Implementasi Kriptografi untuk Pengamanan Basis Data Menggunakan Algoritma *Triangle Chain Chiper* Pada PT Mandiri Pratama Teknik Wahyu Gemilang

Yusran Fariz[1)], **Ferdiyansah**[2)]
[1]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : 1211520026@student.budiluhur.ac.id[1)], ferdiyansyah@budiluhur.ac.id[2)]

Abstark

PT Mandiri Pratama Teknik Wahyu Gemilang merupakan salah satu perusahaan kontraktor yang bekerjasama dengan provider telekomunikasiyang berfokus dalam bidang penawaran material dan jasa untuk internet rumah dan perkantoran. Dalam penjualan material, terdapat data – data penting yang bersifat rahasia, dan hanya orang – orang tertentu saja yang dapat mengakses data tersebut. Oleh karena itu dibutuhkan suatu aplikasi yang dapat memudahkan pengguna untuk menyimpan dan menginput data – data tersebut agar terjaga kerahasiaan dan keamanannya. Pada penilitian ini penulis mengimplementasikan keamanan data pada database. Karena dengan majunya perkembangan teknologi dan komunikasi yang begitu pesat, banyak sekali terjadinya kasus pencurian data – data penting. Teknik pengamanan data ini dilakukan menggunakan teknik kriptografi Triangle Chain. Algoritma ini nantinya akan digunakan untuk mengamankan suatu basis data, Dengan teknik kriptografi Triangle Chain ini dapat dimanfaatkan untuk mengamankan data serta memberikan kemudahan kepada user untuk mengamankan datanya agar isi data tersebut tidak diketahui oleh pihak yang tidak memiliki kepentingan untuk mengakses data tersebut. Pada tugas akhir ini penulis membuat suatu aplikasi pengamanan database dengan menggunakan algoritma kriptografi Triangle Chain. Aplikasi pengamanan database ini berbasis desktop dengan menggunakan bahasa pemograman Visual Basic. Dengan menggunakan aplikasi ini, penulis berharap agar pengguna dapat menyimpan datanya yang bersifat rahasia ke dalam database tanpa khawatir ada orang lain yang dapat mengakses data tersebut.

Kata kunci: *Basis Data, Kriptografi, Triangle Chain*

1. PENDAHULUAN

Salah satu dampak negatif dari perkembangan teknologi dan komunikasi ialah pencurian data oleh orang yang tidak bertanggung jawab. Hal ini merupakan salah satu masalah yang paling ditakuti dalam bidang jaringan komunikasi. Oleh sebab itu masalah keamanan dari suatu data merupakan aspek yang sangat penting dari suatu sistem informasi. Jika data tersebut bersifat sangat penting, maka data tersebut harus diberi pengamanan khusus agar terhindar dari pencurian data dari orang yang tidak bertanggung jawab. Mengimplementasikan suatu teknik kriptografi adalah salah satu cara untuk mengamankan data tersebut dengan proses enkripsi dan dekripsi. Dengan proses enkripsi ini, data asli nantinya akan diubah susunan penulisannya secara acak sehingga data asli tidak dapat dibaca oleh sembarang *user*.

Dengan metode *triangle chain cipher* diharapkan mampu memberikan perlindungan data sebaik mungkin dari berbagai ancaman pencurian data. Algoritma *triangle chain* ini nantinya akan diimplementasikan pada aplikasi pengamanan database di PT Mandiri Pratama Teknik Wahyu Gemilang, dimana memiliki data penting yang harus di lindungi agar terhindar dari pencurian dan pemalsuan data dari pihak yang tidak berwenang. PT Mandiri Pratama Teknik Wahyu Gemilang bergerak di bidang Telekomunikasi yang menawarkan jasa dan penjualan. Data – data penjualan inilah yang nantinya akan di amankan, agar data – datanya tidak dipakai oleh sembarang orang.

2. LANDASAN TEORI

2.1 ALGORITMA *TRIANGLE CHAIN CIPHER*

Algoritma *triangle chain cipher* disebut juga dengan algoritma rantai segitiga, algoritma ini merupakan algoritma yang dibuat untuk memperbaiki algoritma kriptografi klasik khususnya algoritma substitusi abjad tunggal yang sangat mudah diserang dengan teknik analisis frekuensi. *Triangle Chain cipher* memiliki aturan substitusi berdasar pada caesar cipher yaitu dengan pergeseran huruf-huruf. Algoritma ini memiliki aturan yang sama dengan *Caesar Cipher* yaitu dengan pergeseran huruf-huruf.

2.2 Algoritma Enkripsi Triangle Chain

Berikut ini merupakan rumus enkripsi pada algoritma *Triangle Chain*:

a. Matriks enkripsi segitiga pertama
   untuk baris ke-1 :
   M[1j] = P[j] + (K * R[1]) Mod 255
   untuk baris ke-2 dan selanjutnya untuk nilai j ≥ i :
   M[ij] = M[j -1]j + (K * R[i]) Mod 255, nilai ciperteks yang didapat adalah :
   M[ij] pada nilai j = (N + i) - N.
b. Matriks enkripsi segitiga ke-2

nilai P didapat dari nilai M[ij] pada nilai i = j, untuk baris ke-1 :
M[1j] = P[j] + (K * R[1]) Mod 255.
Untuk baris ke-2 dan selanjutnya untuk nilai j ≤ (N + 1) – i :
M[ij] = M[j -1]j + (K * R[i]) Mod 255. Maka nilai ciperteks yang didapat adalah: M[ij] pada nilai j = (N + 1) – i

Keterangan :
P = Plainteks
N = Jumlah karakter pada plainteks
M = Matriks penampung dari hasil penyandian
K = *Key* (Kunci)
R = *Row* (baris)
i = Indeks faktor pengali
j = Indeks karakter plaintext

2.3 Algoritma Dekripsi Triangle Chain

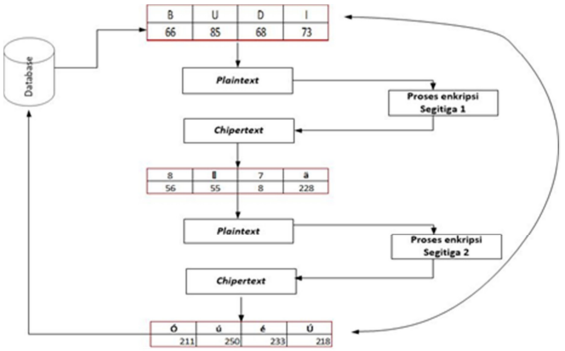
Berikut ini merupakan rumus dekripsi pada algoritma *Triangle chain*:

a. Matriks dekripsi segitiga pertama
untuk baris pertama :
Mij = C[j] – (K * R[1]) Mod 255,
untuk baris kedua dan selanjutnya nilai j ≤ (N + 1) – i
Mij = C[i,j] – (K * R[i]) Mod 255
maka nilai ciperteks dari segitiga pertama diambil dari nilai setiap barisnya dengan ketentuan :
M[ij] pada nilai i = n dan j ≤ (N + 1) – i
b. Matriks dekripsi segitiga kedua
untuk baris pertama berlaku formula :
M1j = C[j] – (K * R[1]) Mod 255
sedangkan untuk baris kedua dan seterusnya nilai j ≥ i, berlaku formula :
Mij = C[i,j] – (K * R[i]) Mod 255
nilai plainteks untuk ciperteks yang pertama adalah : M[ij] pada nilai j = (N + 1) – i.

Keterangan :
C = *Chiper Text*

2.4 Skema Proses Penyandian (Enkripsi) Algoritma Triangle Chain

Proses penyandian (enkripsi) dengan algoritma Triangle Chain, memiliki dua tahap proses penyandian, yaitu proses penyandia segitiga pertama dan kedua.



Gambar 1. Skema Model Proses Penyandian (Enkripsi) Algoritma Triangle Chain

Dari gambar 1 dapat dilihat proses dari enkripsi algoritma *Triangle Chain* dilakukan sebanyak dua kali, dimana proses enkripsi pertama yang menjadi plainteks adalah data asli dan diproses menghasilkan *ciphertext* yang nantinya dijadikan sebagai *plaintext* untuk proses enkripsi kedua (enkripsi segitiga kedua).

Tabel 1. Faktor Pengali Pada Kunci

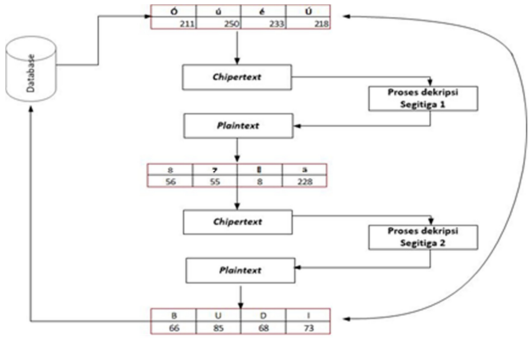
P(B)	*P*(U)	*P*(D)	*P*(I)	*Plaintext*, *i* = 0
C_{11}(B)	C_{12}UI)	C_{13}(D)	C_{14} (I)	*i* = 1 *fp*[1]
	C_{22}(U)	C_{23}(D)	C_{24}(I)	*i* = 2 *fp*[2]
		C_{33} (D)	C_{34}(I)	*i* = 3 *fp*[3]
			C_{44}(I)	*i* = 4 *fp*[4]

Ket: P = *Plaintext*, C = *Ciphertext*

Sesuai dengan tabel ASCII mod 255, setiap karakter yang akan di enkripsi atau dekripsi harus dirubah terlebih dahulu ke dalam bentuk desimal. Contoh: plainteks = B U D I nilai desimal pada ASCII = 66, 85, 68, 73.

2.5 Skema Proses Dekripsi Algoritma Triangle Chain

Proses dekripsi algoritma Triangle Chain kebalikan daripada proses enkripsi algoritma Triangle Chain, proses dekripsi algoritma Triangle Chain memiliki dua proses yaitu: dekripsi segitiga ke-1 dan dekripsi segitiga ke-2. Dekripsi ke-1 yang akan menjadi *ciphertext* adalah data yang telah tersandikan diproses menghasilkan *plaintext* yang nantinya dijadikan *ciphertext* untuk proses dekripsi kedua dan hasil proses dekripsi kedua (*plaintext*) adalah merupakan data asli yang sebelum di enkripsi. Untuk skema dekripsi algoritma *Triangle Chain* dapat dilihat pada gambar 2. Untuk proses dekripsi *Triangle Chain* adalah kebalikan dari proses enkripsi *Triangle Chain*.



Gambar 2. Skema Model Proses Dekripsi Algoritma Triangle Chain

Untuk proses dekripsi Triangle Chain adalah kebalikan dari proses enkripsi Triangle Chain.

2.5.1 Studi Kasus Algoritma Triangle Chain

a. Matriks enkripsi segitiga pertama.

KARAKTER B U D I
NILAI DESIMAL 66 85 68 73

Selanjutnya adalah melakukan proses enkripsi segitiga ke-1 sesuai dengan rumus:

P = BUDI

K = 1010

N = 4

R = 1, 2, 3, 4.

Rumus untuk baris ke-1 (i = 1):

M[1j] = P[j] + (K * R[1]) Mod 255, maka Penyelesain enkripsi baris pertama:

M_{11} = ($P_{[1]}$ + (**1010** * **R[1]**)) **Mod 255**
= (**B** + (**1010** * (**1**))) **Mod 255**
= (**66** + **1010**) **Mod 255** = **56**
(huruf **8** dalam karakter ASCII)

M_{12} = ($P_{[2]}$ + (**1010** * **R[1]**)) **Mod 255**
= (**U** + (**1010** * (**1**))) **Mod 255**
= (**85** + **1010**) **Mod 255** = **75**
(huruf **K** dalam karakter ASCII)

M_{13} = ($P_{[3]}$ + (**1010** * **R[1]**)) **Mod 255**
= (**D** + (**1010** * (**1**))) **Mod 255**
= (**68** + **1010**) **Mod 255** = **58**
(huruf **:** dalam karakter ASCII)

M_{14} = ($P_{[4]}$ + (**1010*** **R[1]**)) **Mod 255**
= (**I** + (**1010** * (**1**))) **Mod 255**
= (**73** + **1010**) **Mod 255** = **63**
(huruf **?** dalam karakter ASCII)

Maka hasil dari enkripsi baris ke-1 adalah **8**, **K**, **:**, **?** dengan nilai desimal **56**, **75**, **58**, **63**. Hasil enkripsi baris ke-1 digunakan sebagai plainteks baris kedua (i = 2), maka penyelesain enkripsi baris kedua :

M_{22} = ($M_{[2-1]1}$ + (**1010** * **R[2]**)) **Mod 255**
= (**K** + (**1010** * (**2**))) **Mod 255**
= (**75** + **2020**) **Mod 255** = **55**
(huruf **7** dalam karakter ASCII)

M_{23} = ($M_{[2-1]3}$ + (**1010** * **R[2]**)) **Mod 255**
= (**:** + (**1010** * (**2**))) **Mod 255**
= (**58** + **2020**) **Mod 255** = **38**
(huruf **&** dalam karakter ASCII)

M_{24} = ($P_{[2-1]4}$ + (**1010** * **R[2]**)) **Mod 255**
= (**7** + (**1010** * (**2**))) **Mod 255**
= (**63** + **2020**) **Mod 255** = **43**
(huruf **+** dalam karakter ASCII)

Maka diperoleh hasil enkripsi baris kedua (i = 2) adalah **7, &, +** dengan nilai desimal **55**, **38**, **43**.Lalu untuk perhitungan baris selanjutnya, prosesnya sama dengan pada baris kedua, hingga terbentuk hasil seperti pada tabel berikut:

Tabel 2. Proses Enkripsi *TCC*

BUDI →sebagai *plaintext*								
Ciphertext				Hasil enkripsi pada		M_{ij}	Nilai	Nilai
				i	j=(N+i)-N		Karakter	Desimal
8	**K**	**:**	**?**	1	(4+1)-4=1	M_{11}	**8**	56
	7	**&**	**+**	2	(4+2)-4=2	M_{12}	**7**	55
		■	**■**	3	(4+3)-4=3	M_{13}	**■**	8
			ä	4	(4+4)-4=4	M_{14}	**ä**	228
87■ä				→ hasil enkripsi segitiga pertama				

Dapat dilihat dari tabel di atas hasil proses enkripsi segitiga pertama adalah **8**,**7**,**■**, **ä** dengan nilai desimal **56**, **55**, **8**, **228**.

b. Matriks enkripsi segitiga kedua

Plaintext :

KARAKTER ` 8 7 ■ ä
NILAI DESIMAL 56 55 8 228

Maka penyelesain untuk baris pertama (i = 1):

M_{11} = ($P_{[1]}$ + (**1010** * **R[1]**)) **Mod 255**
= (**8** + (**1010** * (**1**))) **Mod 255**
= (**56** + **1010**) **Mod 255** = **46**
(huruf **.** dalam karakter ASCII)

M_{12} = ($P_{[2]}$ + (**1010** * **R[1]**)) **Mod 255**
= (**7** + (**1010** * (**1**))) **Mod 255**
= (**55** + **1010**) **Mod 255** = **45**
(huruf **-** dalam karakter ASCII)

M_{13} = ($P_{[3]}$ + (**1010** * **R[1]**)) **Mod 255**
= (**■**+ (**1010** * (**1**))) **Mod 255**
= (**68** + **1010**) **Mod 255** = **58**
(huruf **ý** dalam karakter ASCII)

M_{14} = ($P_{[4]}$ + (**1010*** **R[1]**)) **Mod 255**
= (**ä** + (**1010** * (**1**))) **Mod 255**
= (**228** + **1010**) **Mod 255** = **218**
(huruf **Ú** dalam karakter ASCII)

Hasil enkripsi baris pertama (i = 1) adalah -ýÚ dengan nilai desimal 46, 45, 253, 218.

penyelesain untuk baris kedua sebagai berikut:

i = 2; j≤(4+1)-2 j ≤ 3.

M_{21} = ($P_{[2-1]1}$ + (**1010** * **R[2]**)) **Mod 255**
= (**.** + (**1010** * (**2**))) **Mod 255**
= (**46** + **2020**) **Mod 255**
= **26** (huruf **■** dalam karakter ASCII)

M_{22} = ($P_{[2-1]2}$ + (**1010** * **R[2]**)) **Mod 255**
= (**-** + (**1010** * (**2**))) **Mod 255**
= (**45** + **2020**) **Mod 255**
= **25** (huruf **■** dalam karakter ASCII)

M_{23} = ($P_{[2-1]3}$ + (**1010** * **R[2]**)) **Mod 255**
= (**ý** + (**1010** * (**2**))) **Mod 255**
= (**253** + **2020**) **Mod 255**
= **233** (huruf **é** dalam karakter ASCII)

Hasil enkripsi baris kedua (i = 2) adalah**■, ■,** é dengan nilai desimal 26, 25, 233. Dan untuk baris selanjutnya, sama prosesnya dengan proses baris kedua.

Tabel 3. Hasil Enkripsi *TCC*

87 ü	→ sebagai *plaintext*							
Ciphertext				Hasil enkripsi pada		M_{ij}	Nilai	Nilai
				i	j=(N+1)-i		Karakter	Desimal
.	-	ý	Ú	1	(4+1)-1=4	M_{14}	Ú	218
■	■	é		2	(4+1)-2=3	M_{23}	é	233
ú	ú			3	(4+1)-3=2	M_{32}	ú	250
Ó				4	(4+1)-4=1	M_{41}	Ó	211
ÓúéÚ				→ hasil enkripsi (*ciphertext*) segitiga kedua				

c. Matriks dekripsi segitiga pertama
Ciphertext :
KARAKTER Ó ú é Ú
NILAIDESIMAL 211 250 233 218
Rumus untuk baris pertama (i = 1) : Mij = C[j] – (K * R[1]) Mod 255, maka penyelesain untuk baris pertama (i = 1):

M_{11} = ($C_{[1]}$ - (1010 * R[1])) Mod 255
= (Ó - (1010 * (1))) Mod 255
= (211 - 1010) Mod 255 = 221
(huruf **Ý** dalam karakter ASCII)

M_{12} = ($C_{[2]}$ - (1010 * R[1])) Mod 255
= (ú - (1010 * (1))) Mod 255
= (250 - 1010) Mod 255 = 5
(huruf ■ dalam karakter ASCII)

M_{13} = ($C_{[3]}$ - (1010 * R[1])) Mod 255
= (é - (1010 * (1))) Mod 255
= (233 - 1010) Mod 255 = 243
(huruf **ó** dalam karakter ASCII)

M_{14} = ($C_{[4]}$ - (1010* R[1])) Mod 255
= (Ú - (1010 * (1))) Mod 255
= (218 - 1010) Mod 255 = 228
(huruf **ä** dalam karakter ASCII)

Hasil dekripsi baris pertama (i = 1) adalah Ý, ■, ó, ä dengan nilai desimal 221, 5, 243, 228. Rumus untuk baris kedua:
Mij = C[i,j] – (K * R[2]) Mod 255,
penyelesaian untuk baris kedua sebagai berikut: i = 2; j≤(4+1)-3 j ≤ 3.

M_{21} = ($M_{[2,1]}$ - (1010 * R[2])) Mod 255
= (Ý - (1010 * (2))) Mod 255
= (221 - 2020) Mod 255 = 241
(huruf **ñ** dalam karakter ASCII)

M_{22} = ($M_{[2,2]}$ - (1010 * R[2])) Mod 255
= (■- (1010 * (2))) Mod 255
= (5 - 2020) Mod 255 = 25
(huruf ■ dalam karakter ASCII)

M_{23} = ($P_{[2,3]}$ - (1010 * R[2])) Mod 255
= (ó - (1010 * (2))) Mod 255
= (243 - 2020) Mod 255 = 8
(huruf ■ dalam karakter ASCII)

Hasil dekripsi baris kedua (i = 2) adalah ñ, ■ , ■ dengan nilai desimal 241, 25, 8. Dan untuk baris selanjutnya, sama prosesnya dengan proses baris kedua.

Tabel 4. Hasil Dekripsi *TCC*

ÓúéÚ	→ sebagai *ciphertext*							
Ciphertext				Hasil dekripsi pada		M_{ij}	Nilai	Nilai
				i	j=(N+1)-i		Karakter	Desimal
Ý	■	ó	ä	1	(4+1)-1=4	M_{14}	ä	228
ñ	■	■		2	(4+1)-2=3	M_{23}	■	8
■	7			3	(4+1)-3=2	M_{32}	7	55
8				4	(4+1)-4=1	M_{41}	8	56
87 ä	→ hasil dekripsi (*ciphertext*) segitiga pertama							

d. Matriks dekripsi segitiga kedua
Ciphertext :
KARAKTER 8 7 ■ ä
NILAI DESIMAL 56 55 8 228

Rumus untuk baris pertama: Mij = C[j] – (K * R[1]) Mod 255, maka penyelesain untuk baris pertama (i = 1):

M_{11} = ($C_{[1]}$ - (1010 * R[1])) Mod 255
= (8 - (1010 * (1))) Mod 255
= (56 - 1010) Mod 255 = 66
(huruf **B** dalam karakter ASCII)

M_{12} = ($C_{[2]}$ - (1010 * R[1])) Mod 255
= (7 - (1010 * (1))) Mod 255
= (55 - 1010) Mod 255 = 65
(huruf **A** dalam karakter ASCII)

M_{13} = ($C_{[3]}$ - (1010 * R[1])) Mod 255
= (■- (1010 * (1))) Mod 255
= (8 - 1010) Mod 255 = 18
(huruf■ dalam karakter ASCII)

M_{14} = ($C_{[4]}$ - (1010* R[1])) Mod 255
= (ä - (1010 * (1))) Mod 255
= (228 - 1010) Mod 255 = 238
(huruf **î** dalam karakter ASCII)

Hasil dekripsi baris pertama (i = 1) adalah B,A, ■ ,î dengan nilai desimal 66, 65, 18, 238. Rumus untuk baris kedua :
Mij = C[i,j] – (K * R[i]) Mod 255, maka penyelesaian untuk baris kedua sebagai berikut: i = 2; j ≥ i → j ≥ 2.

M_{22} = ($M_{[2,2]}$ - (1010 * R[2])) Mod 255
= (A - (1010 * (2))) Mod 255
= (65 - 2020) Mod 255 = 85
(huruf **U** dalam karakter ASCII)

M_{24} = ($M_{[2,4]}$ - (1010 * R[2])) Mod 255
= (î - (1010 * (2))) Mod 255
= (238 - 2020) Mod 255 = 3
(huruf ■ dalam karakter ASCII)

M_{23} = ($P_{[2,3]}$ - (1010 * R[2])) Mod 255
= (■- (1010 * (2))) Mod 255
= (18 - 2020) Mod 255 = 38
(huruf**&** dalam karakter ASCII)

Hasil dekripsi baris kedua (i = 2) adalah U, &, ■ dengan nilai desimal 85, 38, 3. Dan untuk baris selanjutnya, sama prosesnya dengan proses baris kedua, dan untuk hasil akhirnya dapat dilihat pada tabel berikut:

Tabel 5. Hasil Dekripsi *TCC*

87 ä	→ sebagai *ciphertext untuk proses dekripsi segitiga kedua*							
Ciphertext				Hasil dekripsi pada		M_{ij}	Nilai	Nilai
				i	j=(N+i)-N		Karakter	Desimal
B	A	■	î	1	(4+1)-4=1	M_{11}	B	66
	U	&	■	2	(4+2)-4=2	M_{22}	U	85
		D	!	3	(4+3)-4=3	M_{33}	D	68
			I	4	(4+4)-4=4	M_{44}	I	73
BUDI	→ hasil dekripsi (*plaintext*) segitiga kedua							

Maka dari tabel diatas hasil proses dekripsi segitiga kedua adalah BUDI dengan nilai desimal 66, 85, 68, 73. Plaintext yang dihasilkan pada proses dekripsi segitiga kedua adalah plaintext yang sebenarnya.

3 RANCANGAN SISTEM DAN APLIKASI

3.1 Analisa Masalah

Data mengenai transaksi penjualan merupakan data yang sangat penting terutama pada PT.Mandiri Pratama Teknik Wahyu Gemilang. Oleh karena itu,

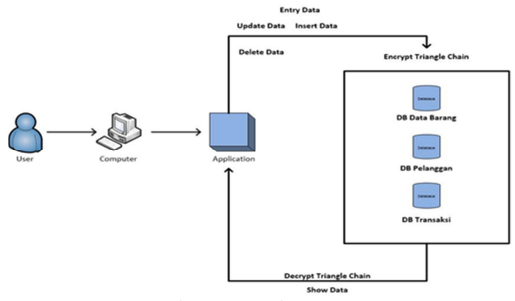
sebuah data harus terjaga kerahasiaannya agar tidak disalah gunakan oleh pihak-pihak yang tidak berkepentingan pada data tersebut. Salah satu cara untuk mengamankan data yaitu dengan mengkodekan data yang disebut kriptografi. Data yang akan masuk kedalam database, terlebih dahulu mengalami proses enkripsi. Dan apabila data dipanggil maka secara otomatis data sudah terdekripsi. Untuk mengimplementasikan kriptografi data dibutuhkan algoritma Kriptografi agar data tersebut bisa dikodekan dan kemudian dikembalikan seperti semula tanpa mengalami perubahan sehingga diperlukan suatu aplikasi yang memberikan dari permasalahan yang ada.

3.2 Solusi Masalah

Pada analisa masalah diatas, untuk memecahkan sebuah masalah maka dibuatlah "Aplikasi Pengamanan Database Menggunakan Algoritma Triangle Chain Berbasis Desktop Pada PT Mandiri Pratama Teknik Wahyu Gemilang". Aplikasi ini nantinya dapat mengamankan sebuah data yang akan masuk kedalam database dengan mengkodekan data yang disebut enkripsi. Dan apabila data dipanggil, secara otomatis ketika data dipanggil terdekripsi. Maka data kembali seperti semula tanpa mengalami perubahan sedikitpun.

3.3 Arsitektur Sistem

Pada gambar 3 menggambarkan arsitektur sistem secara garis besar proses dari keseluruhan sistem**.**



Gambar 3. Arsitektur Sistem

3.4 Spesifikasi Basis Data

Dibawah ini merupakan struktur tabel database yang digunakan dalam pembuatan pada aplikasi ini :

a. Tabel Barang

Tabel 6. Tabel Barang

Field	Type	Length	Keterangan
kdbrg	*varchar*	15	Kode Barang
NmBrg	*varchar*	50	Nama Barang
HrgBeli	*varchar*	15	Harga Beli
HrgJual	*varchar*	50	Harga Jual
Kunci	*varchar*	10	Kunci(*key*)

b. Tabel Detiljual

Tabel 7. Tabel Detiljual

Field	Type	Length	Keterangan
NoJual	*varchar*	7	Kode transaksi penjualan
KdBrg	*varchar*	5	Kode Barang
HrgJual	*varchar*	50	Harga Jual
JmlJual	*varchar*	4	Jumlah Barang yang Terjual
Keuntungan	*varchar*	50	Keuntungan (laba)
Kunci	*varchar*	10	Kunci(*key*)

c. Tabel Pelanggan

Tabel 8. Tabel Pelanggan

Field	Type	Length	Keterangan
KdPlg	*varchar*	50	Kode Pelanggan
NmPlg	*varchar*	100	Nama Pelanggan
Alamat	*varchar*	150	Alamat Pelanggan
Telp	*varchar*	100	Nomor Telepon Pelanggan
Kunci	*varchar*	10	Kunci(*key*)

d. Tabel Kirim

Tabel 9. Tabel Kirim

Field	Type	Length	Keterangan
nosj	*varchar*	7	Nomor surat jalan
kdbrg	*varchar*	5	Kode barang
jmlkirim	*interger*	11	Jumlah barang yang dikirm
kunci	*varchar*	255	Kunci(*key*)

e. Tabel Penjualan

Tabel 10. Tabel Penjualan

Field	Type	Length	Keterangan
NoJual	varchar	7	Kode transaksi penjualan
TglJual	varchar	50	Tanggal Penjualan
KdPlg	varchar	50	Kode Pelanggan
kunci	varchar	4	Kunci(key)

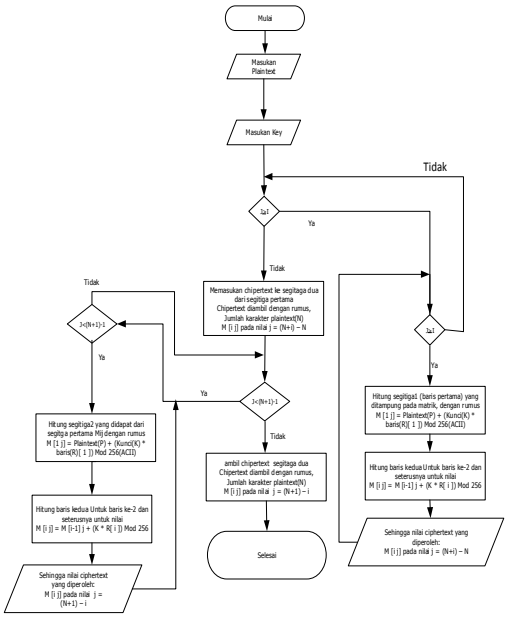
f. Tabel Surat Jalan

Tabel 11. Tabel Surat Jalan

Field	Type	Length	Keterangan
nosj	varchar	7	Nomor surat jalan
tglsj	date	-	Tanggal surat jalan
nojual	varchar	7	Kode transaksi penjualan
kunci	varchar	4	Kunci(key)

3.5 Flowchart Enkripsi Algoritma Triangle Chain

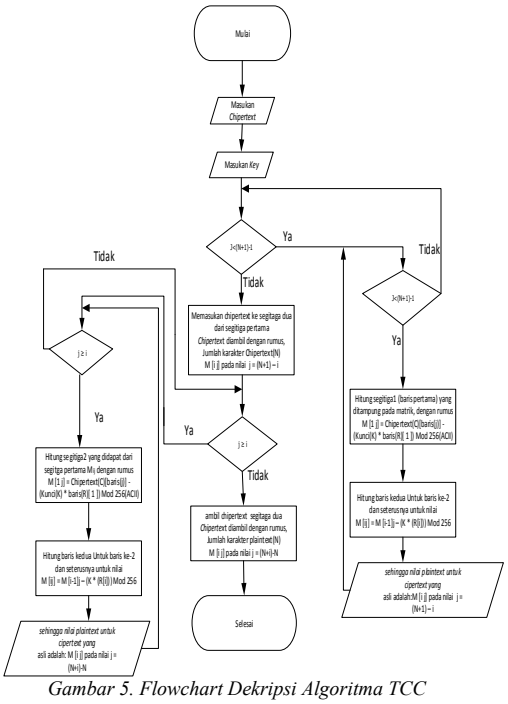
Dalam *flowchart* ini menjelaskan bagaimana proses cara kerja untuk menghasilkan ciperteks pada algoritma *Triangle Chain*



Gambar 4. Flowchart Enkripsi Algoritma TCC

3.6 Flowchart Dekripsi Algoritma Triangle Chain

Dalam *flowchart* ini menjelaskan bagaimana proses cara kerja untuk menghasilkan plainteks pada algoritma *Triangle Chain*.



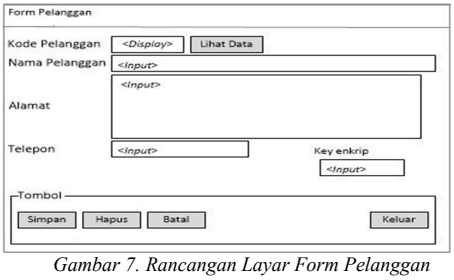
Gambar 5. Flowchart Dekripsi Algoritma TCC

3.7 Rancangan Aplikasi

Berikut ini merupakan tampilan rancangan layar halaman utama ketika *user* berhasil *login*

Gambar 6. Rancangan Layaran Halaman Utama

Didalam halaman utama terdapat banyak menu untuk menginput data yang akan di update ke dalam database.



Gambar 7. Rancangan Layar Form Pelanggan

Pada rancangan layar *form* pelanggan ini, *user* menginput data dari pelanggan yang nantinya data tersebut akan diupdate ke dalam *database*.

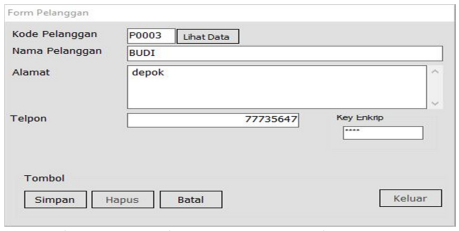
4 HASIL DAN PEMBAHASAN

4.1 Implementasi Aplikasi

Pengujian sistem ini bertujuan untuk mengetahui seberapa efektif dan aman enkripsi yang dilakukan pada record database. Pada ujicoba simulasi untuk mengetahui apakah sistem dapat bekerja dengan baik atau tidak, dan hasilnya apakah sudah sesuai dengan keinginan atau masih perlu dilakukan perbaikan. Hal-hal tersebut akan diketahui jika sudah dilakukan simulasi ujicoba system.

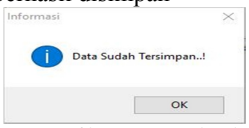
Gambar 8. Tampilan Layar Form Menu Utama

Berikut ini adalah tampilan form pelanggan yang telah diisi kolom yang tersedia, jangan lupa untuk memasukkan *key* enkrip sebelum menekan tombol simpan.



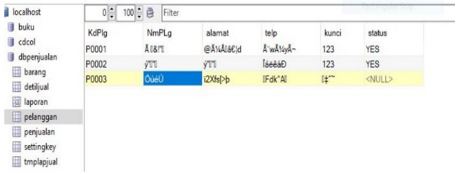
Gambar 9. Tampilan Layar Form Pelanggan yang Siap Untuk disimpan

Jika data berhasil di input, maka akan muncul notifikasi data berhasil disimpan



Gambar 10. Notifikasi Data Berhasil Disimpan

Data yang berhasil disimpan tadi akan tersimpan didalam database dengan kondisi terenkripsi.



Gambar 11. Hasil Enkripsi Data Pelanggan Pada Database

5 KESIMPULAN

Dari hasil analisa yang telah penulis lakukan terhadap masalah dan aplikasi yang dikembangkan, maka didapatkan suatu kesimpulan, sebagai berikut:

Dengan mengimplementasikan penerapan algoritma *Triangle Chain* pada aplikasi pengamanan *database* berbasis dekstop dengan bahasa pemrograman *Visual Basic*, sistem dapat mengamankan data mengenai penjualan atau informasi yang ada di PT Mandiri Pratama Teknik Wahyu Gemilang agar dapat lebih terjaga kerahasiaan data - datanya dari orang-orang yang tidak bertanggung jawab. Aplikasi ini mampu mengamankan data yang masuk kedalam database dengan teknik kriptografi *Triangle Chain,* sehingga data yang tersimpan kedalam database akan sulit untuk dibaca.

6 DAFTAR PUSTAKA

[1] Donny, Ariyus. 2006. Kriptografi Keamanan Data dan Komunikasi. Yogyakarta: Graha Ilmu.

[2] Donny, Ariyus. 2008. Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi. Yogyakarta: C.V Andi Offset.

[3] Hondro, Rivalry Kristanto dan Gunado Widi Nurcahyo. 2014. Analisis dan Perancangan Sistem yang Menerapkan Algoritma Triangle Chain (TCC) Untuk Enkripsi Record Tabel Database. Padang: Jurnal Teknologi Informasi dan Komputer. Vol. 3, No. 2:118-127.

[4] Jogiyanto, 2006. Analisis dan Desain Sistem Informasi. Yogyakarta: Penerbit Andi.

[5] Kromodimoeljo, Sentot. 2011. Teori dan Aplikasi Kriptografi, Jakarta: SPK IT Consulting.

[6] Sonali, J. 2016. 'Integrating Encrypted Cloud Database Service Using Query Processing', International Journal of Computer Application, 148(12), pp. 22- 30.