Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume (2) No.1 Maret 2018, pp. 23-32
ISSN:2548-9771/EISSN:2549-7200
http://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Implementasi Algoritma Data *Encryption Standard* Pada Penyandian *Record Database*

Neti Rusri Yanti[1], Alimah[2], Desi Afrida Ritonga[3]
[1,2,3]Mahasiswa Prorgram Studi Teknik Informatika STMIK Budidarma Medan
[1,2,3]Jln. Sisingamangaraja No. 338 Sp. Limun Medan
[1]netiry95@gmail.com, [2]etekalimah@gmail.com, [3]desiafrida46@gmail.com

Abstract

Record databases are generally still often displayed in text form as information for users, so it can facilitate cryptanalyst to access and provide opportunities to do the leak, distribute or modify the database records. One of the cryptographic algorithms used to secure data is using the DES algorithm to encrypt the data to be stored or sent. The DES algorithm belongs to a cryptographic system of symmetry and is a type of block cipher. DES operates on a 64-bit block size. DES describes 64 bits of plaintext to 64 bits of ciphertext using 56 bits of internal key (internal key) or up-key (subkey). The internal key is generated from an external key 64-bit length. This research describes the process of securing database records by encrypting it based on DES algorithm, resulting in text record databases in the form of passwords that are difficult to understand and understand by others. This is done in an attempt to minimize the misuse of database records.

Keywords: *Cryptography, DES, Database, Record*

Abstrak

Record database umumnya masih sering ditampilkan dalam bentuk teks sebagai informasi bagi pengguna, sehingga dapat mempermudah kriptanalis untuk mengakses serta memberi peluang untuk melakukan pembocoran, mendistribusikan maupun memodifikasi record database tersebut. Salah satu algoritma kriptografi yang digunakan untuk mengamankan data yaitu menggunakan algoritma DES untuk mengenkripsi data yang akan disimpan maupun dikirim. Algoritma DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis cipher blok. DES beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (internal key) atau up-kunci (subkey). Kunci internal dibangkitkan dari kunci eksternal (external key) yang panjangnya 64 bit. Penelitian ini menguraikan proses pengamanan record database dengan menyandikannya berdasarkan algoritma DES, sehingga dihasilkan teks record databases dalam bentuk sandi yang sulit dipahami dan dimengerti oleh orang lain. Hal ini dilakukan sebagai upaya untuk meminimalisir tindakan-tindakan penyalahgunaan record database.

Kata Kunci*: Kriptografi, DES, Database, Record*

1. PENDAHULUAN

Masalah keamanan dan kerahasiaan suatu sangatlah yang penting dalam suatu instansi atau pun perusahaan. Data yang akan digunakan ataupun disimpan agar benar-benar aman secara fisik maupun sistem perlu terlebih dahulu untuk diamankan agar tidak dapat dibaca atau dilacak oleh pihak-pihak yang tidak bertanggung jawab [1].

Database secara umum merupakan susunan atau kumpulan dari *record* data yang disimpan dalam komputer yang saling berhubungan dan dapat dijadikan sebagai salah satu sumber dari sistem informasi yang sedang berjalan sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh pengguna. *Record database* masih sering ditampilkan dalam bentuk teks sebagai informasi bagi pengguna, sehingga dapat mempermudah kriptanalis untuk mengakses serta memberi peluang untuk melakukan pembocoran, mendistribusikan maupun memodifikasi *record database* tersebut [2].

Kriptografi dapat memelihara dan mejaga suatu data agar tetap terpelihara kerahasiaan dan keasliannya melalui tiga aspek yaitu kerahasiaan pesan, keabsahan, keaslian dan ketiadaan penyangkalan. Kemampuan metode kriptografi dalam mengacak isi data, seperti teks, gambar, audio, video dan sebagainya untuk membuat data tidak terbaca, tersembunyi atau berarti semua jalan melalui transmisi atau penyimpanan [3].

Algoritma *Data Encryption Standard* (DES) merupakan algoritma enkripsi yang paling banyak digunakan di dunia yang diadopsi oleh NIST (*National Institue of Standards and Technology*) sebagai standar pengolah informasi Federal AS. *Plain* dienkrip dalam blok-blok 64 bit menjadi 64 bit data *ciphertext* menggunakan kunci 56 bit kunci internal (internal key). DES mentransformasikan *input* 64 *bit* dalam beberapa tahap enkripsi ke dalam *output* 64 *bit*. Dengan demikian, DES termasuk *block cipher*. Berdasarkan tahapan dan kunci yang sama, DES digunakan untuk membalik enkripsi. Kunci internal pada algoritma DES dibangkitkan dari kunci eksternal (*external key*) 64 bit4]. DES beroperasi pada ukuran blok 64 bit dan termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *cipher block*. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal *(internal key)* atau up-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit [5].

Penelitian ini menjabarkan prosedur pengamanan *record database* berdasarkan algoritma DES, dimana *record database* yang diamankan akan dienkripsi dengan 56 bit kunci dan melalui 16 *round* proses untuk menghasilkan *cipher* yang kuat sehingga dapat mempersulit pihak-pihak yang tidak berhak untuk memecahkan *cipher* data tersebut kebentuk aslinya.

2. METODOLOGI PENELITIAN

2.1 Database

Secara umum *database* merupakan susunan atau kumpulan dari *record* data yang disimpan dalam komputer. Keterhubungan antara elemen-eleen dalam database dapat dimanfaatkan sebagai salah satu sumber informasi bagi pengguna. Hingga saat ini, masih banyak *record database* yang masih ditampilkan dalam bentuk teks sebagai informasi bagi pengguna. Hal inilah yang menjadi salah satu celah baggi para kriptanalis untuk dapat mengakses, memanipulasi maupun melakukan pembocoran dan mendistribusikan *record database* tersebut [2].

2.2 Kriptografi

Defenisi sederhana dari kriptografi adalah teknik untuk menjaga kerahasian pesan dengan cara menyandikannya sehingga tidak dapat dimengerti lagi maknanya [6]. Kriptografi memiliki algoritma dalam melakukan proses penyandian suatu agar dapat terjaga keasliannya. Algoritma kriptografi terdiri dari tiga fungsi dasar [7]:

a. Enkripsi
   Enkripsi merupakan istilah lain dari proses menyandikan data penting ke dalam bentuk simbol-simbol yang tidak dapat dimengerti lagi oleh pihak lain sehinggan keaslian dan keamanan data dapat terjaga.
b. Dekripsi
   Dekripsi adalah proses untuk merubah atau mengembalikan data tersandi kebentuk aslinya agar arti data dapat dimengerti oleh penerima.
c. Kunci
   Kunci merupakan elemen yang paling penting dalam mengimplementasikan proses enkripsi dan dekripsi. Keamanan kunci di dalam kriptografi menjadi prioritas karena serumit apapun algoritma yang digunakan akan dapat dipecahkan bila kunci yang digunakan berhasil ditemukan. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).

Ada beberapa aspek keamanan yang harus dicapai pada penerapan teknik kriptografi, yaitu kerahasiaan (*confidentiality*), integritas data *(data integrity*), autentikasi *(authentication*), ketiadaan penyangkalan *(non-repudiation*) [8][11]. Beberapa kekuatan yang dimiliki oleh algoritma kriptografi dalam proses mengenkripsi data2][3], yaitu:

a. Konfusi/pembingungan (*confusion*), yaitu suatu proses dimana teks sulit dikembalikan pada bentuk awal secara tanpa melalui proses dekripsi.
b. Difusi/peleburan *(difusion*), yaitu suatu proses dimana karakteristik suatu teks dihilangkan sehingga mengamankan suatu informasi.

Umumnya algoritma kriptografi *modern* beroperasi dalam mode bit. semua data dan informasi (baik kunci, *plainteks* , maupun *cipher* teks) dalam operasi mode bit dinyatakan dalam rangkaian (*string*) bit biner, 0 dan 1. Algoritma enkripsi dan dekripsi memproses semua data dan informasi dalam bentuk rangkaian bit. Rangkaian bit yang menyatakan *plainteks* dienkripsi menjadi *cipher* teks dalam bentuk rangkaian bit, demikian sebaliknya.

2.3 Algoritma Data Encryption Standard

Algoritma Data Encryption Standard (DES) termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *cipher* blok. DES beroperasi pada ukuran blok 64 *bit* dan mengenkripsikan 64 bit *plainteks* menjadi 64 bit *cipher* teks dengan menggunakan 56 bit kunci internal (internal *key*) atau up-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit. Adapun proses dari algoritma DES [5] adalah :

a. Blok *plainteks* dipermutasi dengan matriks permutasi awal (initial permutation atau IP).

b. Hasil permutasi awal kemudian *dienciphering* sebanyak 16 kali (16 *round*). Setiap putaran menggunakan kunci internal yang berbeda.
c. Hasil *enciphering* kemudian dipermutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP-1) menjadi blok *cipher* teks.

Adapun konsep kerja dari algoritma Data Encryption Standard (DES) [5] adalah sebagai berikut:

a. Blok *plainteks* proses enkripsi dibagi menjadi dua bagian yaitu L[0] dan R[0] yang panjangnya 32 bit. Kemudian masukkan ke dalam putaran 16 *round*.
b. Setiap putaran i, blok R merupakan masukan untuk fungsi transformasi yang disebut f.
c. Fungsi f dalam blok R dikombinasikan dengan kunci internal Ki. Keluaran dari fungsi f di XOR-kan dengan blok L untuk mendapatkan blok R yang baru.
d. Bagian blok L yang baru langsung diambil dari blok R sebelumnya. Ini adalah satu putaran DES.

Adapun proses kerja pada algoritma DES terdiri dari dua proses, yaitu:

a. *Enkripsi*

Proses enkripsi (*enciphering*) terhadap blok *plainteks* dilakukan setelah permutasi awal (Setiap blok *plainteks* mengalami 16 kali putaran *enciphering*. Setiap putaran *enciphering* merupakan jaringan Feistel yang secara matematis dinyatakan sebagai:

$Li = Ri - 1$ …… (1)

$Ri = Li - 1 \oplus f(Ri - 1, Ki)$ …… (2)

E adalah fungsi ekspansi yang memerluas blok Ri– 1 yang panjangnya 32-bit menjadi blok 48 bit. Hasil ekpansi E(Ri–1) diXOR-kan dengan Ki dan menghasilkan vektor A.

$E(Ri - 1) \oplus Ki = A$…… (3)

Vektor A dikelompokkan menjadi 8 kelompok, setiap kelompok terdiri 6 bit yang menjadi masukan bagi proses substitusi. Proses substitusi dilakukan menggunakan delapan buah kotak-S-Box yaitu S1 sampai S8. Setiap kotak-S-Box menerima masukan 6 bit dan menghasilkan keluaran 4 bit. Kelompok 6-bit pertama disubsitusikan ke dalam kotak S-Box1, kelompok 6-bit kedua disubsitusikan ke dalam kotak S-Box2, dan seterusnya.

b. Dekripsi

Proses dekripsi terhadap *cipher* teks merupakan kebalikan dari proses enkripsi. Kunci pada proses dekripsi yaitu kebalikan dari kunci proses enkripsi, yang mana terdiri dari K[16], K[15] hingga K[1]. Untuk tiap putaran 1, 2, …, 16, keluaran pada setiap putaran *deciphering* [HYPERLINK \l "Roh121" 5], adalah:

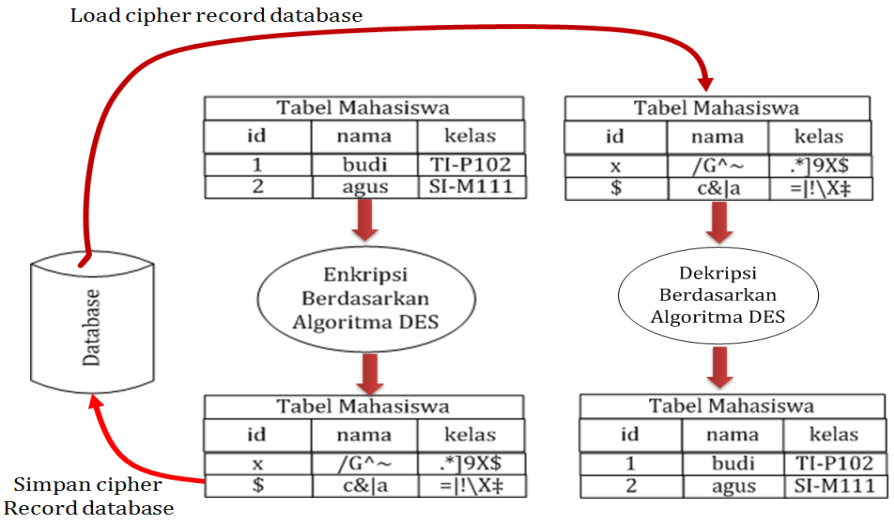
$Li = Ri - 1$ …… (4)

$Ri = Li - 1$ **Error!**$\oplus$**}** $f(Ri - 1, Ki)$…… (5)

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Penyimpanan data menggunakan *database* telah umum digunakan oleh setiap instansi, namun penyimpanan secara langsung tanpa memberi suatu keamanan akan dapat menyebabkan data dapat dibobol dan dicuri atau dirusak oleh pihak-pihak yang tidak bertanggung jawab. Solusi yang perlu diterapkan agar data tersebut tetap aman dari penyimpanannya, maka digunakan salah satu algoritma kriptografi untuk mengekripsi data tersebut. Teks *record* di dalam *database* akan disandikan menjadi simbol-simbol lain yang tidak dapat lagi dapat dipahami oleh pihak lain. Bila *record* yang telah disandikan tersebut dibutuhkan oleh pengguna, maka terlebih dahulu dilakukan proses pengembalian *record* tersandi menjadi karakter *record* asli. Skema proses pengamanan *record database* ditunjukkan pada diagram di bawah ini.



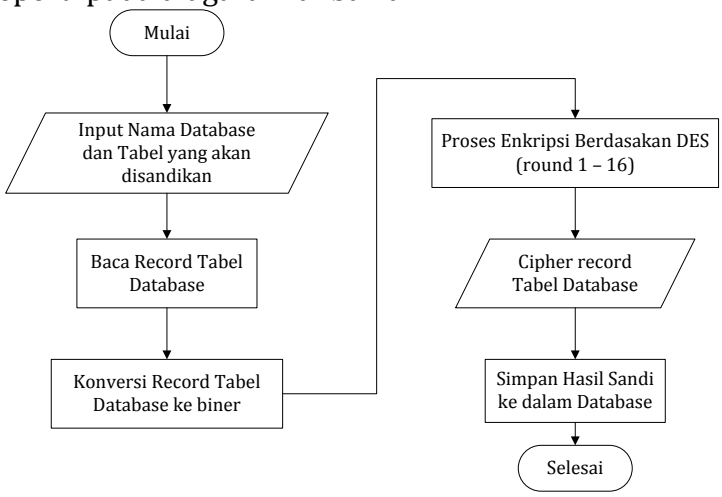
Gambar 1. Skema Penyandian Record Database

Berdasarkan gambar 1 di atas, diketahui bahwa sebelum *record* tabel disimpan ke dalam *database*, terlebih dahulu dilakukan proses penyandian teks *record* tersebut, sehingga teks *record* yang tersimpan ke dalam tabel *database* adalah sandi dari teks *record* asli. Bila makna *record* tersandi tersebut dibutuhkan, maka terlebih dahulu dilakukan proses dekripsi sehingga *record* terebut ditampilkan sesuai dengan karakter aslinya.

Langkah-langkah yang dilakukan pada proses enkripsi *record* mengikuti aturan-aturan yang berlaku pada algoritma DES. Adapun langkah-langkah proses enkripsi *record* yaitu:

a. Konversikan key ke dalam bentuk biner
b. Lakukan proses pembangkitan kunci sesuai aturan dari algoritma DES.
c. Konversikan karakter dari *record* yang dipilih ke dalam biner.
d. Gabungkan setiap bit dari plain, setiap bit terdiri dar 32 bit, sehingga dapat ditentukan R[0] dan L[0].
e. Lakukan proses round 1-16 sesuai aturan dari algoritma DES.

Proses enkripsi *record database* berdasarkan algoritma DES dapat digambarkan seperti pada diagaram di bawah ini.



Gambar 2. Diagaram Penerapan DES dalam Penyandian Record Database

3.2 Implementasi

Record database yang akan dienkripsi yaitu record dari database mahasiswa dengan nama tblmahasiswa, dimana *record* dari tabel dapat dilihat di bawah ini.

Table 1. Tabel Mahasiswa

NPM	Nama	Kelas
14110615	Neti Rusri Yanti	TI-S1412
14110821	Alimah	TI-S1401
14110822	Desi Afrida	TI-M1416

Berdasarkan tabel 1 di atas, sebagai contoh *record database* yang disandikan adalah :

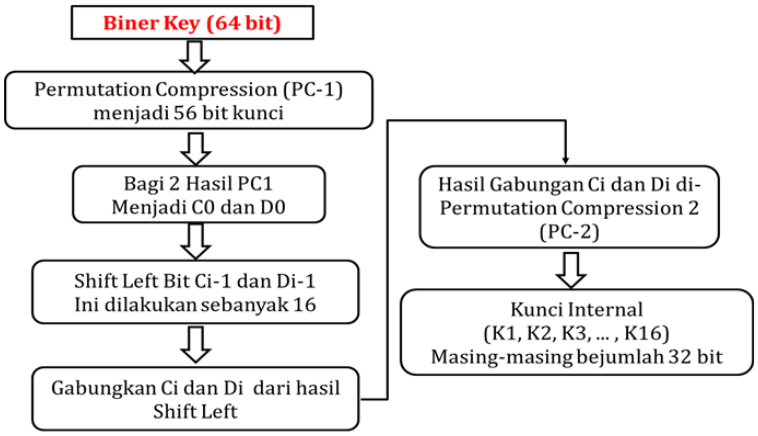
Plaintext = TI-S1412
Key = NETI&DKK

a. *Proses Generate Key* (Pembangkitan Kunci)
Langkah awal adalah mengkonversi kunci ke biner, sehingga dihasilkan biner kunci sebagai berikut :

Tabel 2. Biner Kunci

Char Kunci	Decimal	Biner
N	78	0100 1110
E	69	0100 0101
T	84	0101 0100
I	73	0100 1001
&	38	0010 0110
D	68	0100 0100
K	75	0100 1011
K	75	0100 1011

Adapun diagaram untuk melakukan proses pembangkitan kunci adalah :



Gambar 3. Diagaram Generate Key

Berdasarkan proses pembangkitan kunci, maka diperoleh kunci untuk proses enkripsi sebagai berikut :

K[1] = 100000001001011001000010101000001101101101010010
K[2] = 101000000000001001010010110011110000100110011000
K[3] = 001001000101001000110000100000010101001101011101
K[4] = 100001100001000101010000010100111001001010100100
K[5] = 000011100100001001010001110100000000110110101101
K[6] = 000011110101000100001000000010100011101010011101
K[7] = 000010100000000111001001011100110111000110110001
K[8] = 000110010100100000001001001000110000100100101011
K[9] = 000111010000100010001000001101100100011111010010
K[10] = 000100100010000010001100100111011000000101000011
K[11] = 000110000000110000000100110001101110011001000000
K[12] = 010000000010100000101100011110001010011101001100
K[13] = 100000001010010000100100101110001101010010001010
K[14] = 110000000000111000100010010011000111011000100011
K[15] = 111000001011001000100000101111100110100001101000
K[16] = 101000001001000000100110010000110100110000110111

b. Proses Enkripsi

Kelompokkan biner *plain* menjadi 64 bit setiap kelompok. Karena karakter yang dienkripsi hanya 8 karakter (64 bit) berarti memenuhi satu kelompok.

Biner Plain = 01010100010010010010110101010011
00110001001101000011000100110010

Lakukan *Initial Permutation* (IP) terhadap 64 bit *plain* berdasarkan tabel IP Algoritma DES, maka diperoleh IP dari biner *plain* adalah :

00001011111110010010010101011110
00000000111101000000011010001000

Bagi 2 kelompok hasil IP biner plain, masing-masing 32 bit.

L[0] = 00001011 11111001 00100101 01011110

R[0] = 00000000 11110100 00000110 10001000

Round 1 (i=1)

1. Expansi Nilai R[0]
   R[0] = 00000000111101000000011010001000
   E(R[0] = 0000000000010111101010000000000011010100010100000
2. E(R[0] di-XOR dengan K[1]
3. E(R[0] = 0000000000010111101010000000000011010100010100000
   K[1] = 10000000100101100100001010100000110110110101001⊕
   A[1] = 10000000100000011110101010100000000011110000010
4. A[1] disubsitusikan ke dalam S-Box DES

Tabel 3. Hasil S-Box

Klpk A[1]	Biner Hasil klpk	Decimal dari Biner		Hasil SBOX [Dec]	Biner Hasil SBOX
		b1, b6 [baris]	b2, b3, b4,b5 [klpk]		
1	100000	2	0	4	0100
2	001000	0	4	6	0110
3	000111	1	3	9	1001
4	101010	2	5	11	1011
5	101000	2	4	10	1010
6	000000	0	0	12	1100
7	111100	2	14	9	1001
8	000010	0	2	8	1000

   Gabungkan kembali biner hasil S-Box, sehingga didapatkan :
   B[1] = 01000110100110111010110010010010
5. Permutasikan B[1], berdasarkan tabel P-Box DES
   B[1] = 01000110100110111010110010010010
   Permutasikan berdasarkan tabel P-BOX, sehingga didapatkan :
   P[1] = 11010111010000101000000111011001
6. Mendapatkan Nilai R[1] dan L[1]
   R[1] = P[1] ⊕ L[0]
   P[1] = 11010111010000101000000111011001
   L[0] = 00001011111110010010010101011110 ⊕
   R[1] = 11010100101110011010010010000111
   L[1] = R[0]
   L[1] = 0000 0000 1111 0100 0000 0110 1000 1000

Seterusnya untuk Round 2-16 dilakukan dengan proses yang sama dengan Round 1, hingga dapatlah hasil *cipher* akhir seperti berikut ini.

Tabel 4. Hasil Proses Enkripsi (Round 16)

Biner	Dec	Char Cipher
11100101	229	å
10100000	160	
11000110	198	Æ
11100001	225	á

Biner	Dec	Char Cipher
11000101	197	Å
00111001	57	9
00110000	48	0
00001110	14	

Sehingga bila proses enkripsi ini diterapkan pada masing-masing field tabel, maka record tabel akan terlihat seperti berikut.

Table 5. Tabel Mahasiswa yang Telah Dienkripsi

NPM	Nama	Kelas
Êº,}šy*0	Š‰c^*¢ÃñB?=8&%	å ÆáÅ90
Êº,}?<$!	1êïþh5	‘┬ 7i£[\
Êº,}\]^_	2Ýî‘›†b-.R‰	‘┬W`z∞8¥

c. Proses Dekripsi
Proses dekripsi berdasarkan algoritma DES dilakukan seperti proses enkripsi, hanya saja susunan kunci pada proses dekripsi digunakan secara terbalik. Artinya dimulai dari K[16], K[15], K[14], ..., K[1]. Proses dekripsi diawali dengan memanfaatkan biner-biner cipher yang dibaggi menjadi dua kelompok yaitu L[0] dan R[0]. Dua blok ini kemudian di permutasikan berdasarkan tabel permutasi invers (IP^1). *Output* dari dekripsi adalah blok L[0] dan R[0] sehingga didapatkan biner-biner *plaintext* seperti semula.

4. KESIMPULAN

Berdasarkan pembahasan dalam penelitian ini, maka disimpulkan bahwa:

a. Penyandian *record database* berdasarkan algoritma DES mampu mempersulit pihak-pihak lain untuk memahami dan mengerti isi dari *record database.*
b. Penyandian *record database* diawali dengan penentuan nama *database* dan pemilihan tabel yang memiliki *record* yang akan disandikan, dimana maasing-masing *record* pada tabel yang dipilih disandikan berdasarkan algoritma DES.
c. Tingkat keamanan dari yang dibuat cukup aman karena algoritma DES memiliki panjang kunci yang besar. Kunci internal yang berjumlah 56 bit didapatkan dari kunci eksternal yang berjumlah 64 bit.

DAFTAR PUSTAKA

[1] S.H. Suryawan, Hamdani, "Pengamanan Data File Dengan Menggunakan Algoritma Enkripsi Rivest Code 5 ," *Jurnal Informatika Mulawarman* , vol. Vol. 8, No. 2 , pp. 44-49, 2013.
[2] E. Setyaningsih, *Kriptografi & Implementasinya Menggunakan Matlab*, Yogyakarta:Andi, 2015.
[3] T. Zebua and E. Ndruru, “PENGAMANAN CITRA DIGITAL BERDASARKAN MODIFIKASI ALGORITMA RC4,” *J. Teknol. Infomasi dan Ilmu Komput.*, vol. 4, no. 4, pp. 275–282, 2017.

[4] E. Setyaningsih, "Penyandian Citra Menggunakan Metode Playfair Cipher", J. Teknol., vol. 2, no. 2, pp. 213–219, 2009.
[5] Hamdani, S.H. Suryawan, A. Septiarini,"Pengujian Algoritma Rivest Code 5 Untuk Enkripsi Struktur File Dokumen," *Jurnal Informatika Mulawarman*, vol. Vol. 8 No. 2, pp. 44-49, Juni 2013.
[6] E. Aribowo, "Aplikasi Pengamanan Dokumen Office Dengan Algoritma Kriptografi Kunci Asimetris Elgamal," *J. Inform.*, vol. 2, no. 2, pp. 209–219, 2008.
[7] Rifki Sadikin, *Kriptografi Untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java*, Yogyakarta: Andi, 2012.
[8] H. Pandiangan and S. Sijabat "PERANCANGAN MEDIA PENGIRIMAN PESAN TEKS DENGAN PENYANDIAN PESAN MENGGUNAKAN ALGORITMA RC4 BERBASIS WEB ," *Jurnal Matik Penusa* , vol. Volume XIX, No. 1 , no. ISSN 2088-3943 , pp. 63-71, Juni 2016
[9] T. Zebua, "ANALISA DAN IMPLEMENTASI ALGORITMA TRIANGLE CHAIN PADA PENYANDIAN RECORD DATABASE," *Pelita Inform. Budi Darma*, vol. 3, no. 2, pp. 37–49, 2013.
[10] N. Widyastuti, "Pengembangan Metode Beaufort Cipher Menggunakan Pembangkit Kunci Chaos," *J. Teknol.*, vol. 7, no. 1, pp. 73–82, 2014.
[11] U. R. S. Lubis, Mesran, and T. Zebua, "Implementasi Algoritma Chua Chaotic Noise Pada Enkripsi Citra RGB," in *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 2017, vol. I, no. 1, pp. 220–224.