Jurnal CyberTech
Vol.x. No.x, April 202x, pp. xx~xx
P-ISSN : 9800-3456
E-ISSN : 2675-9802

Penerapan Algoritma Rivest Code 4 Untuk Kamanan Data Keuangan Pada PT. Panca Pilar Tangguh

Ari Marlina*, Nurcahyo Budi Nugroho, S.Kom., M.Kom., Jufri Halim, S.E., M.M.**
* Program Studi Mahasiswa, STMIK Triguna Dharma
** Program Studi Dosen Pembimbing, STMIK Triguna Dharma

Article Info
Article history:

Keyword:
Kriptografi, Rivest Code 4, Keamanan Data, Data Keuangan.

ABSTRACT

Teknologi komputer saat ini mengalami perubahan yang sangat signifikan dan terus berkembang. Sehingga sudah banyak sistem yang digunakan oleh perusahaan-perusahaan untuk mempermudah pekerjaan mereka. Namun, dalam setiap perusahaan pasti selalu ada oknum yang berpikiran untuk mengganti data dalam sistem untuk kepentingannya sendiri yang dapat merugikan perusahaan. Jadi dibutuhkan sebuah metode untuk mengamankan data tersebut agar data tersebut tetap aman. PT. Panca Pilar Tangguh adalah salah satu perusahaan yang harus menjaga data keuangannya.

Dalam hal ini diperlukan sebuah sistem dalam pengamanan data yang dapat melakukan penyandian atau penyembunyian pesan yang berbasis komputer. Pengamanan ini dilakukan dengan menerapkan sebuah algoritma kriptografi yang bertujuan untuk mengenkripsi dan dekripsi sebuah pesan text. Algoritma kriptografi yang digunakan algoritma RC4 (Rivest Code 4).

Hasil pengujian menunjukkan bahwa sistem keamanan data keuangan dapat mengamankan data keuangan dengan sangat baik dan menghindari terjadinya penyalahgunaan atau manipulasi data oleh orang-orang yang tidak memiliki wewenang atas data tersebut.

Corresponding Author:
Nama : Ari Marlina
Program Studi: SistemInformasi
STMIK Triguna Dharma
Email : Arimarlina@gmail.com

1. PENDAHULUAN

Masalah privasi dan keamanan pesan, data, maupun informasi menjadi isu yang tidak ada habis-habisnya dibahas dari dulu hingga sekarang. Teknologi komputer saat ini mengalami perubahan yang sangat signifikan dan terus berkembang. Banyak kejahatan *cyber* yang pernah kita dengar dari media massa, pelaku memanfaatkan celah dari keamanan yang ada dalam sistem berbasis komputer. Kita sering mendengar kasus-kasus seperti penyadapan percakapan penting, pembobolan informasi sensitif, penyadapan surat-surat penting di perusahaan dan lain-lain. Seperti kasus *Wikileaks* yang menghebohkan baru-baru ini. Dimana dokumen-dokumen rahasia dari berbagai negara dan berbagai perusahaan dibocorkan ke publik melalui situs webnya[1]. Kasus tersebut, memberikan pesan moral kepada kita bahwa keamanan pesan, data maupun informasi yang besifat rahasia bukan lagi menjadi sebuah pelengkap pada era digital saat ini, tetapi sudah menjadi kebutuhan utama .

Pada kondisi diatas, menyebabkan perusahaan-perusahaan harus mempersiapkan untuk melakukan pengamanan data, pesan maupun informasi yang bersifat rahasia khususnya data keuangan. Salah satu perusahaan yang memerlukan pengamanan data adalah PT. Panca Pilar Tangguh. PT. Panca Pilar Tangguh merupakan salah satu

perusahaan yang bergerak dalam bidang penyaluran (distributor) berbagai macam produk dari pihak principal kepada konsumen

Kriptografi merupakan salah satu teknik kemananan data informasi yang dimana bekerja merubah data menjadi kode yang rumit dan susah dipahami untuk mencegah pencurian pesan. Dalam ilmu kriptografi terdapat beberapa algoritma yang dapat kita kuganakan dalam penyandian, salah satunya adalah algoritma RC4 (Rivest Code 4). Algoritma kriptografi RC4 umumnya dinyatakan sangat aman dan diterapkan secara luas pada sejumlah aplikasi. Metode ini melakukan penyandian dengan melakukan operator logika XOR antara kunci dengan Plaintext. Untuk mewujudkan penelitian ini akan dirancang aplikasi berbasis dekstop.

Berdasarkan latar belakang permasalahan di atas untuk mengembangkan sistem informasi pada PT. Panc Pilar Tangguh dalam keamanan data menggunakan Kriptografi Algoritma RC4 (Rivest Code 4) ,maka diangkat sebuah penilitian berjudul "Penerapan Algoritma Rivest Code 4 Untuk Keamanan Data Keuangan Pada Panca Pilar Tangguh".

2. KAJIAN PUSTAKA

2.1 Keamanan Data Keuangan

Menjaga integritas dan keamanan data merupakan pencegahan atas keamanan data yang tersimpan diluar agar tidak hilang, rusak, dan diakses oleh pihak yang tidak berkepentingan.

Sebuah bisnis yang menyediakan layanan jasa atau barang kepada pelanggan tentu bekerja dengan berbagai macam jenis data. Salah satunya adalah data keuangan. Data ini bersifat sensitif dan rahasia yang harus diamankan dengan cara terbaik untuk mencegah terjadinya kebocoran data yang dapat memengaruhi nama baik dalam sebuah bisnis.

2.2 Kriptografi

Menurut Munir, R.,2019, kata kriptografi berasal dari bahasa Yunani yang terdiri dari dua kata yaitu *cryptos* dan *graphein*. Kata *crypto* (*secret*) berarti rahasia sedangkan *graphein* (*writing*) berarti tulisan. Kriptografi juga dapat diartikan sebagai ilmu yang mempelajari teknik-teknik matematika untuk menjaga keamanan pesan ataupun informasi dengan cara menyandikannya ke dalam bentuk yang tidak dapat dipahami lagi maknanya.

Sistem kriptografi adalah sebuah kumpulan sistem yang terdiri dari algoritma deskripsi, algoritma enkripsi, ruang kunci, semua plainteks dan cipherteks. Ada dua macam sistem kriptografi, yaitu sistem kriptografi kunci-simetri (*symmetric-key cryptosystem*) dan sistem kriptografi kunci-publik (*public- key cryptosystem*).

2.2.1 Kriptografi Kunci-Simetri (*Symmetric-Key Cryptosystem*)

Pada kriptografi kunci simetris proses enkripsi maupun dekripsi pesan rahasia menggunakan kunci yang sama. Jadi sebelum melakukan pengiriman pesan rahasia, pengirim dan penerima harus memilih suatu kunci tertentu yang sama untuk dipakai bersama dalam proses enkripsi dan dekripsi. Keamanan kriptografi dengan teknik ini terletak pada kerahasiaan kunci[2]. Keuntungan dari kriptografi asimetris adalah tidak ada kebutuhan untuk mendistribusikan kunci rahasia, kunci dapat dikirim ke penerima melalui saluran yang sama dengan saluran yang digunakan untuk mengirimkan.

Contoh algoritma kriptografi modern yang termasuk dalam kelompok kriptografi simetris yaitu : *DES, Blowfish, Twofish, Triple-DES, IDEA, Serpent, Mars, RC4, RC5, RC6, AES*, dan masih banyak lagi[1].

2.2.2 Kriptografi Kunci-Publik (*Public- Key Cryptosystem*)

Pada kriptografi kunci asimetris proses enkripsi dan dekripsi menggunakan kunci yang berbeda. Kunci untuk proses enkripsi menggunakan kunci publik (*public key*), sedangkan kunci untuk proses dekripsi menggunakan kunci rahasia (*private key*).

Terdapat kelebihan dan kelemahan dari kriptografi kunci simetris (*symmetric key cryptosystem*) adalah sebagai berikut :

1. Kelebihan *Symetric Cryptosystem* adalah[3] :
   a. Proses enkripsi atau dekripsi *Symetric Cryptosystem* membutuhkan waktu yang singkat.
   b. Ukuran kunci simetri relatif lebih pendek
   c. Otentikasi pengiriman pesan langsung diketahui dari cipherteks yang diterima, karena kunci hanya diketahui oleh penerima dan pengirim saja.
2. Kekurangan dari *Symetric Cryptosystem* adalah:

a. Kunci simetris harus dikirim melalui saluran komunikasi yang aman dan kedua entitas yang berkomunikasi harus menjaga kerahasian kunci.

Kunci harus sering diubah, setiap kali melaksaanakan komunikasi

Contoh algoritma kriptografi asimetris adalah *RSA, Elgamal, DSA, Diffie-Hellman, Elliptic Curve Cryptography*, dan lain-lain [2]. Algoritma kriptografi lainnya yang digunakan dalam mengamankan data adalah *Caesar Cipher, Vigeneere Cipher, Affine Cipher,* GOST, dan lain-lain[4].

2.3 Algoritma RC4

RC4 merupakan jenis aliran kode yang operasi enkripsinya dilakukan karakter 1 *byte* untuk sekali operasi. Faktor utama yang menjadi kesuksesan dari RC4 adalah kecepatan dan dan kemudahan dalam melakukan pengembangan sehingga memudahkan untuk mengimplementasikan[5].

Algoritma *RC4* menggunakan dua buah *S-Box* yaitu *array* sepanjang 256 yang berisi permutasi dari bilangan 0 sampai 255, dan *Sbox* kedua, yang berisi permutasi merupakan fungsi dari kunci dengan panjang yang variabel. Cara kerja algoritma *RC4* yaitu inisialisasi *S-Box* pertama, S[0],S[1],...,S[255], dengan bilangan 0 sampai 255. Pertama isi secara berurutan S[0] = 0, S[1] = 1,...,S[255] = 255. Kemudian inisialisasi *array* lain (*S-Box* lain), misal *array* K dengan panjang 256. Isi *array* K dengan kunci yang diulangi sampai seluruh *array* K[0], K[1],...,K[255] terisi seluruhnya [6].

Dalam algoritma enkripsi metode ini akan membangkitkan *pseudo ramdom* yang nanti dimana *byte* dari *key* akan dihadapkan pada operasi XOR pada plainteks dalam menghasilkan cipherteks[7]. Secara umum algoritma ini merupakan salah satu bagian dari metode algoritma *Rivest Code* 4, yang dimana sudah dibagi menjadi 2 yaitu *Key Schedulling Algorithm* (*KSA*) dan *stream generation* atau *Pseudo Random Generation Algorithm* (*PRGA*).

2.3.1 *Key Setup / Key Schedulling Algorithm*

Key Setup sendiri memiliki beberapa tahap dalam memproses suatu operasi ialah sebagai berikut :

1. Inisialisasi S-Box

Inisialiasasi S-Box yang nanti akan memasukkan data dengan nilai yang telah disesuaikan dengan indeks untuk mendapatkan hasil dari *table* S-Box.

2. Key Byte Array

Pada tahap kedua yaitu *key byte array* ialah dimana kunci akan dapat digunakan untuk melakukan proses enkripsi dan dekripsi yang nanti akan dimasukkan agar *array* menjadi ukuran maksimal yaitu 255. Secara berulang sampai seluruh *array* terisi.

3. Perbandingan Sebuah Nilai Dalam Permutasi

Tahap terakhir yang dilakukan ialah tahap perbandingan sebuah nilai dalam permutasi terhadap S-Box. Dimana langkah awal yang dilakukan ialah mengurutkan S(0) = 0, S(1) = 1, …., S(255) = 255. Kemudian isi *array* 256 *byte* yang nanti akan dilakukan seperti *looping* dimana seluruh *array* seperti *array* K(0), K(1), …., K(255) terisi selurusnya. Set indeks j dengan nol.

2.3.2 *Stream Generation* Atau *Pseudo Random Generation Algorithm* (PRGA)

Pada tahapan ini, dimana tahap *stream generation atau pseudo random generation algorithm* akan menghasilkan suatu *pseudo random* yang nanti akan dilakukan suatu operasi XOR dalam menghasilkan cipherteks dan juga dapat menghasilkan plainteks[7]. Dimana pada tahap ini ialah merupakan tahap awal pada proses *stream generation* atau juga bisa disebut *pseudo random generation algorithm*. Berikut adalah algoritmanya [8] :

1. Isi indeks I dengan nilai 0, dan indeks J isi juga dengan nilai 0
2. Pada i=0 sampai i=panjang *ciphertext*, panjang *ciphertext* = panjang *plaintext*
3. Isi nilai I beserta hasil dari operasi (i+1) mod 256
4. Sedangkan nilai j dengan hasil dari operasi (j+S(i)) mod 256
5. Tukar nilai antara S(j) dan S(i)
6. Masukan nilai t beserta hasil dari operasi (S(i)+(S(j) mod 256)) mod 256
7. Masukan nilai y dan nilai S(t)
8. Nilai y digunakan pada operasi XOR kepada *ciphertext*
9. Jumlahkan i dan 1, dan akan kembali pada point 2

3 ANALISA DAN HASIL

3.1 Algoritma Sistem

RC4 menggunakan panjang kunci dari 1 sampai 256 *byte.* Tabel ini digunakan untuk generasi yang berikut dari *pseudo random* yang menggunakan XOR dengan *plaintext* untuk menghasilkan *chipertext.* Masing-masing elemen dalam tabel saling ditukarkan minimal sekali.

1. Proses inisialisasi S-BOX (*Array S)*
   for i = 0 to 255, S[i] = i
2. Proses inisialisasi S-BOX (*Array K)*
   for i = 0 to 255, K[i] = K[i]
3. Kemudian lakukan langkah pengacakan S-BOX
   i = 0; j = 0
   for i = 0 to 255
   j = (j+S[i]+K[i]) mod 256
   swap S[i] dan S[j]
4. Membuat *pseudo random byte*
   i = (i+1) mod 256
   j = (j+S[i]) mod 256
   swap S[i] dan S[j]
   t = (S[i]+S[j]) mod 256
   K = S[t]
   Byte K di-XOR-kan dengan *plaintext* untuk menghasilkan *chipertext* atau di-XOR-kan dengan *chipertext* untuk menghasilkan *plainteks.*

Berikut adalah implementasi algoritma RC4 dengan mode 256 *byte* :

1. Inisialisasi S-BOX dengan panjang 256 *byte*, dengan S[0]=0, S[1]=1, S[2]=2, S[3]=3, ..., S[255]=255 sehingga *array* S menjadi :

Tabel 3.1 Tabel Inisialisasi S-BOX Dengan Panjang 256 *Byte*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

2. Inisialisasi 20 *byte* kunci *array K-i*, misalkan kunci terdiri dari 20 *byte* yaitu “KEUANGANPTPPT2019RC4” maka kalimat akan diubah kedalam bentuk desimal “75 69 85 65 78 71 65 78 80 84 80 80 84 50 48 49 57 82 67 52”. Ulangi kunci sampai memenuhi seluruh *array K* sehingga *array K* menjadi :

Tabel 3.2 Tabel Inisialisasi 20 *byte* Kunci *Array* K-i

Iterasi-i	Key-char	Key[i]	Sbox[i]
1	K	75	0
2	E	69	1
3	U	85	2
4	A	65	3
5	N	78	4
6	G	71	5
7	A	65	6

Tabel 3.2 Tabel Inisialisasi 20 *byte* Kunci *Array* K-i

Iterasi-i	Key-char	Key[i]	Sbox[i]
8	N	78	7
9	P	80	8
10	T	84	9
11	P	80	10
12	P	80	11
13	T	84	12
14	2	50	13
15	0	48	14
16	1	49	15
17	9	57	16
18	R	82	17
19	C	67	18
20	4	52	19
…	…	…	…
…	…	…	…
255	0	48	254
256	1	49	255

3. Berikutnya mencampur operasi dimana akan menggunakan variabel i dan j ke index array S[i] dan K[i]. Pertama kita beri nilai inisisal untuk i dan j dengan 0. Operasi pencampuran adalah pengulangan rumusan (j+S[i]+K[i]) mod 256 yang diikuti dengan penukaran S[i] dengan S[j]. Untuk contoh ini, karena kita menggunakan *array* dengan panjang 256 *byte* maka algoritma menjadi :
   For i = 0 to 256
   j = (j+S[i]+K[i]) mod 256
   swap S[i] dan S[j]

Dengan algoritma diatas maka dengan nilai awal i = 0 sampai i = 255 akan menghasilkan *array S* seperti dibawah ini :

Iterasi ke-1 :
i = 0, maka
j = (j+S[i]+K[i]) mod 256
= (j+S[0]+K[0]) mod 256
= (0+0+75) mod 256
= 75
Swap S[0] dan S[75]

Iterasi ke-2 :
i = 1, maka
j = (j+S[i]+K[i]) mod 256
= (j+S[1]+K[1]) mod 256
= (75+1+69) mod 256
= 145
Swap S[1] dan S[145]

Iterasi ke-3 :
i = 2, maka
j = (j+S[i]+K[i]) mod 256
= (j+S[2]+K[2]) mod 256
= (145+2+85) mod 256
= 232
Swap S[2] dan S[232]

Iterasi ke-4 :
i = 3, maka

j = (j+S[i]+K[i]) mod 256
= (j+S[3]+K[3]) mod 256
= (232+3+65) mod 256
= 44

Swap S[3] dan S[44]

…
…
…

Iterasi ke-256 :
i = 255, maka
j = (j+S[i]+K[i]) mod 256
= (j+S[255]+K[255]) mod 256
= (8+255+49) mod 256
= 56

Swap S[255] dan S[56]

Hasil yang didapat setelah melakukan seluruh iterasi dari 0 s/d 255 kali iterasi dan melakukan pertukaran S-BOX (*swap)* adalah sebagai berikut :

Tabel 3.3 Tabel Hasil Pertukaran S-BOX (*swap)*

93	121	232	44	37	98	17	60	204	43	117	208	71	4	139	237
197	19	227	70	172	18	16	166	48	198	53	9	10	251	95	195
79	196	119	73	58	85	49	200	59	169	181	222	206	66	47	177
132	189	81	76	179	52	151	171	138	167	29	147	102	20	56	38
142	178	127	120	69	5	80	24	214	104	1	91	83	242	3	228
105	7	13	125	14	89	124	97	27	182	78	193	136	75	6	84
116	122	202	63	22	233	144	26	175	149	241	94	112	36	82	135
42	250	244	25	192	62	129	40	46	201	146	140	188	39	111	234
217	213	163	30	243	107	209	155	229	8	0	159	212	156	247	194
65	2	187	114	35	115	33	41	191	15	123	72	128	215	236	185
231	225	108	253	133	219	100	101	210	207	239	137	143	31	168	203
254	12	205	154	161	64	216	199	106	158	96	11	45	109	230	220
87	153	32	74	226	54	162	249	68	61	221	211	190	92	164	90
183	21	23	238	186	55	130	252	126	148	118	131	113	248	51	110
145	152	103	165	150	34	235	57	157	245	218	224	184	223	246	86
88	173	170	176	99	255	67	77	240	28	160	141	134	50	180	174

4. Berikutnya adalah proses enkripsi yaitu meng-XOR-kan *pseudo random byte* dengan *plaintext*, jumlah *plaintext* sama dengan jumlah interasi. Sebelum di-iterasi, ubah karakter menjadi bentuk bilangan biner.

Tabel 3.4 Tabel Biner *Plaintext*

Hexa	Desimal	Karakter	Biner
32	50	2	00110010
30	48	0	00110000
34	52	4	00110100
38	56	8	00111000
35	53	5	00110101
38	56	8	00111000
34	52	4	00110100
32	50	2	00110010

Tabel 3.4 Tabel Biner *Plaintext* (Lanjutan)

Hexa	Desimal	Karakter	Biner
37	55	7	00110111
33	51	3	00110011
35	53	5	00110101

Berikut iterasi 1 : Inisialisasi i dan j dengan i = 0; j = 0;

i = (i+1) mod 256
= (0+1) mod 256
= 1
j = (j+S[i] mod 256
= (j+S[1] mod 256
= (0+93) mod 256
= 93
Swap S[1] dan S[93]
t = (S[i] + S[j] mod 256
= (S[1] + S[145] mod 256
= (145+93) mod 256
= 238
K = S[t] = S[238] = 94 = 01011110

Tabel 3.5 Enkripsi *Plaintext* "2"

Plaintext				Key	XOR	Ciphertext		
Karakter	Desimal	Hexa	Biner			Karakter	Desimal	Hexa
2	50	32	00110010	01011110	01101100	l	108	6C

iterasi 2 :
i = (i+1) mod 256
= (1+1) mod 256
= 2
j = (j+S[i] mod 256
= (j+S[2] mod 256
= (145+196) mod 256
= 85
Swap S[2] dan S[85]
t = (S[i] + S[j] mod 256
= (S[2] + S[85] mod 256
= (49+196) mod 256
= 245
K = S[t] = S[245] = 247 = 11110111

Tabel 3.6 Enkripsi *Plaintext* "0"

Plaintext				Key	XOR	Ciphertext		
Karakter	Desimal	Hexa	Biner			Karakter	Desimal	Hexa
0	48	30	00110000	11110111	11000111	Ç	199	C7

iterasi 3 :
i = (i+1) mod 256
= (2+1) mod 256
= 3
j = (j+S[i] mod 256
= (j+S[3] mod 256
= (85+44) mod 256
= 129
Swap S[3] dan S[129]

t = (S[i] + S[j] mod 256
= (S[3] + S[129] mod 256
= (84+44) mod 256
= 128
K = S[t] = S[128] = 160 = 10100000

Tabel 3.7 Enkripsi *Plaintext* "4"

Plaintext				Key	XOR	Ciphertext		
Karakter	Desimal	Hexa	Biner			Karakter	Desimal	Hexa
4	52	34	00110100	10100000	10010100	"	148	94

5. Berikutnya adalah proses dekripsi yaitu meng-XOR-kan *pseudo random byte* dengan *chipertext,* dan *chipertext* dalam bentuk hexa desimalnya adalah "**6C, C7, 94, 3B, IE, 7F, DC, 68, 3E, C2, D9".** *Ciphertext* terdiri dari 11 karakter maka terjadi 11 iterasi. Sebelum di-iterasi ubah karakter menjadi bentuk bilangan biner.

Tabel 3.8 Tabel Biner *Ciphertext*

Karakter	Desimal	Hexa	Biner
l	108	6C	01101100
Ç	199	C7	11000111
"	148	94	10010100
;	59	3B	00111011
RS	30	1E	00011110
Delete	127	7F	01111111
Ü	220	DC	11011100
h	104	68	01101000
>	62	3E	00111110
Â	194	C2	11000010
Ù	217	D9	11011001

Data dalam bentuk *ciphertext* sehingga setelah disimpan dapat kembali diubah menjadi *plaintext* dengan meng-XOR-kan dengan kunci yang sama. Proses dekripsi *ciphertext* menggunakan algoritma RC4 ini sama untuk proses *key-schedule*-nya. Untuk mendapatkan *plaintext, ciphertext* yang diperoleh di XORkan dengan *pseudo random byte* yang didapat sebelumnya. Maka hasilnya adalah plainteks atau teks asli.

Tabel 3.9 Tabel Dekripsi *Ciphertext*

Iterasi-i	Ciphertext				Key	XOR	Plaintext		
	Karakter	Desimal	Hexa	Biner			Karakter	Desimal	Hexa
1	l	108	6C	01101100	01011110	00110010	2	50	32
2	Ç	199	C7	11000111	11110111	00110000	0	48	30
3	"	148	94	10010100	10100000	00110100	4	52	34
4	;	59	3B	00111011	00000011	00111000	8	56	38
5	RS	30	1E	00011110	00101011	00110101	5	53	35
6	Delete	127	7F	01111111	01000111	00111000	8	56	38
7	Ü	220	DC	11011100	11101000	00110100	4	52	34
8	h	104	68	01101000	01011010	00110010	2	50	32
9	>	62	3E	00111110	01110001	00110111	7	55	37
10	Â	194	C2	11000010	11110001	00110011	3	51	33
11	Ù	217	D9	11011001	11101100	00110101	5	53	35

4 IMPLEMENTASI DAN UJI COBA

4.1 Form Login

Sebelum masuk dan mengakses aplikasi, terlebih dahulu Bendahara harus melakukan *Login* dengan cara menginput *Username* dan *Password* dengan benar sesuai dengan sistem *database*. Berikut ini adalah tampilan *form login* dari aplikasi :

Gambar 4.1 Tampilan *Form* Login

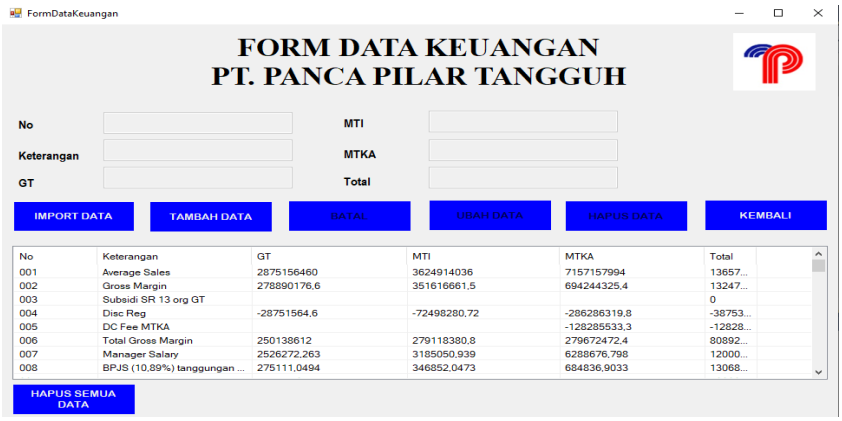
4.2 Tampil Form Menu Utama

Setelah melakukan login, maka akan muncul tampilan form menu utama. Dimana form ini berisi 4 tombol, yaitu Data Keuangan, Enkripsi, Dekripsi, About Us. Berikut adalah tampilan form menu utama :

Gambar 4.2 Tampilan *Form* Menu Utama

4.3 Tampilan Form Data Keuangan

Form data keuangan ini berisi tentang data keuangan yang dapat di import, ubah, tambah, hapus yang akan tersimpan pada database. Berikut adalah tampilan form data keuangan :



No	Keterangan	GT	MTI	MTKA	Total
001	Average Sales	2875156460	3624914036	7157157994	13657...
002	Gross Margin	278890176,6	351616661,5	694244325,4	13247...
003	Subsidi SR 13 org GT				0
004	Disc Reg	-28751564,6	-72498280,72	-286286319,8	-38753...
005	DC Fee MTKA			-128285533,3	-12828...
006	Total Gross Margin	250138612	279118380,8	279672472,4	80892...
007	Manager Salary	2526272,263	3185050,939	6288676,798	12000...
008	BPJS (10,89%) tanggungan ...	275111,0494	346852,0473	684836,9033	13068...

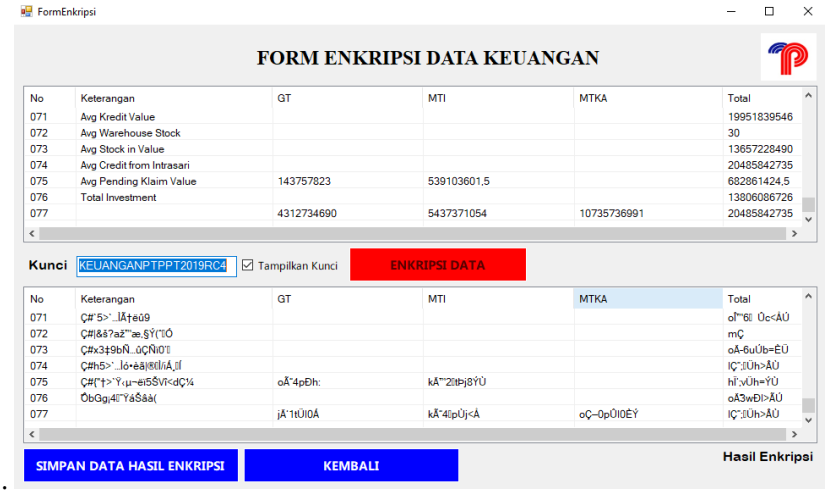
Gambar 4.3 Tampilan *Form* Data Keuangan

4.4 Pengujian

Setelah dilakukan perancangan dan pembuatan sistem, maka akan dilakukan pengujian pada sistem. Dimana pengujian ini dapat memberitahu apakah sistem bisa berjalan dengan yang diharapkan atau tidak, pengujian ini berisi tentang Pengamanan Data Keuangan Pada PT. Panca Pilar Tangguh.

1. *Form* Enkripsi

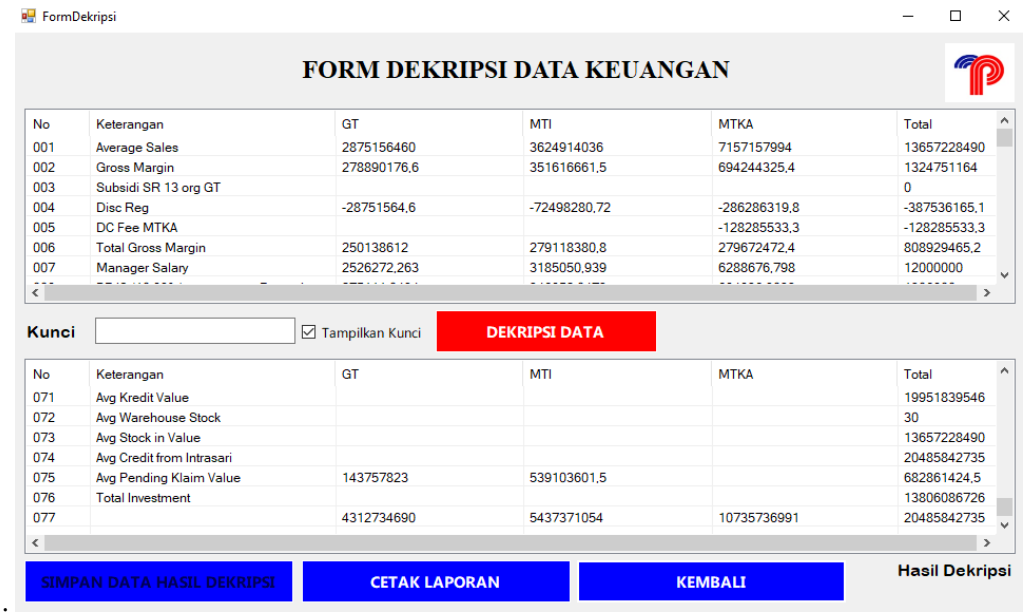
Proses ennkripsi merupakan proses perubahan dimana data yang masih bersifat asli kemudian dirubah kedalam bentuk sandi. Proses ini dilakukan setalah bendahara sudah melakukan penginputan data gaji dan kunci pada textbox yang sudah tersedia



Gambar 4.4 Tampilan *Form* Enkripsi

2. *Form* Dekripsi Data

Proses Deskripsi merupakan proses perubahan dari data yang tersandi dirubah kembali menjadi data yang bisa dibaca (asli)



Gambar 4.5 Tampilan *Form* Dekripsi

5. KESIMPULAN

Berdasarkan dari penelitian yang sudah dilakukan mengenai perancangan perangkat lunak, ada kesimpulan yang diambil sebagai berikut :

1. Mengamankan data keuangan menggunakan algoritma RC4, dengan melakukan enkripsi dan dekripsi terlebih dahulu pada data keamanan. Data keamanan yang sudah di enkripsi menjadi sebuah sandi dapat di dekripsi menjadi data keuangan asli kembali.
2. Data yang di input akan di simpan pada database dalam keadaan terenkripsi sehingga keamanan dan kerahasiaan datanya dapat terjaga
3. Perancangan sistem data keuangan dapat dilakukan dengan menggunakn bahasa pemrograman berbasis *desktop* guna mempermudah proses mengamankan data keuangan pada PT. Panca Pilar Tangguh.

UCAPAN TERIMA KASIH

Syukur Alhamdulillah saya ucapkan kehadirat Allah Subhanahu Wa Ta'ala atas rahmat dan hidayah-Nya serta memberi saya kesempatan dalam menyelesaikan jurnal ilmiah ini dengan baik. Ucapan terima kasih yang besar ditujukan untuk kedua orang tua, yang telah mengasuh, membesarkan dan selalu memberikan doa, motivasi serta pengorbanan baik bersifat moril maupun materil yang tidak terhingga selama menjalani pendidikan. Ucapan terima kasih yang sebesar-besarnya juga ditujukan terutama kepada Bapak Rudi Gunawan, SE., M.Si., selaku Ketua Sekolah Tinggi Manajemen Informatika Dan Komputer (STMIK) Triguna Dharma Medan. Bapak Zulfian Azmi, ST., M.Kom., selaku Wakil Ketua I Bidang Akademik STMIK Triguna Dharma Medan. Bapak Marsono, S.Kom., M.Kom.,selaku Ketua Program Studi Sistem Informasi STMIK Triguna Dharma Medan. Bapak Nurcahyo Budi Nugrogo, S.Kom., M.Kom., selaku Dosen Pembimbing I yang telah meluangkan waktu untuk membimbing dan memberikan arahan kepada saya sehingg penelitian ini dapat terselesaikan dengan baik dan tepat waktu. Bapak Jufri Halim, S.E., M.M., selaku Dosen Pembimbing II yang telah memberikan bimbingan tata cara penulisan, saran sehingga penelitian ini dapat terselesaikan dengan baik dan tepat waktu. Seluruh Staff Karyawan di STMIK Triguna Dharma yang menuntun saya selama mengikuti perkuliahan sampai dengan selesai.

REFERENSI

[1] Rinaldi Munir, *Kriptografi*, Kedua. Bandung: Informatika Bandung.
[2] Martono, "Model Modifikasi Kriptografi Algoritma Rsa Untuk Keamanan Data Pada Database E-Voting," *J. Ilm. Media Sisfo*, vol. 11, no. 2, pp. 896–910, 2017.
[3] Sumarno, I. Gunawan, H. S. Tambunan, and E. Irawan, "Analisis Kinerja Kombinasi Algoritma Message-Digest Algortihm 5 (Md5), Rivest Shamir Adleman (Rsa) Dan Rivest Cipher 4 (Rc4) Pada Keamanan E-Dokumen," *JUSIKOM PRIMA (Junal Sist. Inf. Ilmu Komput. Prima)*, vol. 2, no. 1, pp. 41–48, 2018.
[4] M. Diana and T. Zebua, "Optimalisasi Beaufort Cipher Menggunakan Pembangkit Kunci RC4 Dalam Penyandian SMS," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 2, no. 1, p. 12, 2018, doi: 10.30645/j-sakti.v2i1.52.
[5] F. Danil, Z. Rohman, and Mufti, "Implementasi Kriptografi Pada Pngiriman Pesan Email Dengan Menggunakan Mtode RC4 Dan Blowfish Berbasis Web Paa PT.Dascom Jaya Sakti," vol. 1, no. 2, pp. 788–793.
[6] R. Y. Rifai, Y. Christyono, and I. Santoso, "Implementasi Algoritma Kriptografi Rivest Code 4, Rivest Shamir Adleman, Dan Metode Steganografi Untuk Pengamanan Berkas Rahasia Pada Berkas Teks Digital," *TRANSIENT J. IlmiahTeknik Elektro*, vol. 5, no. 1, pp. 86–91, 2016.
[7] A. R. Wahid *et al.*, "Implementasi Algoritma Rc4 Dan Kompresi Lzw Untuk Pengamanan Database Pada Pt . Mpp International," vol. 1, no. 3, pp. 1045–1050, 2018.
[8] W. E. Winanto, "Aplikasi Keamanan Email Data Produksi PT Kunyun Gravure Industries Indonesia Dengan RC4 Dan Base64," vol. 1, no. 1, pp. 303–308, 2018.

BIBLIOGRAFI PENULIS

	**Ari Marlina** Wanita kelahiran Grobogan, 27 Juni 1999, Mempunyai pendidikan Taman Kanak-kanak TK MUSLIMAT NU KALIWEDI tamat tahun 2004, kemudian melanjutkan pendidikan Sekolah Dasar SD N 1 KALIWEDI tamat tahun 2010, kemudian melanjutkan pendidikan Sekolah Menengah Pertama MTs YPI KLAMBU tamat tahun 2013, kemudian melanjutkan pendidikan Sekolah Menengah Kejuruan SMK N 3 KUDUS tamat tahun 2016. Saat ini menempuh pendidikan Strata Satu (S-1) di SMTIK Triguna Dharma Medan mengambil jurusan Program Studi Sistem Informasi. E-mail Arimarlina72@gmail.com
	**Nurcahyo Budi Nugroho, S.Kom., M.Kom.** Beliau merupakan dosen tetap STMIK Triguna Dharma, serta aktif sebagai dosen pengajar khusus pada bidang ilmu Sistem Informasi.
	**Jufri Halim, S,E., M.M.** Beliau merupakan dosen tetap STMIK Triguna Dharma, serta aktif sebagai dosen pengajar khusus pada bidang ilmu Sistem Informasi.