografi tergantung pada kekuatan kunci, dengan menggunakan kunci yang kuat pada proses enkripsi dan dekripsi, kita dapat meningkatkan keamanan data rahasia(Wang & Wu, 2018).

Algoritma One Time Pad (OTP) dikarakterisasi, dalam hal kunci hanya digunakan sekali dalam setiap proses enkripsi. Algoritma OTP menggunakan angka acak yang dapat dibangun dari berbagai jenis algoritma penghasil angka acak.Kunci algoritma OTP akan dibuat sepanjang plaintext yang akan dienkripsi. Jika plaintext lebih panjang dari kunci, kunci akan diulang sepanjang plaintext. Karena kunci unik ini, algoritma OTP dapat dianggap sebagai algoritma paling kuat dalam kriptografi.Muhammad Iqbal et al mengimplementasikan algoritma OTP di aplikasi SMS untuk menjaga keamanan pesan rahasia. Operator XOR digunakan dalam proses enkripsi dan dekripsi. Pemilihan kunci dilakukan secara manual dengan memilih beberapa karakter yang akan dikonversi menjadi kode ASCII. Pesan dihasilkan dari aplikasi ini adalah pesan yang dapat dijamin kerahasiaannya karena pesan hanya dapat dibaca oleh pengirim dan penerima.Algoritma OTP adalah versi terbaru dari algoritma Vernam Cipher. Kunci yang digunakan dalam Algoritma OTP juga dapat dihasilkan.

algoritma bilangan acak Nomor acak pertama yang dirujuk disebut Seed, Jhessica Clawdia et al menggunakan algoritma Linear Congruential Generator (LCG) untuk menghasilkanangka acak. Algoritma ini digunakan untuk menghindari pembuatan kunci berulang(Wang & Wu, 2018).

Proses dapat dilihat bahwa semakin lama rentang nilai m semakin kecil kemungkinan kunci akan diulang. Menggunakan algoritma LCG untuk menghasilkan kunci, juga dapat menghasilkan kunci dinamis. Seperti yang telah dilakukan oleh Zeenat Mahmood et al tampak bahwa algoritma kriptografi sudah sulit dipecahkan pembacaan sandi.Algoritma generator angka acak memiliki peran

penting dalam kriptografi karena mereka memiliki resistensi terhadap serangan cryptanalysis. Mina Mishra et al ia melakukan penelitian tentang pengujian beberapa algoritma penghasil angka acak, salah satunya adalah LCG. Algoritma LCG memiliki ketahanan terhadap

serangan cryptanalysis dan memiliki sensitivitas kunci yang baik(Wang & Wu, 2018).

Berbagai kombinasi algoritma juga dilakukan untuk menghasilkan angka acak. Massoud Sokouti et al ia menghasilkan angka acak dengan algoritma genetika. Kunci yang telah yang terbentuk akan digunakan oleh algoritma OTP. Hasil penelitian ini, memiliki algoritma OTP yang kuat resistensi dan mampu menjaga keamanan data dari serangan cryptanalyst(Wang & Wu, 2018).

Selain algoritma penghasil angka acak LCG, ada juga Quadratic Congruential Algoritma Generator (QCG).Algoritma ini menghasilkan angka acak dengan rumus matematika dan pangkat kuadrat seperti yang ditulis oleh Jurgen Eichenauer et al.

Proses enkripsi dan dekripsi yang tidak melibatkan karakter plaintext pertama.

Ini adalah rumus Linear Congruential Generator :

$X_n = (a * X_{n-1} + b) \bmod m$

Deskripsi formula:

Xn = Nomor acak ke n seri

Xn-1 = angka acak sebelumnya

a = pengganda

c = kenaikan

m = modulus

2.6 Algoritma One Time Pad Dimodifikasi dengan LCG

Kriptografi adalah suatu ilmu atau seni mengamankan pesan dan dilakukan oleh *cryptographer*.Sedangkan *cryptanalysis* adalah suatu ilmu dan seni membuka (*breaking*)*ciphertext* dan orang yang melakukannya disebut *cryptanalyst*.Ditinjau dari terminologinya, kata kriptografi berasal dari bahasa Yunani, yaitu *kryptos*, yang berarti menyembunyikan, dan *graphein*, yang berarti menulis, sehingga kriptografi dapat didefinisikan sebagai ilmu yang mengubah informasi dari keadaan/bentuk normal (dapat dipahami) menjadi bentuk yang tidak dapat dipaham(Wang & Wu, 2018).

Secara sederhana, proses kriptografi dapat OTP adalah salah satu contoh metode kriptografi dengan algoritme jenis simetri, sehingga kunci yang digunakan untuk proses enkripsi sama dengan kunci yang digunakan untuk proses dekripsi.OTP adalah algoritme kriptografi yang diklaim sempurna. OTP (*pad* = kertas *blocknote*) berisi deretan karakter-karakter kunci yang dibangkitkan secara acak.Penerima pesan memiliki salinan (*copy*) *pad* yang sama. Satu *pad* hanya digunakan sekali (*one-time*) saja untuk mengenkripsi pesan. Sekali telah digunakan, *pad* dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain(Wang & Wu, 2018).

Suatu algoritma dikatakan aman apabila tidak ada cara untuk menemukan *plaintext*-nya. Sampai saat ini, hanya algoritme OTP yang dinyatakan tidak dapat dipecahkan meskipun diberikan sumber daya yang tidak

terbatas. Proses enkripsi dapat dilakukan dengan persamaan matematis seperti pada

plaintext : ONETIMEPAD

Kunci : TBFRGFARFM

Misalkan A = 0, B = 1, …, Z = 25.

ciphertext: HOJKOREGHP

yang diperoleh dengan cara sebagai berikut.

(O + T) mod 26 = H

(N + B) mod 26 = O

(E + F) mod 26 = J, dan seterusnya

Proses enkripsi dan dekripsi dengan kombinasi yang dimodifikasi OTP dan LCG. Algoritma OTP yang dimodifikasi tidak berbeda dari algoritma OTP asli(Wang & Wu, 2018).

Perbedaan ada dalam proses enkripsi dan dekripsi yang tidak melibatkan plaintext pertama karakter. Dalam menghasilkan kunci, kami menggunakan a = 3; b = 17; dan m = 101.

Proses Enkripsi:

Ada beberapa langkah untuk melakukan proses enkripsi:

Langkah – 1 : Ambil plaintext untuk dienkripsi

Langkah – 2 : Ubah huruf pertama menjadi kode ASCII

Langkah – 3 : Buat angka di Langkah -2 sebagai Z0 (seed) untuk menghasilkan angka acak

Langkah – 4 : Meninggalkan huruf pertama tidak terenkripsi

Langkah – 5 : Kunci = Z

Langkah – 6 : Lakukan proses enkripsi pada surat berikutnya

Langkah – 7 : Hasilkan angka acak Zn menggunakan LCG

Langkah – 8 : Kembali ke langkah 5 hingga semua huruf terenkripsi

Proses Deskripsi

Ada beberapa langkah untuk melakukan proses dekripsi:

Langkah – 1 : Ambil ciphertext untuk didekripsi

Langkah – 2 : Ubah huruf pertama menjadi kode ASCII

Langkah – 3 : Buat angka pada Langkah -2 sebagai Z0 untuk menghasilkan angka acak

Langkah – 4 : Meninggalkan huruf pertama tidak terenkripsi

Langkah – 5 : Kunci = Z

Langkah – 6 : Lakukan proses dekripsi pada surat berikutnya

Langkah – 7 : Menghasilkan angka acak Zn

Langkah – 8 : Kembali ke langkah 5 hingga semua huruf dideskripsi

2.7 *Unified Modelling Language* (UML)

Bahasa pemodelan perangkat lunak *unified modeling language* (UML), sejak pertama kali diperkenalkan pada tahun 1997, saat ini telah berkembang menjadi sebuah bahasa pemodelan yang baku (*de facto*) di dalam sebuah pengembangan perangkat lunak.UML digunakan dalam pengembangan sistem perangkat lunak yang menggunakan pendekatan berorientasi objek.Intensitas penggunaan UML yang tinggi ini didukung dengan semakin matangnya konsep pemodelan yang

dirumuskan dalam setiap rilis spesifikasi UML yang dikembangkan oleh *Object Management Group*(OMG).Sampai tahun 2017, OMG telah merilis 11 versi spesifikasi UML, yang terakhir adalah versi 2.5.1 yang termasuk dalam revisi UML 2.0.Di sisi lain, pengembangan alat bantu untuk pemodelan dengan UML berkembang cukup pesat dan sebagiannya tergolong sebagai free software sehingga tersedia banyak pilihan bagi pengembang perangkat lunak untuk menggunakannya, antara lain: StarUML, ArgoUML, UML Designer(Wati & Kusumo, 2016).

UML menyediakan banyak sekali diagram yang diperlukan untuk menjelaskan sistem yang sedang dikembangkan, baik dari aspek statis maupun dinamisnya (OMG, 2017). Salah satu diagram penting yang digunakan untuk mengilustrasikan kebutuhan (*requirements*) dari sistem adalah *use case* (UC) *diagram*, yang menjelaskan secara visual konteks dari interaksi antara aktor dengan sistem. Setiap *use case* menyatakan spesifikasi perilaku (*fungsionalitas*) dari sistem yang sedang dijelaskan yang memang dibutuhkan oleh aktor untuk memenuhi tujuannya. Namun demikian, penjelasan detil dari interaksi yang terjadi antara aktor dan sistem, berkaitan dengan sebuah *use case* tertentu, harus dijelaskan secara deskriptif dalam sebuah *use case* (UC) scenario. Oleh karena itu, UC scenario dan UC diagram, yang dibutuhkan dalam pemodelan UC dari sebuah sistem, harus mampu menjelaskan fungsionalitas sistem secara lengkap dan valid(Wira, Putra, & Andriani, 2019).

Dalam praktiknya, pembuatan UC scenario dan UC diagram bisa dilakukan dengan cukup mudah, meskipun oleh pengembang sistem yang belum

berpengalaman. Namun demikian, pembuatan sebuah penjelasan *use case* yang baik dan bermanfaat ternyata membutuhkan keahlian dan pengalaman yang cukup (Wati & Kusumo, 2016). Sebagai akibat dari minimnya pengetahuan dan pengalaman dalam pemodelan UC maka UC scenario dan UC diagram yang dihasilkan ternyata masih sering mengandung kesalahan sehingga model gagal merepresentasikan fungsionalitas dari sistem dengan tepat. Terlebih lagi, tidak sedikit pengembang yang masih memiliki paradigma yang salah yang memandang bahwa program yang bisa dieksekusi (*executable codes*) adalah satu-satunya produk yang penting dari sebuah pengembangan perangkat lunak sehingga aspek pemodelan dan dokumentasi sistem menjadi terabaikan(Wira et al., 2019).

Selanjutnya, berdasarkan pengalaman penulis dalam mengampu mata kuliah Rekayasa Perangkat Lunak dan/atau Analisis dan Perancangan Sistem dan/atau Pemodelan Perangkat Lunak di tingkat sarjana dan/atau pascasarjana selama hampir 15 tahun, tidak sedikit mahasiswa-mahasiswi yang menghasilkan UC scenario dan UC diagram yang salah, baik secara sintaksis maupun semantik. Disamping itu, laporan skripsi yang mengandung penjelasan sistem dengan menggunakan UC scenario dan UC diagram juga tidak sedikit yang salah, meskipun sudah melewati proses pembimbingan dan pengujian. Lebih dari itu, ternyata tidak sedikit artikel jurnal ilmiah yang juga mengandung penjelasan UC scenario dan UC diagram yang tidak tepat. Hal yang sama juga terjadi pada dunia industri perangkat lunak. Lebih dari itu, pengalaman penulis di dunia praktis industri perangkat lunak selama hampir 7 tahun juga mengonfirmasikan fenomena tersebut. Kesalahan-kesalahan yang umum dilakukan dalam beberapa kasus

tersebut, antara lain penjelasan yang umum dan hanya interaksi dari sisi aktor saja pada pembuatan UC scenario, dan penggambaran UC yang mengandung urutan sebagaimana yang ada dalam konsep data flow diagram (DFD) pada pembuatan UC diagram. Kesalahan dalam pemodelan UC, sehingga tidak sesuai dengan sistem sebenarnya yang dikembangkan, akan membuat proses pengembangan perangkat lunak menjadi lebih sulit untuk dikelola akibat adanya inkonsistensi. Kesalahan ini juga akan berdampak pada kemudahan proses pemeliharaan (*maintenance*) yang dilakukan setelah perangkat lunak selesai dibuat. Untuk menghindari terjadinya inkonsistensi tersebut, sudah ada beberapa penelitian yang mencoba melakukan pengecekan kesalahan yang ada pada UC scenario sehingga bisa dilakukan antisipasi perbaikan sedini mungkin. Namun demikian, potensi permasalahan yang mungkin terjadi pada pembuatan UC scenario dan UC diagram akan lebih efektif untuk diantisipasi jika kita mampu mengetahui kesalahan-kesalahan yang harus dihindari(Wati & Kusumo, 2016).

Artikel ini mencoba mengidentifikasi dan menjelaskan berbagai bentuk kesalahan yang sering terjadi pada pembuatan UC scenario dan UC diagram. Selanjutnya, rekomendasi pembetulan yang dibutuhkan sesuai dengan kaidah yang benar juga akan dijelaskan secara detil dengan menggunakan beberapa contoh kasus yang relevan. Dengan demikian, kesalahan-kesalahan yang sama bisa dihindari dalam proses pengembangan perangkat lunak sehingga bisa dihasilkan produk perangkat lunak yang berkualitas dan konsisten(Wira et al., 2019).

Unifed Modeling Language(UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendekskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek(Wati & Kusumo, 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"(Wira et al., 2019).

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*(Dharwiyanti, 2003).

"*Unified Modeling Language* (UML) bukanlah suatu proses melainkan bahasa pemodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh artifak sistem perangkat lunak. Penggunaan model ini bertujuan untuk mengidentifikasikan bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain di luarnya."(Dharwiyanti, 2003).

"*Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan dari sebuah sistem pengembangan software berbasis object oriented."(Dharwiyanti, 2003).

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (UML). UML muncul karena adanya kebutuhan pemodelan *visual* untuk menspesifikasikan, menggambarkan, membangun, dokumentasi dari sistem perangkat lunak. UML merupakan bahasa *visual* untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung(Dharwiyanti, 2003).

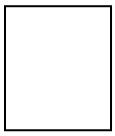
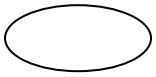
UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented programming*).

Diagram-diagram dalam UML antara lain:

1. *Use Case Diagram*

   Bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

Tabel 2.1 **Daftar Simbol *Use Case Diagram***

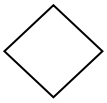
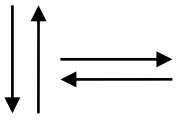
No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2	<<*include*>> ---------->	*Include*	Menspesifikasikan bahwa *use case* sumber secara eksplisit
3	<<*extend* >> <----------	*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
4	————	*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

Sumber : (Dharwiyanti, 2003)

2. *Activity Diagram*

Bersifat dinamis. Diagram aktifitas ini adalah tipe khusus dari diagram *state* yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem.

Tabel 2.2 **Daftar Simbol *Activity Diagram***

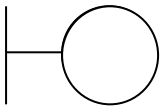
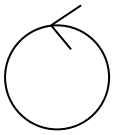
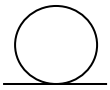
NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		*Action*	*State* dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan diakhiri.
5		*Decision*	Digunakan untuk menggambarkan suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.
6		*Line Connector*	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

Sumber : (Dharwiyanti, 2003)

3. *Sequence Diagram*

Bersifat dinamis.Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.

Tabel 2.3 **Daftar Simbol *Sequence Diagram***

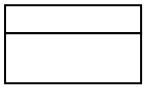
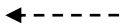
NO	GAMBAR	NAMA	KETERANGAN
1		*Life Line*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Actor*	Digunakan untuk menggambarkan *user*/ pengguna
3	*Message*()	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
4		*Boundary*	Digunakan untuk menggambarkan sebuah *form*
5		*Control Class*	Digunakan untuk menghubungkan *Boundary* dengan tabel.
6		*Entity Class*	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

Sumber : (Dharwiyanti, 2003)

4. *Class Diagram*

Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

Tabel 2.4 Daftar Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Class*	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama
2		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya
3		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil
4		*Realization*	Operasi yang benar benar dilakukan oleh suatu objek
5		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent)* akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri

Sumber : (Dharwiyanti, 2003)

2.8 *Flowchart*

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan *programmer*untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu flowchart harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman (Algoritma & Kom, 2013)**.**

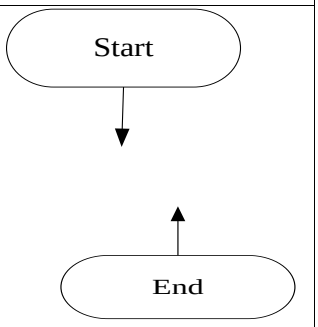
Dalam pembuatan flowchart program tidak ada rumus atau patokan yang bersifat mutlak. Karena flowchart merupakan gambaran hasil pemikiran dalam menganalisis suatu masalah yang nantinya akan diubah menjadi program komputer. Sehingga flowchart yang dihasilkan dapat bervariasi antara satu pemrogram dengan yang lainnya. Namum demikian terdapat beberapa anjuran yang harus diperhatikan, yaitu :

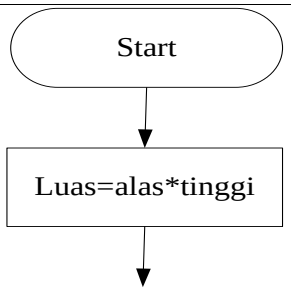
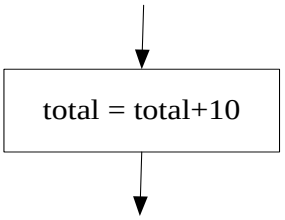
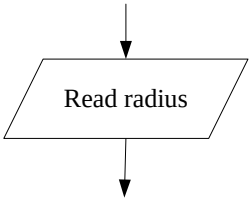
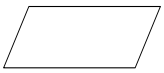
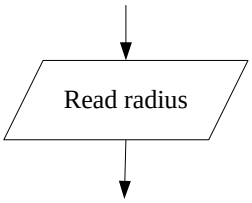
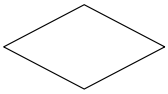
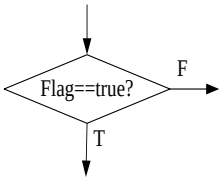
1. *Flowchart* digambarkan di suatu halaman dimulai dari sisi atas ke bawah dan dari sisi kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan dengan menggunakan bahasa dan simbol yangtepat dan definisi ini harus dapat dimengerti oleh pembacanya.

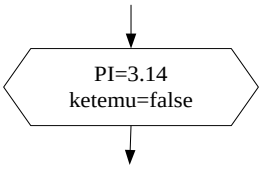
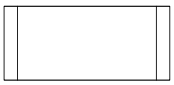
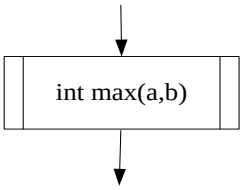
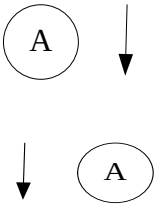
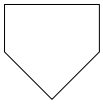
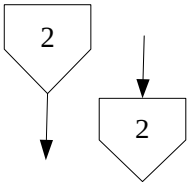
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas. Hanya terdapat satu titik awal dan satu titik akhir.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hatihati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol *flowchart* yang standar.

Simbol-simbol *flowchart* yang biasanya dipakai adalah simbol-simbol *flowchart* standar yang dikeluarkan oleh ANSI dan ISO. Tabel 2.5 merupakan beberapa simbol *flowchart* yang digunakan dalam menggambar suatu *flowchart:*

Tabel 2.5**Simbol-Simbol *Flowchart***

SIMBOL	NAMA	FUNGSI	CONTOH
	Terminator	Simbol Awal (Start) / Simbol Akhir (End)	

	Flow Line	Simbol aliran / penghubung	
	Proses	Perhitungan / pengolahan	
	Input / Output Data	Mempresentasikan pembacaan data (read) / penulisan (write)	
	Input / Output Data	Mempresentasikan pembacaan data (read) / penulisan (write)	
	Decision	Simbol pernyataan pilihan, berisi suatu kondisi yang selalu menghasilkan 2 nilai keluaran yaitu benar atau salah	

	Preparation	Inisialisasi / pemberian nilai awal	
	Predefined Process (subprogram)	Proses menjalankan sub program / fungsi / prosedur	
	On Page Connector	Penghubung Flowchart pada satu halaman	
	Off Page Connector	Penghubung Flowchart pada halaman berbeda	

Sumber : (Algoritma & Kom, 2013)

2.9 *Microsoft Visual Studio*

Microsoft Visual Studio adalah sebuah *Integrated Development Environment* buatan *Microsoft Coroporation*. *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas .NET *Framework*). Selain itu, *Visual Studio* juga

dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas .NET *Compact Framework*). *Visual Basic* mencakup sebuah kode editor yang didukung oleh fitur *intellisense* atau yang disebut dengan code *refactoring*. *Debugger* telah terintegrasi bekerja pada *level source level debugger* dan *level debugger mesin*. *Toll built in* mencakup *form desainer* untuk membangun sebuah aplikasi GUI, web *desainer*, *class desainer* dan database *schema desainer*.(Sains & Pengantar, 2018).

Microsoft Visual Studio didukung bahasa pemrograman yang berbeda. Adapun bahasa pemrograman yang didukung oleh *Visual Basic Studio* adalah bahasa pemrograman C++, *Visual Basic*, *Visual* C#. *Visual Studio* juga dapat mendukung bahasa pemrograman lain seperti M, *phyton* dan *ruby* yang semuanya itu terdapat pada *pack extra* yang terpisah dari *visual studio*(Sains & Pengantar, 2018).

2.10 Database

Database merupakan salah satu komponen yang sangat penting dalam sistem informasi, karena merupakan basis sistem dalam menyediakan informasi bagi para pemakai (Users & Independence, n.d.)**.**

Menurut **Kusrini** bahwa "*database* adalah kumpulan data yang saling terkait yang diorganisasi untuk memenuhi kebutuhan dan struktur sebuah organisasi serta bisa digunakan oleh lebih dari satu orang dan lebih dari satu aplikasi"(Users & Independence, n.d.).

2.9.1 Model Relasional

Model relasional pertama kali diperkenalkan oleh Dr. E.F. Codd pada 1970 dalam "*A Relational Model of Data for Large Shared Data Banks*" – Model relasional menggunakan *tables* dengan dua dimensi untuk menyimpan informasi.

1. Komponen Dasar Model Relasional

Terdapat tiga komponen dasar dari sebuah model relasional:

a. Struktur data relasional (*Table* dan *Index*): *Tables* apa saja yang ada dalam sebuah *database.*
b. Aturan-aturan (*Rules*) dari struktur data tersebut (*Con-strains*). Apakah sebuah *column* dalam *table* bisa memiliki nilai *Null*? *Column* mana yang menjadi *primary key* dan *foreign key*?
c. Operasi yang dilakukan terhadap struktur data tersebut (*Insert, Update, Delete*, dan *Merge*). Misalnya: apakah data bisa dihapus jika terhubung dengan data lain? Jika bisa, apa yang akan terjadi dengan data tersebut?

2. Istilah Umum dalam Model Relasional

Berikut adalah istilah-istilah umum model relasional yang sering kita temukan dalam buku-buku *database*.

a. *Entity* : struktur logikal dalam sebuah model relasional.
b. Relasional : juga disebut *tables* (tabel). Struktur 2 dimensi yang berada dalam *database* fisikal.
c. *Attributes* : juga disebut *columns* (kolom). Mendefinisikan *field* yang berada dalam sebuah relasi.

d. *Tuples* : juga disebut *rows* (baris) atau *records* (rekord). Berisi nilai data yang sebenarnya (*actual* data).

e. *Field* : Interseksi antara *column* dan *rows* adalah *field* yang berisi data

2.9.2 *Key* dan *Referential Integrity*

Setelah mengetahui komponen dan istilah-istilah umum yang digunakan dalam model relasional, kita akan melihat bagaimana data dalam sebuah *table* dapat dibedakan dengan data-data lainnya yang berada dalam *table* yang sama.

Pembeda ini disebut *Relation Key* atau *Key Values*. Nilai *key* adalah *column* unik yang digunakan untuk memilih *rows* dalam sebuah *table*/relasi.

1. *Primary Key*

Relation Key juga disebut sebagai *primary key*. *Primary key* adalah sebuah *column* atau beberapa *columns* yang dikelompokkan (*group of columns*) yang secara unik mengidentifikasi *row* dalam sebuah *table*.

2. *Unique Key*

Unique keys bisa memiliki nilai *NULL*. Kegunaan *unique keys* lebih condong untuk mencegah duplikasi dalam sebuah *table* daripada untuk mengidentifikasi rows.

3. *Foreign Key*

Istilah lain yang banyak dipakai dalam membuat relasi antara satu *table* dan *table* lainnya adalah *Foreign Key*. *Foreign key* adalah *columns* dalam sebuah *table* yang bukan *primary key*, *column* tersebut merupakan *primary key* dalam *table* lainnya(Users & Independence, n.d.).

2.9.3 Kardinalitas dalam Hubungan Relasi

Kardinalitas (*cardinality*) menunjukkan banyaknya hubungan antara *entity* dalam sebuah model relasional. Secara sederhana dapat disimpulkan ada 4 (empat) macam hubungan relasi (*relationship*), yaitu:

1. *One-to-One* (1:1) Satu-ke-Satu

   Setiap orang memiliki satu mobil dan setiap mobil dimiliki oleh satu orang.

2. *One-to-Many* (1:N) Satu-ke-Banyak

   Setiap orang dapat memiliki banyak mobil dan setiap mobil hanya dimiliki oleh satu orang.

3. *Many-to-One* (N:1) Banyak-ke-Satu

   Setiap orang memiliki satu mobil dan setiap mobil bisa dimiliki oleh lebih dari satu orang.

4. *Many-to-Many* (N:M) Banyak-ke-Banyak

Setiap orang bisa memiliki banyak mobil dan setiap mobil bisa dimiliki oleh banyak orang (Users & Independence, n.d.)

2.10 *Microsoft Access*

Microsoft access adalah program manajemen data hebat yang bisa digunakan untuk menyortir, mengatur dan melaporkan informasi penting yang dibutuhkan user sehari-hari. *Microsoft Access* adalah program manajemen data hebat yang bisa digunakan untuk menyortir, mengatur dan melaporkan informasi penting yang dibutuhkan user sehari-hari. *Microsoft access* membantu user mengelola *database* dengan cara menyediakan struktur yang efisien untuk menyimpan dan

mengambil informasi. Karena *Microsot Access* adalah sistem manajemen *database* relasional (RDBMS), user dapat mengatur data tentang *subyek-subyek* yang berbeda ke dalam tabel-tabel, kemudian membuat hubungan di antara tabel untuk menghindari adanya penggandaan data, menghemat ruang simpan dalam komputer dan memaksimalkan kecepatan dan akurasi bekerja dengan data(Users & Independence, n.d.).

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan oleh penulis ini dengan judul Pengembangan Pesan *Text* Menggunakan Kriptografi Untuk Keamanan Data Konsumen Pada Showroom Mobil Mitshubishi adalah sebagai berikut:

Gambar 3.1 Tahapan Penelitian

3.2 Metode Pengumpulan Data

Pengumpulan data adalah pencarian terhadap susuatu kerena ada perhatian dan keinginan terhadap hasil suatu aktivitas. Metode pengumpulan data dalam penulisan ini dibagi menjadi 3, yaitu :

1. Pengamatan (*Observation*)

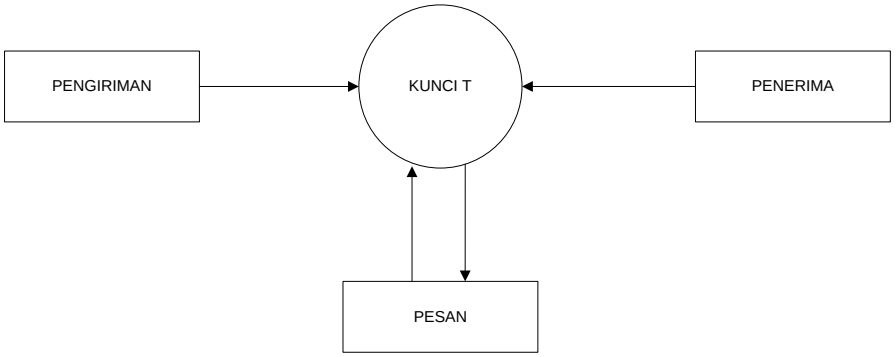
   Penulis melakukan pengamatan langsung terhadap proses transaksi yang terjadi antara konsumen dengan pihak lesing, mulai dari proses negosiasi, pengisian biodata konsumen, hinga poses serah terima kendaraan.

2. Penelitian Kepustakaan (*Library Research*)

   Merupakan cara untuk mencari referensi dengan mengumpulkan bahan-bahan pustaka yang dilakukan di perpustakaan kampus dan melakukan pencarian lewat internet, dengan mengunjungi situs-situs yang memaparkan jurnal-jurnal yang berhubungan dengan kriptografi *vigener cipher* dengan *Dynamic Key Linear Congruential Generator* (LCG).

3.3 Analisa Permasalahan yang Berjalan

Pertukaran data dalam hal ini pesan rahasia berbentuk teks dengan menggunakan metode tradisional yaitu dengan cara bertukar kata kunci tunggal. Diagram dibawah adalah penggambaran bagaimana pertukaran pesan rahasia menggunakan kunci tunggal terjadi.



Gambar 3.2 Skema Pengiriman Pesan

Pemberitahuan kata kunci dari pengirim ke penerima menggunakan media yang umum digunakan oleh banyak orang.

3.3.1 Analisa Kelemahan yang Berjalan

1. Penggunaan kata kunci tunggal berpotensi terjadinya salah pemahaman. Dalam hal ini kemungkinan penerima salah mengartikan kunci yang diberikan oleh pengirim adalah hal yang dapat terjadi.
2. Pemberitahuan atau pertukaran kata kunci yang dikirimkan oleh pengirim ke penerima memiliki potensi dapat diketahui oleh orang lain sehingga pesan rahasia dapat terbongkar.

3.3.2 Solusi Pemecahan Masalah

Pemecahan masalah yang penulis lakukan adalah dengan melakukan penerapan metode ini yang didalamnya terdapat Algoritma *Vigenere Cipher*. Penggunaan metode ini dapat digunakan sebagai solusi agar pengirim dan penerima

tidak lagi harus bertukar kunci tunggal untuk membuka pesan melainkan dapat memiliki kata kunci masing-masing.

Tabel 3.1 Tabel Perencanaan Rancangan

No	Sistem yang Berjalan	Sistem yang Diusulkan	Hasil yang Ingin Dicapai
1.	Penggunaan kunci tunggal yang harus diketahui oleh pengirim dan penerima untuk membuka pesan.	Pengirim dan penerima memiliki kunci masing-masing untuk membuka pesan	Tidak ada lagi kesalahan pemahaman atau salah tafsir kunci tunggal karena pengirim dan penerima memiliki kunci yang dapat ditetapkan masing-masing pihak.
2	Pertukaran kunci tunggal menggunakan media komunikasi yang rentan untuk dapat diketahui orang lain.	Pengirim dan penerima dapat menentukan sendiri kunci yang ingin digunakan untuk membuka pesan.	Kemungkinan bocornya kunci saat proses pertukaran informasi kunci tunggal dapat dihindari.

3.3.3 Analisa Proses Sistem Yang Berjalan

Berikut ini proses enkripsi dan dekripsi dari sebuah pesan (*plaintext*) yaitu "MITSUBISHI", dimana akan terlebih dahulu dibangkitkan kunci secar acak sepanjang 8 *digit*.

1. Bangkitkan kunci kedua dengan menggunakan algoritma LCG, dengan rumus :
   ki = (a * ki-1 + c) mod m
2. Dimana ki-1 = k1 yang artinya dibutuhkan k1 untuk membangkitkan k2, k3 dan seterusnya.
3. Untuk memperoleh k2, terlebih dahulu ditentukan nilai dimana k-1 = 7, a = 3, c = 19, dan m = 26, sehingga diperolehlah nilai k2 sebagai berikut :

k2	= (a * k1 + c) mod m	k3	= (a * k2 + c) mod m
	= (3 * 7 + 19) mod 26		= (3 * 14 + 19) mod 26
	= 40 mod 26		= 61 mod 26
	= 14		= 9
k4	= (a * k3 + c) mod m	k5	= (a * k4 + c) mod m
	= (3 * 9 + 19) mod 26		= (3 * 20 + 19) mod 26
	= 46 mod 26		= 79 mod 26
	= 20		= 1
k6	= (a * k5 + c) mod m	k7	= (a * k6 + c) mod m
	= (3 * 1 + 19) mod 26		= (3 * 22 + 19) mod 26
	= 22 mod 26		= 85 mod 26

= 22 = 7

k2 = (a * k7 + c) mod m

= (3 * 7 + 19) mod 26

= 40 mod 26

= 14

Berdasarkan langkah-langkah di atas, didapatkan suatu kunci 7, 14, 9, 20, 1, 22, 7 dan 14 yang apabila dikonversi kedalam bentuk huruf menjadi HOJUBW. Setelah proses *generate* kunci selesai, selanjutnya proses enkripsi *plaintext* dengan rumus : Ci = (Pi + Ki) mod 256. Jika diketahi *plaintext* = MITSUBISHI yang apabila dikonversi berdasarkan kode ASCII adalah sebagai beritu : 77, 73, 84, 83, 85, 66, 73, 83, 72 dan 73. Langkah-langkah untuk proses enkripsinya adalah sebagai berikut :

Dikarenakan panjang *plaintext* melebihi panjang *key*, maka diadakan pengulangan digit key untuk mengimbangi panjang *plaintext*. Adapun proses enkripsinya adalah sebagai berikut :

Plaintext	77	73	84	83	85	66	73	83	72	73
Key	7	14	9	20	1	22	7	14	9	20

1) C1 = (P1 + K1) mod 256

= (77 + 7) mod 256

= (84) mod 256

= 84

2) C2 = (P2 + K2) mod 256

= (73 + 14) mod 256

= (87) mod 256

= 87

3) C3 = (P3 + K3) mod 256

4) C4 = (P4 + K4) mod 256

= (84 + 9) mod 256

= (93) mod 256

= 93

5) C5 = (P5 + K5) mod 256

= (85 + 1) mod 256

= (86) mod 256

= 86

7) C7 = (P7 + K7) mod 256

= (73 + 7) mod 256

= (80) mod 256

= 80

9) C9 = (P9 + K9) mod 256

= (72 + 9) mod 256

= (81) mod 256

= 81

= (83 + 20) mod 256

= (103) mod 256

= 103

6) C6 = (P6 + K6) mod 256

= (66 + 22) mod 256

= (88) mod 256

= 88

8) C8 = (P8 + C8) mod 256

= (83 + 14) mod 256

= (97) mod 256

= 97

10) C10 = (P10 + K10) mod 256

= (73 + 20) mod 256

= (93) mod 256

= 93

Dari langkah-langkah diatas, didapatkan hasil enkripsi 84, 87, 93, 103, 86, 88, 80, 97, 81 dan 93 yang apabila dikonversi kedalam bentuk huruf menjadi TW]gVXPaQ].

Untuk proses dekripsinya dapat dilakukan dengan menggunakan rumus Pi = (Ci - Ki) mod 256. Adapun langkah-langkah proses dekripsi *ciphertext* menjadi *plaintext* adalah sebagai berikut :

1) P1 = (C1 - K1) mod 256
= (84 - 7) mod 256
= (77) mod 256
= 77

2) P2 = (C2 - K2) mod 256
= (87 - 14) mod 256
= (73) mod 256
= 73

3) P3 = (C3 - K3) mod 256
= (93 - 9) mod 256
= (84) mod 256
= 84

4) P4 = (C4 - K4) mod 256
= (103 - 20) mod 256
= (83) mod 256
= 83

5) P5 = (C5 - K5) mod 256
= (86 - 1) mod 256
= (85) mod 256
= 85

6) P6 = (C6 - K6) mod 256
= (88 - 22) mod 256
= (66) mod 256
= 66

7) P7 = (C7 - K7) mod 256
= (80 - 7) mod 256
= (73) mod 256
= 73

8) P8 = (C8 - K8) mod 256
= (97 - 14) mod 256
= (83) mod 256
= 83

9) P9 = (C9 - K9) mod 256
= (81 - 9) mod 256
= (72) mod 256
= 72

10) P10 = (C10 - K10) mod 256
= (93 - 10) mod 256
= (83) mod 256
= 83

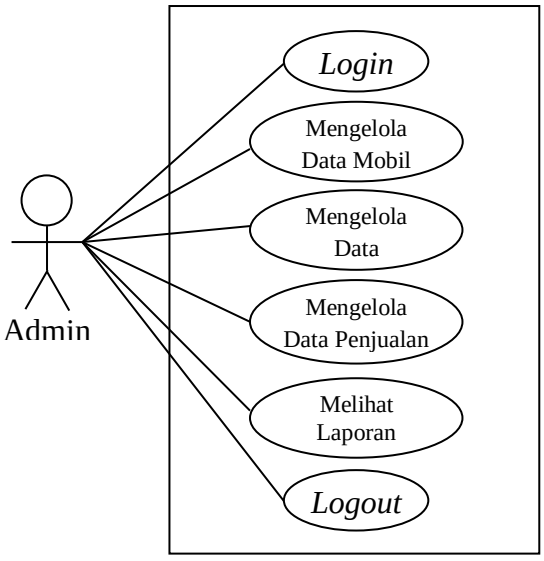
Dari langkah-langkah diatas, didapatkan hasil dekripsi 77, 73, 84, 83, 85, 66, 73, 83, 72 dan 83 yang apabila dikonversi kedalam bentuk huruf menjadi MITSUBISHI.

3.4 Perancangan Berorientasi Objek

Perancangan atau Pemodelan Berorientasi Ojek merupakan proses mendapatkan informasi dari model dan menampilkannya secara grafik dengan menggunakan sebuah standar elemen grafik.

3.4.1 *Use case Diagram*

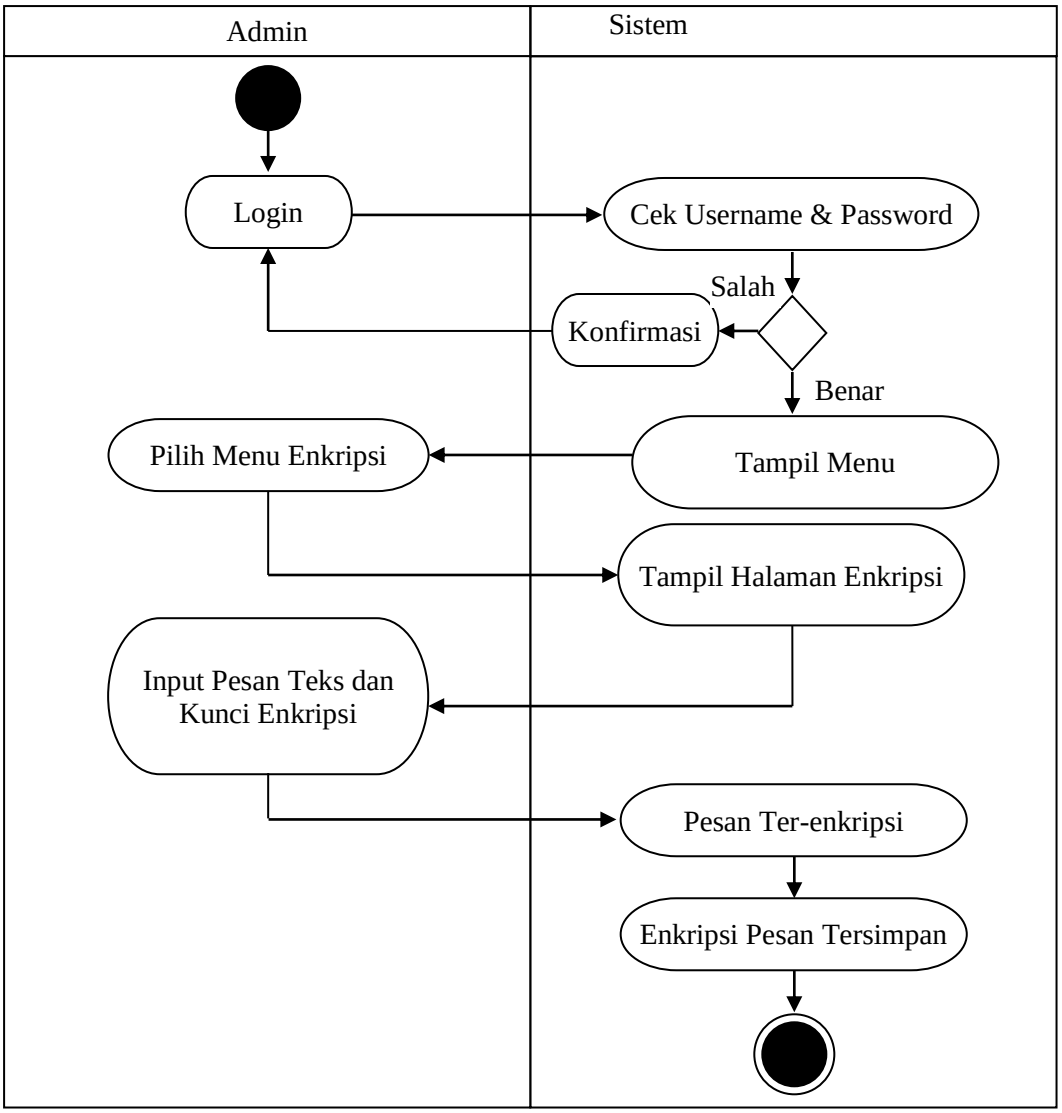
Berikut adalah *use case diagram* yang menggambarkan kegiatan.



Gambar 3.3. *Use Case Diagram*

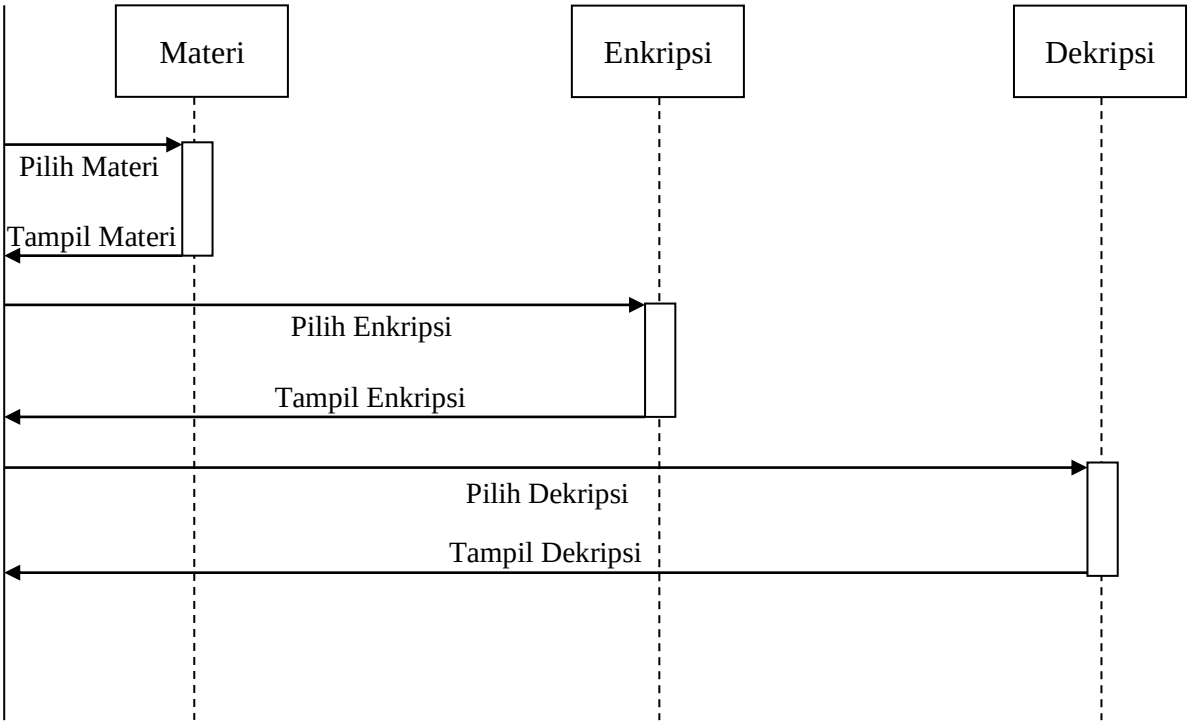
3.4.2 Pembuatan *Activity Diagram*

Activity diagram menggambarkan aktifitas-aktifitas yang terjadi dalam aplikasi dari aktivitas dimulai sampai aktivitas berhenti.



Gambar 3.4. *Activity Diagram*

3.4.3 *Sequence Diagram*



Gambar 3.5 *Sequence Diagram*

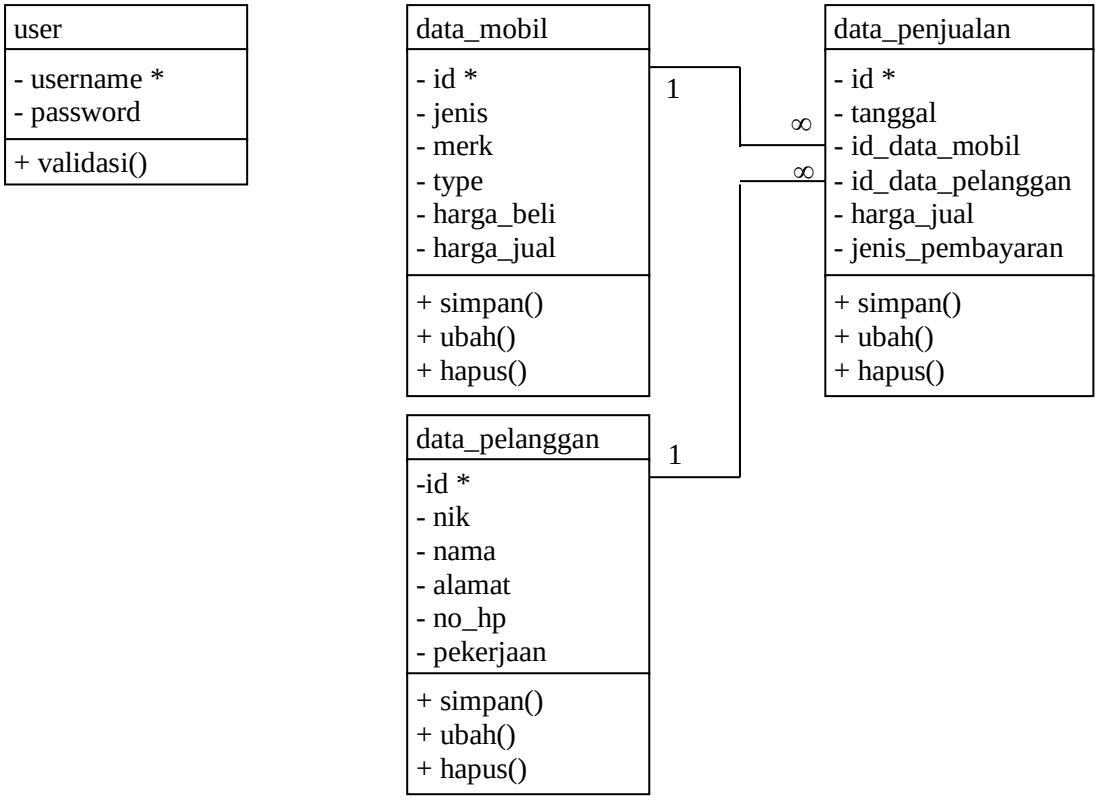
Keterangan Gambar :

1. Dalam diagram di atas menjelaskan bahwa *user* memilih materi kemudian Sistem menampilkan materi yang berkaitan dengan materi
2. *User* me*request* Enkripsi kemudian Sistem menampilkan menu Enkripsi
3. *User* me*request* Deskripsi kemudian Sistem menampilkan menu Deskripsi

3.4.4 Class Diagram

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam

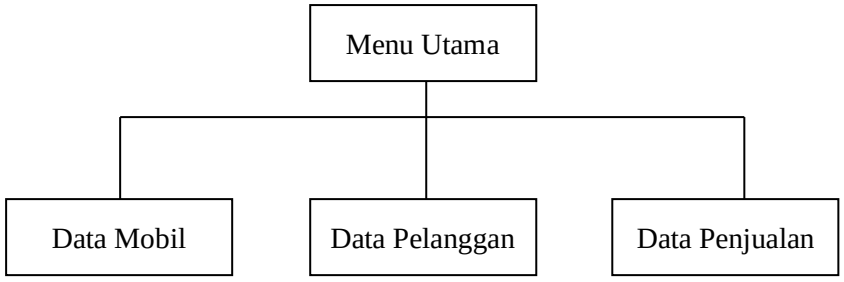
hubungan-hubungan objek tersebut. Dibawah ini digambarkan *class diagram* dari rancangan sistem yang akan dibuat.



Gambar 3.6 *Class Diagram*

3.4.5 Struktur Program

Struktur program mempresentasikan organisasi komponen program (modul) serta mengimplementasikan suatu hirarki kontrol. *Hirearki control* tidak mengimplementasikan aspek *procedural* dari perangkat lunak seperti urutan proses, kejadian atau urutan dari keputusan atau perulangan operasi.

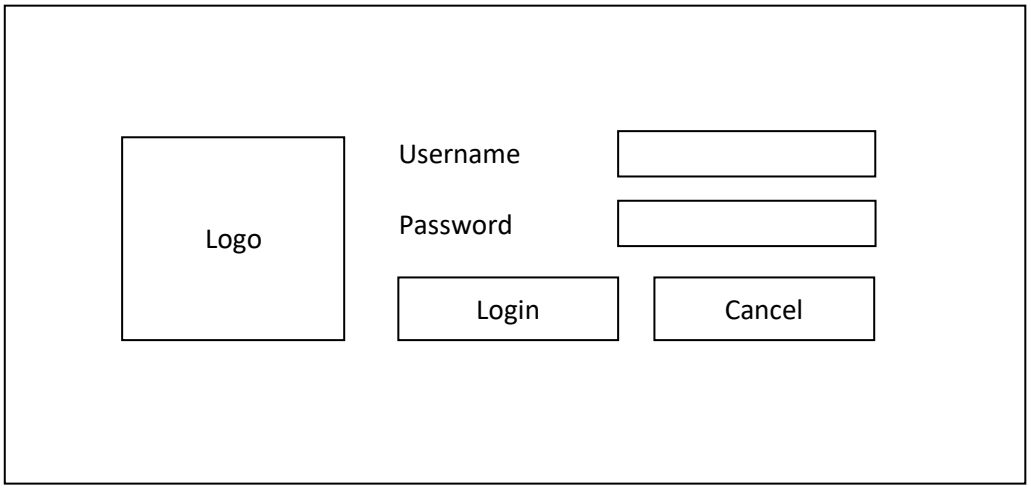


Gambar 3.7 Struktur Navigasi Enkripsi

3.5 Perancangan Antarmuka

3.5.1 *Form Login*

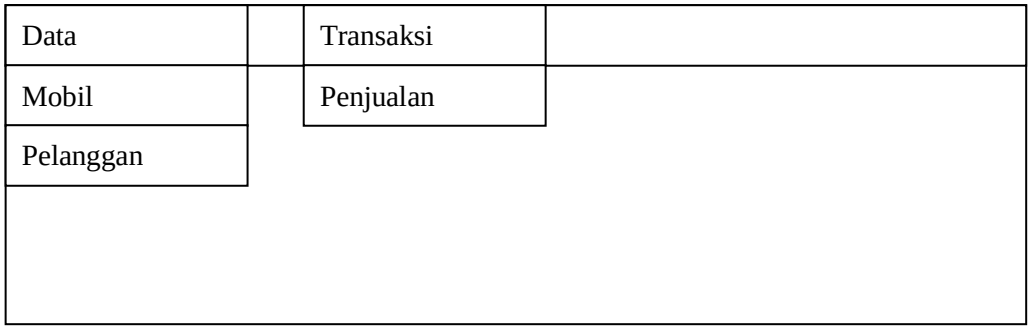
Form login adalah *form* yang digunakan oleh admin untuk dapat masuk kedalam sistem.



Gambar 3.8 *Form Login*

3.5.2 *Form* Menu Utama

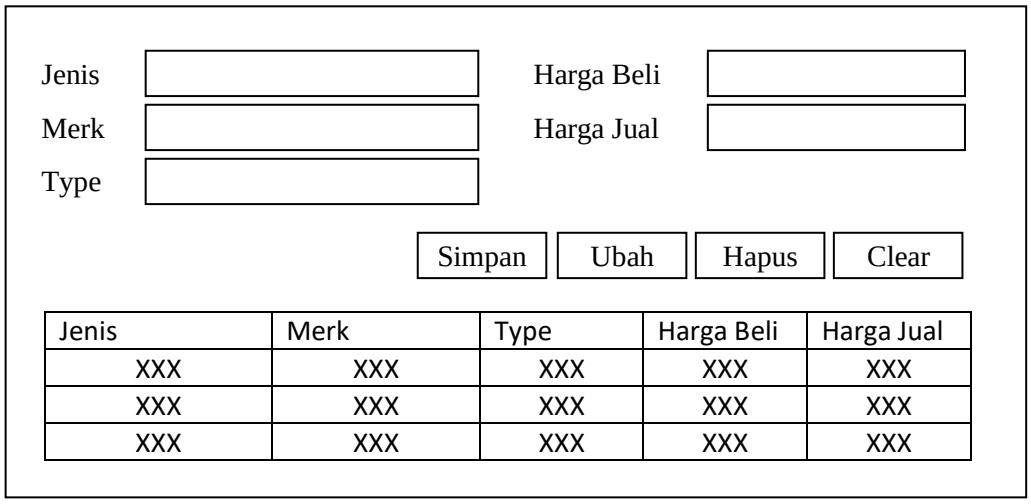
Form menu utama adalah *form* yang diakases saat admin pertama kali berhasil login.



Gambar 3.9 *Form* Menu Utama

3.5.3 *Form* Data Mobil

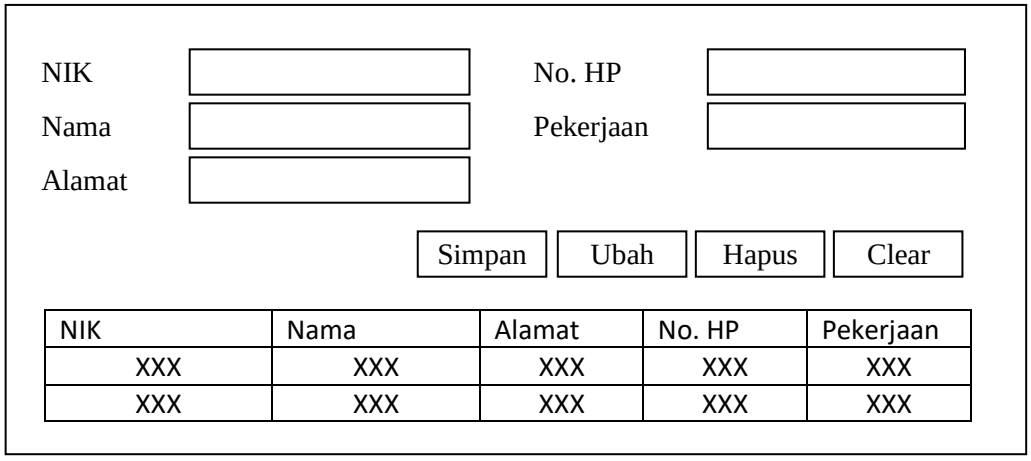
Form data mobil adalah *form* yang digunakan admin untuk menyimpan, mengubah dan menghapus data mobil.



Gambar 4.0 *Form* Data Mobil

3.5.4 *Form* Data Pelanggan

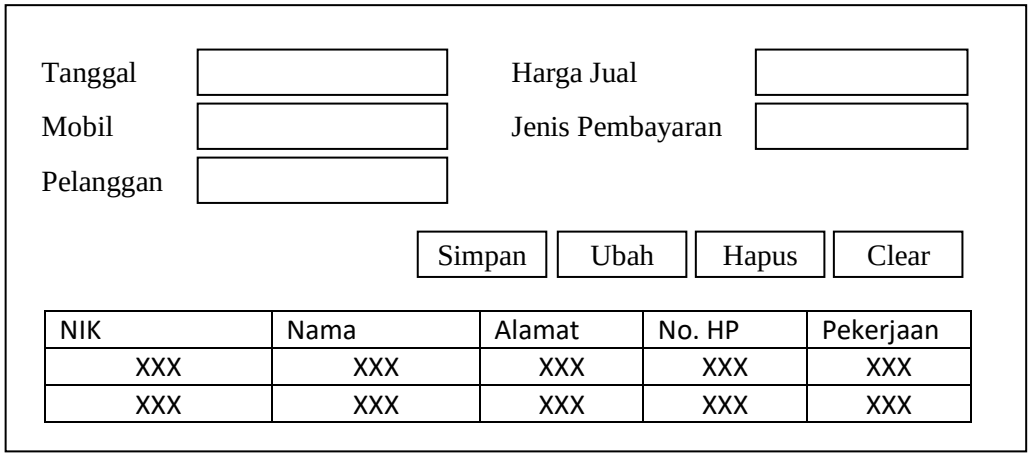
Form data pelanggan adalah *form* yang digunakan admin untuk menyimpan, mengubah dan menghapus data pelanggan.



Gambar 4.1 *Form* Data Pelanggan

3.5.5 *Form* Data Penjualan

Form data penjualan adalah *form* yang digunakan admin untuk menyimpan, mengubah dan menghapus data penjualan.



Gambar 4.2 *Form* Data Penjualan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Spesifikasi Minimum *Hardware* dan *Software*

Dalam implementasi dan pengujian sistem yang dibangun membutuhkan dua perangkat yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*). Adapun perangkat lunak dan keras yang dibutuhkan adalah sebagai berikut :

1. Perangkat Lunak (*Software*)
   a. Sistem operasi Windows 7, Windows 8, Windows 8.1.
   b. Apache2triad.
   c. Mozilla Firefox.
2. Perangkat Keras (*Hardware*)
   a. Processor minimal Intel Dual Core.
   b. RAM minimal 1GB.
   c. Harddisk minimal 160GB.
   d. Monitor.
   e. Printer.
   f. Mouse dan Keyboard.

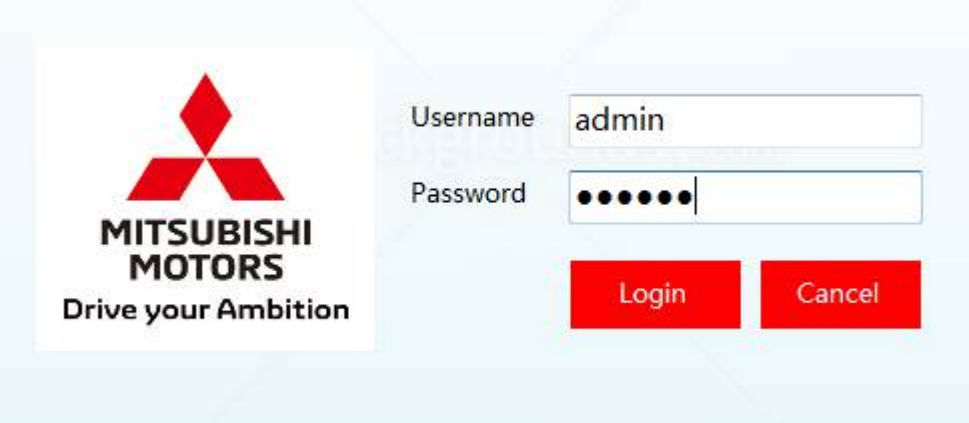
4.2 Pengujian Aplikasi dan Pembahasan

Impelementasi sistem merupakan merupakan langkah yang dilakukan untuk mengoperasikan sistem yang dibangun. Dalam bab ini akan dijelaskan bagaimana

menjalankan sistem tersebut. Rancangan dan keterangan dari masing-masing halaman yang dirancang akan dijelaskan sebagai berikut :

a. Halaman *Login*

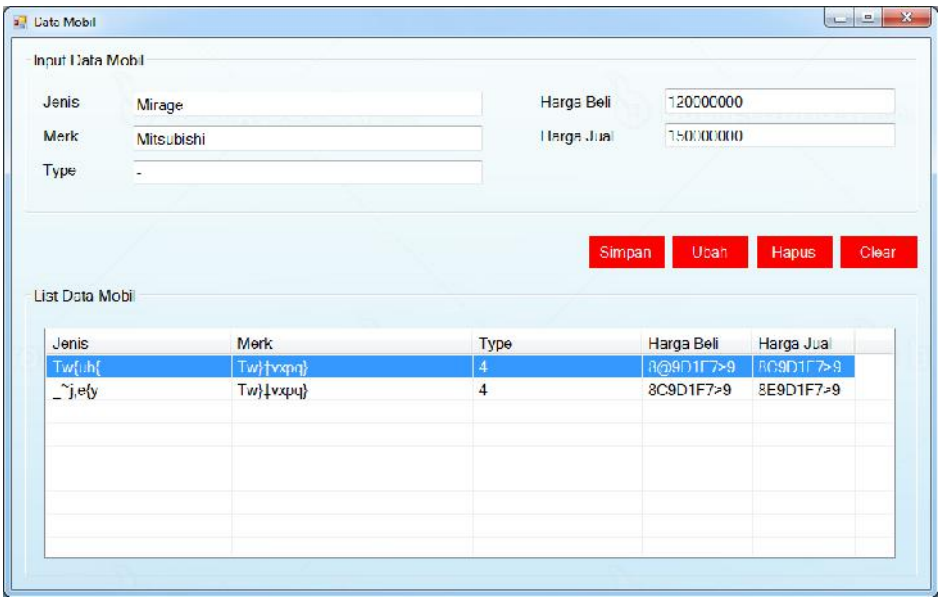
Halaman *login* merupakan halaman yang digunakan admin untuk masuk kedalam sistem.



Gambar 4.3 Tampilan Halaman *Login*

b. Halaman Data Mobil

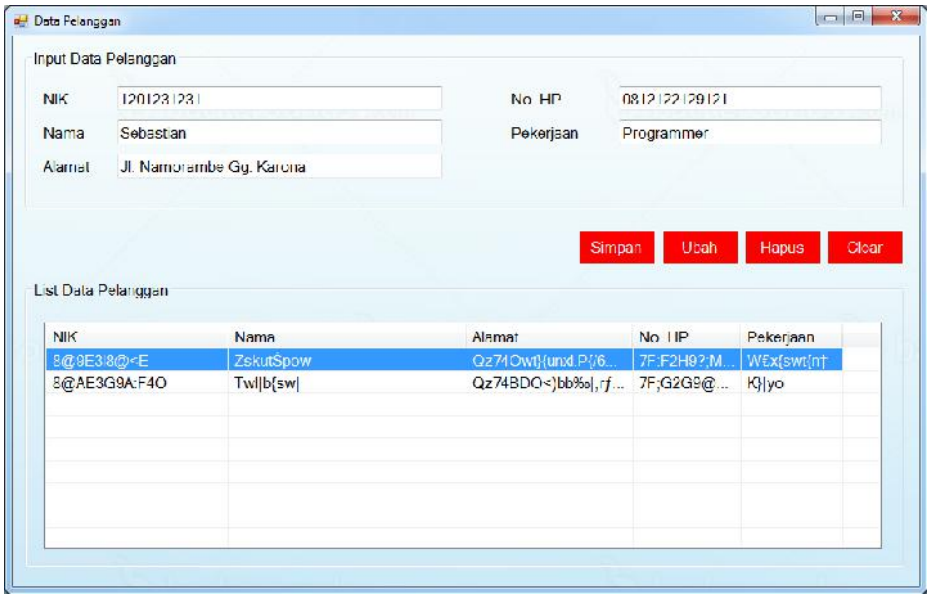
Halaman data mobil merupakan halaman yang digunakan oleh admin untuk menginput, mengubah dan menghapus data mobil.



Gambar 4.4 Tampilan Halaman Data Mobil

c. Halaman Data Pelanggan

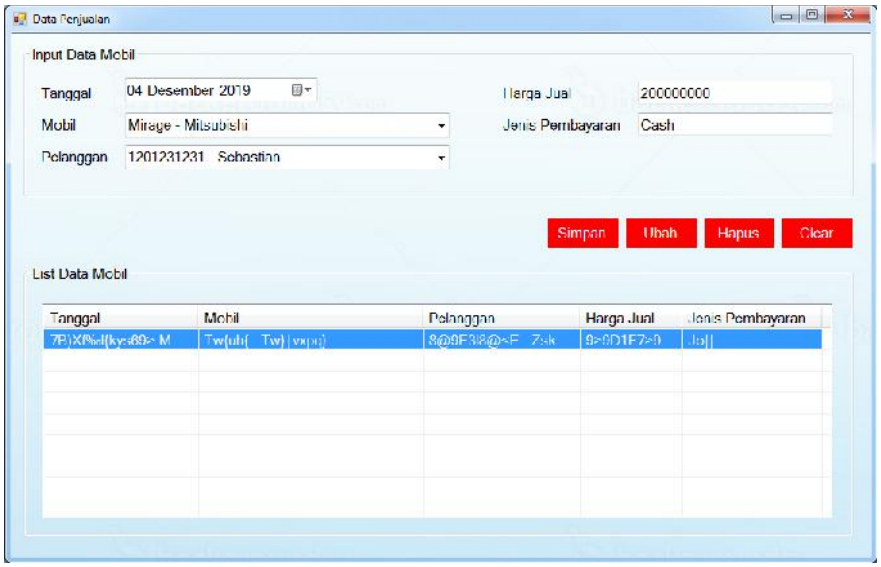
Halaman data pelanggan merupakan halaman yang digunakan admin untuk menginput, mengubah dan menghapus data pelanggan.



Gambar 4.5 Tampilan Halaman Pelanggan

d. Halaman Data Penjualan

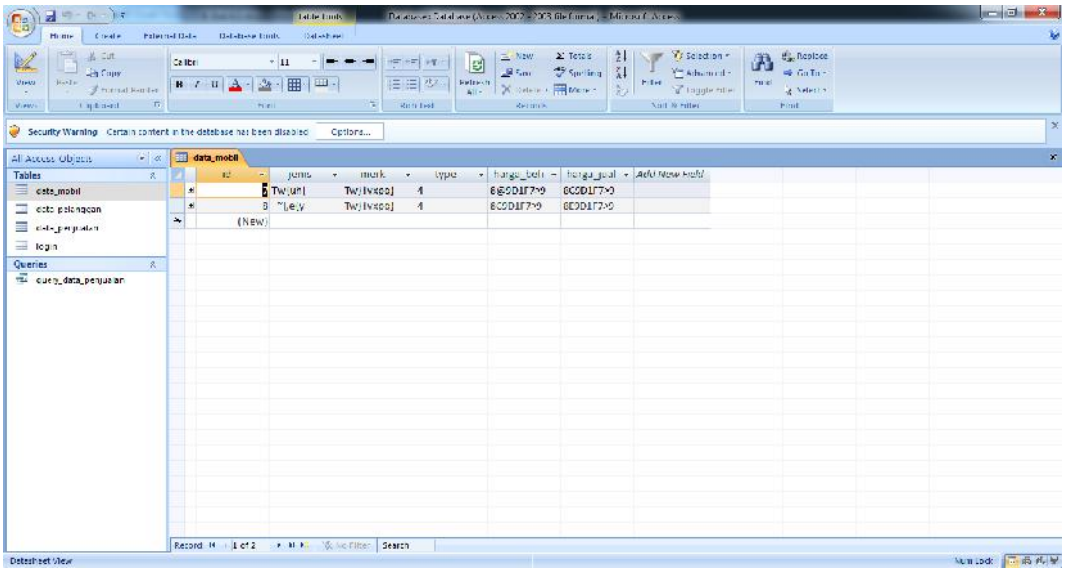
Halaman data penjualan merupakan halaman yang digunakan admin untuk mengelola data penjualan.



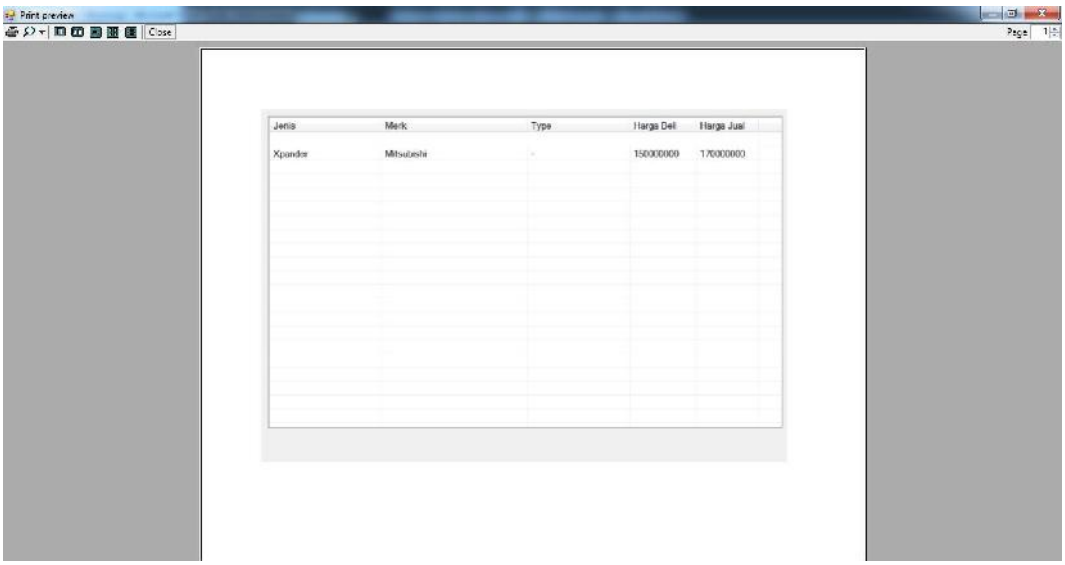
Gambar 4.6 Tampilan Halaman Data Penjualan

4.3 Pengujian

Proses uji coba enkripsi yang dilakukan pada sistem yaitu dengan mengenkripsikan data mobil, data pelanggan dan data penjualan. Hasil enkripsi dapat dilihat pada gambar di bawah ini :



Gambar 4.7 Hasil Enkripsi



Gambar 4.8 Hasil Dekripsi

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dalam pengembangan pesan text menggunakan kriptografi untuk keamanan data antaralain :

1. Pengamanan pesan text memberi dampak yang begitu besar dalam hal keamanan data konsumen.
2. Pengembanganpesan text menggunakan kriptografi dapat dilakukan dengan menggunakan aplikasi *Microsoft Visual Studio 2008*.
3. Dalam mengamankan pesan text menggunakan kriptografi mengalami beberapa tahap proses transformasi sehingga akhirnya menjadi teks yang tersandikan.

5.2 Saran

Demi penyempurnaan aplikasi yang telah dibuat, adapun saran yang ingin disampaikan yaitu :

1. Diharapkan kedepannya sistem yang dibangun dapat dilengkapi dengan fasilitas penginputan simbol-simbol matematika.
3. Diharapkan kedepannya sistem yang telah dibangun dapat dilengkapi dengan fasilitas *export* data sehingga mempermudah proses penginputan data.

DAFTAR PUSTAKA

Algoritma, l. D. A. N., & kom, b. S. (2013). *Logika dan algoritma*.

Applications, b. (2018). Implementasi algoritma one time pad untuk proteksi file data pribadi pada aplikasi berbasis web, *03*(02), 140–150.

Dharwiyanti, s. (2003). P e n g a n t a r u n i f i e d m o d e l i n g l a n g u a g e (u m l), 1–13.

Diani, f., & widhiyasana, y. (2018). Enkripsi sms dengan menggunakan one time pad (otp) dan kompresi lempel-ziv-welch (lzw), *7*(3), 3–8.

Fachri, barany, agus perdana windarto, and ikhsan parinduri. "penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik." jepin (jurnal edukasi dan penelitian informatika) 5.2 (2019): 202-208.

Fachri, b., windarto, a. P., & parinduri, i. (2019). Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. Jepin (jurnal edukasi dan penelitian informatika), 5(2), 202-208.

Fachri, barany; windarto, agus perdana; parinduri, ikhsan. Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. Jepin (jurnal edukasi dan penelitian informatika), 2019, 5.2: 202-208.

Firdaus, i. L., marwati, r., & sispiyati, r. (n.d.). Aplikasi kriptografi komposisi one time pad cipher dan affine, 42–51.

Hamdi, nurul. "model penyiraman otomatis pada tanaman cabe rawit berbasis programmable logic control." jurnal ilmiah core it: community research information technology 7.2 (2019).

Handoyo, a. B., & teknik, s. (2013). Generator key vigenere cipher dengan menggunakan randomisasi dari key tertentu.

Harahap, m. K. (2019). Analisis algoritma one time pad dengan algoritma cipher transposisi sebagai pengamanan pesan teks, *1*(april 2017), 58–62.

Nasution, a. N., teknik, j., sekolah, i., teknik, t., & medan, h. (n.d.).

Pengamanan file dokumen berbasis teks menggunakan metode kriptografi one time pada dan algoritma pertukaran kunci diffie-helman.

Penulis, n., & rickson, f. (n.d.). Penggunaan kriptografi one time pad (algoritma vernam) dalam pengamanan informasi.

Permana, aminuddin indra. "kombinasi algoritma kriptografi one time pad dengan generate random keys dan vigenere cipher dengan kunci em2b." (2019).

Pratiwi, l. E., marwati, r., & yusnitha, i. (n.d.). Program aplikasi kriptografi penyandian one time pad menggunakan sandi vigenere, 43–53.

Putra, randi rian. "sistem informasi web pariwisata hutan mangrove di kelurahan belawan sicanang kecamatan medan belawan sebagai media promosi." jurnal ilmiah core it: community research information technology 7.2 (2019).

Putra, randi rian, et al. "decision support system in selecting additional employees using multi-factor evaluation process method." (2019).

Putra, randi rian. "implementasi metode backpropagation jaringan saraf tiruan dalam memprediksi pola pengunjung terhadap transaksi." jurti (jurnal teknologi informasi) 3.1 (2019): 16-20.

Sains, i., & pengantar, k. (2018). Membuat aplikasi rekam medis dengan microsoft visual studio 2010 dan database mysql program studi sistem informasi.

Saputra, muhammad juanda, and nurul hamdi. "rancang bangun aplikasi sejarah kebudayaan aceh berbasis android studi kasus dinas kebudayaan dan pariwisata aceh." journal of informatics and computer science 5.2 (2019): 147-157

Series, i. O. P. C., & science, m. (2018). Vigenere cipher algorithm modification by adopting rc6 key expansion and double encryption process. Https://doi.org/10.1088/1757-899x/420/1/012119

Sidik, a. P., efendi, s., & suherman, s. (2019, june). Improving one-time pad algorithm on shamir's three-pass protocol scheme by using rsa and elgamal algorithms. In journal of physics: conference series (vol. 1235, no. 1, p. 012007). Iop publishing.

Sitepu, n. B., zarlis, m., efendi, s., & dhany, h. W. (2019, august). Analysis of decision tree and smooth support vector machine methods on data mining. In journal of physics: conference series (vol. 1255, no. 1, p. 012067). Iop publishing.

Tasril, v., wijaya, r. F., & widya, r. (2019). Aplikasi pintar belajar bimbingan dan konseling untuk siswa sma berbasis macromedia flash. Jurnal informasi komputer logika, 1(3).

Users, d., & independence, d. (n.d.). Author : abhishek taneja lesson : Introduction lesson no . : 01.

Wang, j. B., & wu, h. Y. (2018). A comparison of one time pad random key generation using linear congruential generator and quadratic congruential generator a comparison of one time pad random key generation using linear congruential generator and quadratic congruential generator.

Wati, e. F., & kusumo, a. A. (2016). Penerapan metode unified modeling language (uml) berbasis desktop pada sistem pengolahan kas kecil studi kasus pada pt indo mada yasa tangerang, *5*(1), 24–36.

Wira, d., putra, t., & andriani, r. (2019). Unified modelling language (uml) dalam perancangan sistem informasi permohonan pembayaran restitusi sppd, *7*(1).