PENGAMANAN DATA NILAI PADA APLIKASI E-RAPORT BERDASARKAN ALGORITMA 2DES

Indra Williamsyah Sinaga¹, Imam Saputra¹, Taronisokhi Zebua²

¹Program Studi Teknik Informatika STMIK Budi Darma, Medan, Indonesia

²AMIK STIEKOM Sumatera Utara, Rantauprapat, Indonesia

Email: ¹indrawilliamsyahsinaga@gmail.com, ¹saputraimam69@gmail.com

Abstrak

Aplikasi E-Raport salah satu upaya pengendalian mutu penilaian, mengingat keterbatasannya sumber daya dan kualifikasi yang ada. Data nilai pada E-Raport merupakan hal yang sangat dijaga kerahasiannya agar terhindar dari pihak yang tidak bertanggung jawab. Salah satu solusi dari permasalahan di atas adalah menyandikan record database e-raport agar data nilai dapat terjaga dan terlindungi kerahasiannya. Salah satu algoritma yang dapat diterapkan adalah Double Data Encryption standard (2DES). Dengan menerapkan algoritma 2DES data nilai pada aplikasi E-Raport terlindungi dari pihak lain yang tidak bertanggung jawab serta sistem penilaian dapat berjalan secara baik dan data nilai hanya dapat diakses oleh pengguna yang memiliki hak akses untuk mengolah data nilai tersebut.

Kata Kunci: Data Nilai, E-Raport, Kriptografi, algoritma 2DES

Abstract

E-Raport application is an effort to control the quality of assessment, given the limitations of existing resources and qualifications. Value data in E-Raport is a matter of confidentiality that is very guarded to avoid irresponsible parties. One solution to the problem above is to encode the e-raport database record so that the value data can be maintained and confidentiality protected. One of the algorithms that can be applied is the Double Data Encryption standard (2DES). By applying the 2DES algorithm the value data in the E-Raport application is protected from other irresponsible parties and the assessment system can run well and the value data can only be accessed by users who are has access rights to process the data value.

Keywords: Value Data, E-Raport, Cryptography, 2DES algorithm

1. PENDAHULUAN

Penilaian pendidikan merupakan proses pengumpulan dan pengolahan informasi untuk mengukur pencapaian hasil belajar peserta didik mencakup penilaian kinerja, penilaian diri, penilaian berbasis portofolio, penilaian harian, penilaian tengah semester, penilaian akhir semester, ujian tingkat kompetensi, ujian mutu tingkat kompetensi, ujian nasional, ujian sekolah berstandar nasional, dan ujian sekolah/madrasah. Peningkatan kualitas layanan pendidikan merupakan salah satu prioritas pembangunan pendidikan, di samping tersedianya kurikulum yang handal, salah satu aspek terpenting dalam upaya menjamin kualitas layanan pendidikan adalah menyediakan system penilaian yang komprehensif. Pengamanan suatu data sangat perlu dilakukan agar kerahasiaan data dapat terjaga, terutama bila informasi atau isi dari data hanya pihak tertentu saja yang berhak mengetahui. Penyadapan merupakan salah satu resiko pengamanan yang sering terjadi bila kita melakukan pengiriman data atau informasi [1]. Aplikasi *E-Raport* adalah salah satu upaya pengendalian mutu penilaian, karena Aplikasi *E-Raport* merupakan suatu penyelesaian dalam mengingat keterbatasan sumber daya dan kualifikasi personal yang ada, dengan tujuan membantu pendidik dalam mengolah data nilai pengetahuan, keterampilan, sikap spiritual, dan membantu menyusun laporan hasil penilaian. Tujuan akhirnya tidak lain adalah untuk mempermudah dalam menyusun laporan hasil penilaian peserta didik berupa laporan per penilaian, laporan pencapaian kompetensi, rapor, dan legger. Di dunia pendidikan data nilai sangat penting dijaga kerahasiaannya, karena akan sangat berbahaya bila data nilai yang sedang didistribusikan terkena penyadapan [2]. Salah satu cara untuk mencegah masalah penyadapan dalam ilmu komputer adalah penerapan teknik kriptografi (peyandian data).

Kriptografi merupakan teknik matematika yang berupa kerahasiaan, integritas data, dan otentikasi dalam keamanan informasi. Kriptografi tidak hanya memberi keamanan informasi, melainkan lebih kearah teknik-teknik. Proses utama dalam suatu algoritma kriptografi adalah enkripsi dan dekripsi. Enkripsi merubah sebuah *Plaintext* kedalam bentuk *Ciphertext*, sedangkan Dekripsi kebalikan dari enkripsi, merubah *Ciphertext* kembali kedalam bentuk *Plaintext* [3].

Double Data Encryption Standard (2DES) adalah variasi yang lebih kuat dari algoritma enkripsi DES. Algoritma *Double DES* termasuk sistem kriptografi simetris dan tergolong jenis blok kode. 2DES mengenkripsikan 64 bit teks-asli menjadi 64 bit teks-kode dengan menggunakan 56 bit kunci internal (*internal key*) atau sub-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit. Sehingga dengan demikian membuat para kriptanalisis kesulitan untuk menyadap suatu data atau pesan [3].

2. TEORITIS

2.1 Keamanan Data

Keamanan data adalah kebebasan dari bahaya atau sebagai kondisi keselamatan. Keamanan adalah perlindungan data di dalam suatu sistem melawan terhadap otorisasi tidak sah, modifikasi, atau perusakan dan perlindungan sistem komputer terhadap penggunaan yang tidak bertanggung jawab [1]. Keamanan data dapat dibedakan menjadi dua kategori, yaitu keamanan fisik dan keamanan sistem. Keamanan fisik merupakan bentuk keamanan berupa fisik dari server, terminal/client router sampai dengan cabling. Sedangkan keamanan sistem adalah keamanan pada sistem pengoperasiannya atau lebih khususnya pada lingkup perangkat lunak, misalnya dengan penggunaan kriptografi dan steganografi.

2.2 Kriptografi

Kriptografi (*Cryptography*) berasal dari bahasa Yunani, yaitu dari kata *crypto* dan *graphia* yang berarti 'penulisan rahasia'. Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah [3]. Kriptografi merupakan suatu seni keamanan yang memiliki proses mengacak atau mengubah pesan asli menjadi pesan acak yang susah untuk dipahami, sehingga penerima yang bukan berhak akan susah untuk mengerti isi dari pesan tersebut. Tahapan yang dilakukan pada kriptografi adalah pengubahan atau pengacakan *plaintext* (file asli) yang berbentuk digital menjadi data yang sulit untuk dipahami *ciphertext*, tahapan tersebut adalah *encryption* dan tahapan sebaliknya mengubah data *ciphertext* menjadi *plaintext* disebut *decryption*.

2.3 Algoritma Double Data Encryption Standard

Double Data Encryption Standard (2DES) merupakan suatu algoritma pengembangan dari algoritma DES (*Data Encryption Standard*). Pada dasarnya algoritma yang digunakan sama hanya pada 2DES dikembangkan dengan melakukan enkripsi dan implementasi algoritma DES sebanyak dua kali 2DES memiliki dua buah kunci yang berukuran 96-bit (dua kali kunci 48-bit dari DES). Pada algoritma 2DES dibagi menjadi dua tahap, setiap tahapnya merupakan implementasi dari algoritma DES. 2DES mempunyai kelemahan yaitu dapat diserang dengan algoritma yang dikenal sebagai *meet-in-the-middle-attack*, yang pertama kali ditemukan oleh Diffie dan Helman. *Double DES* mengeksekusi algoritma DES sebanyak dua kali.

Double DES menggunakan dua buah kunci eksternal, yaitu K1 dan K2. Fungsi enkripsi dan dekripsinya adalah [3]:

Gambar 1. Diagram enkripsi dan dekripsi 2DES

3. ANALISA DAN PEMBAHASAN

3.1 Analisa

Berdasarkan analisa yang dilakukan terhadap data yang tidak diamankan menggunakan teknik enkripsi dan dekripsi, maka data hanya disimpan dalam bentuk data yang tidak aman, sehingga siapapun masih dapat membaca dan mengerti isinya. Hal ini sangat rentan dengan terjadinya kecurangan yang dapat dilakukan oleh pihak-pihak yang tidak bertanggung jawab. Dalam data nilai ini yang diamankan adalah *record database* karena didalamnya berisi informasi-informasi yang bersifat rahasia. Sehingga pentingnya *record database e-raport* diberikan keamanan yang lebih agar informasi dari *e-raport* dapat terjaga dari pihak yang ingin melakukan kecurangan. Proses awal analisa enkripsi yang dilakukan memasukkan data nilai yang akan disimpan dan proses enkripsi dilakukan pada saat disimpan, kemudian data masuk ke *database* dan untuk melakukan proses dekripsi dilakukan penekanan pada tombol cetak, di dalam tombol cetaklah dilakukan proses dekripsi sehingga menghasilkan laporan berbentuk *e-raport*.

Gambar 2. Prosedur Enkripsi dan Dekripsi Data Nilai Aplikasi E-Raport

3.2 Penerapan Algoritma 2DES

Penyimpanan data menggunakan *database* sangat umum digunakan oleh setiap instansi, namun dalam penyimpanan secara langsung tanpa memberi keamanan dapat menyebabkan kerusakan yang dilakukan oleh pihak-pihak yang tidak bertanggung jawab. Solusi yang perlu diterapkan dengan menggunakan salah satu algoritma kriptografi untuk mengenkripsi data tersebut. Teks *record* dalam *database* akan disandikan menjadi simbol-simbol lain yang tidak dapat dipahami oleh pihak lain.

Database yang digunakan untuk menerapkan metode 2DES berisi informasi siswa(e-raport siswa) dengan NIS sebagai *primary key*. Kunci dari algoritma 2DES akan ditanamkan langsung ke dalam sistem untuk dijadikan *default key* sehingga proses enkripsi dan dekripsi dilakukan berdasarkan kunci yang sudah ditanamkan tersebut.

Salah satu contoh isi *record* tabel dari *database* yang diamankan sebagai berikut:

NIS	Nama	Kelas
123456789	Indra	X_UNGUL
987654321	Sinaga	X_UNGUL
787878455	Williamsyah	X_UNGUL
232345456	Ranga	X_UNGUL
121212234	Tri Anggi	X_UNGUL

Gambar 3. *record database*

Dalam proses Enkripsi dan Dekripsi dilakukan proses per *field* dari setiap tabel dan proses per *record* dari setiap *field*

Langkah pertama merubah *Plaintext* Dan *Key* Menjadi Bentuk Biner

Plaintext : P = X_UNGUL

Kunci : K1 = SILALAH

K2 = SIPAYUNG

	KEY 1		KEY 2		PLAINTEXT	
S	01010011	S	01010011	X	01011000	
I	01001001	I	01001001	_	01011111	
L	01001100	P	01010000	U	01010101	
A	01000001	A	01000001	N	01001110	
L	01001100	Y	01011001	G	01000111	
A	01000001	U	01010101	G	01000111	
H	01001000	N	01001110	U	01010101	
I	01001001	G	01000111	L	01001100	

1. Proses Pembangkitan Kunci

Pada proses pembangkitan kunci dilakukan dari kunci 1 sampai kunci 2, Melakukan *Permutation Compression 1* (PC-1) pada tiap-tiap kunci berdasarkan tabel PC-1 yaitu dengan cara memindahkan biner sesuai dengan urutan pada tabel PC-1 (posisi 1 pada kunci diubah ke kolom 2 pada tabel PC-1), dan kemudian dibagi menjadi 2 kelompok yaitu *Ci* dan *Di*.

Tabel 1. *Permutation Compression 1* (PC-1)

57	49	41	33	25	17	9	Ci
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
63	55	47	39	31	23	15	Di
7	62	54	45	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

Cara memindahkan biner-biner menjadi hasil PC-1 sebagai berikut:

Index	1	2	3	4	5	6	7	8
Bit	0	1	0	1	0	0	1	1

Sehingga akan menghasilkan kunci sebagai berikut:

Karakter Kunci 1 : S I L A L A H I
 Biner Key 1 : 01010011 01001001 01001100 01000001 01001100 01000001 01001000 01001001
 Hasil PC-1 Key 1 : 0000000 0100000 1100000 0000000 1100000 0000000 0011010 0111111
 Karakter Kunci 2 : S I P A Y U N G
 Biner Key 2 : 01010011 01001001 01010000 01000001 01011001 01010101 01001110 01001001
 Hasil PC-1 Key 2 : 0000000 0100000 1100000 0000000 1100000 0000000 0011010 0111111
 Kemudian melakukan *Shift Left* DES terhadap C0 dan D0 sebanyak 16 kali berdasarkan aturan jumlah perpindahan bit disetiap putaran.

Tabel 2. *Shift Left* DES

Putaran	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
Jumlah Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Sehingga menghasilkan perpindahan kunci sebagai berikut:

Hasil perpindahan biner key 1 (Hasil PC-1)

C0 = 00000000111111110000000000011	D0 = 1100000111100000010100100101
C1 = 000000011111111100000000000110	D1 = 1000001111000000101001001011
C2 = 0000001111111111000000000001100	D2 = 0000011110000001010010010111
C3 = 000011111111110000000000110000	D3 = 0001111000000101001001011100
C4 = 001111111110000000000110000000	D4 = 0111100000010100100101110000
C5 = 1111111100000000001100000000	D5 = 1110000001010010010111000001
C6 = 111111000000000110000000011	D6 = 1000000101001001011100000111
C7 = 1111000000000011000000001111	D7 = 0000010100100101110000011110
C8 = 110000000001100000000111111	D8 = 0001010010010111000001111000
C9 = 100000000011000000001111111	D9 = 0010100100101110000011110000
C10 = 0000000001100000000111111110	D10 = 1010010010111000001111000000
C11 = 0000000110000000011111111000	D11 = 1001001011100000111100000010
C12 = 0000011000000001111111100000	D12 = 0100101110000011110000001010
C13 = 0001100000000111111110000000	D13 = 0010111000001111000000101001
C14 = 0110000000011111111000000000	D14 = 1011100000111100000010100100
C15 = 1000000001111111100000000001	D15 = 1110000011110000001010010010
C16 = 0000000011111111000000000011	D16 = 1100000111100000010100100101

Selanjutnya gabungkan kembali setiap Ci dan Di hasil dari *shift left* dan hasil PC-1.

INDEX	1	2	3	4	5	6	7	8	9	10	...	64
Biner C0D0	0	0	0	0	0	0	0	0	1	1	...	1

Gambar 3. Hasil Gabungan CiDi

Kemudian di proses kembali berdasarkan tabel PC-2 dengan mengubah posisi bit dari CiDi sesuai dengan PC-2. Misalnya Posisi biner 1 dari C0D0 diubah ke kolom 5 berdasarkan tabel (PC-2) dan menghasilkan Ki.

Tabel 3.3 *Permutation Compression 2* (PC-2)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Cara memindahkan biner-biner PC-2 agar menghasilkan Ki sebagai berikut:

INDEX	1	2	3	4	5	6
Bit	0	0	0	0	0	0

Gambar 4. Proses pemindahan Biner menjadi Ki

Proses pemindahan dilakukan secara terus menerus sehingga menghasilkan Ki.

Kunci	Biner
K1-0	10100000100100100010000000011101000100111100
K1-1	101000001001001001000010100011000100111000001000

K1-2	101000000001001001010010000000010001110010100000
K1-3	00100100010100100101000011001000000100000110101
K1-4	000001100101000101010000000000110100101010011100
K1-5	000011100100000101010001000100010001000110010001
K1-6	000011110100000100001001100000110000000000100101
K1-7	000010110000000110001001010000100010101110000100
K1-8	000110010000100010001001001100000000000110011101
K1-9	000110010000100010001000101001100010010001000100
K1-10	000100000010100010001100001010001010001111000010
K1-11	000100000010110000000100001101001100010000000011
K1-12	010000000010110000100100010011100000010001000010
K1-13	110000001010010000100100100011001110000101001000
K1-14	110000001000011000100010001000001101011001000000
K1-15	111000001001001000100010110110001000010000100010
K1-16	101000001001001000100010000000011101000100111100

2. Proses Enkripsi

Proses enkripsi memiliki tahapan-tahapan untuk mencari sebuah ciphertext, proses ini diawali dengan menginput plaintext, kemudian mengkonversi plaintext ke biner dan menjadikan pengelompokkan sebarang 2 bagian L_i dan R_i , melakukan *Initial Permutation* (IP) pada biner plaintext hingga menghasilkan biner baru, kemudian melakukan iterasi hingga 16 round, R_i akan melakukan ekspansi dengan menggunakan tabel *Expansi*, Kemudian hasil tersebut akan menghasilkan $E(R_i)$ yang akan di Xor dengan $K1$ dari kunci 1 sehingga menghasilkan $A1$. Kemudian biner dari $A1$ dibagikan menjadi 8 kelompok yang masing-masing terdiri dari 6 biner. Setiap kelompok di substitusi kedalam S-BOX hingga mendapatkan $B1$. Kemudian $B1$ melakukan permutasi dengan tabel P-BOX hingga menghasilkan P-BOX $B1$, Kemudian melakukan Xor dengan $L0$ dan akan menghasilkan $R1$ dan $L1$ hingga menghasilkan $L16$ dan $R16$. Setelah mendapatkan hasil $L16$ dan $R16$ biner dipermutasi dengan *Initial permutation Invers* (IP^{-1}), kemudian hasil biner dikembalikan dalam bentuk karakter, Hingga menghasilkan *Ciphertext*.

a) Melakukan *Initial Permutation* (IP) Pada Biner *Plaintext*.

Biner dari *plaintext* diacak berdasarkan tabel IP dan akan menghasilkan 2 kelompok yaitu L_i dan R_i .

Tabel 3. *Initial Permutation* (IP)

58	50	42	34	26	18	10	2	L0
60	52	44	36	28	20	12	4	
62	54	46	38	30	22	14	6	
64	56	48	40	32	24	16	8	
57	49	41	33	25	17	9	1	R0
59	51	43	35	27	19	11	3	
61	53	45	37	29	21	13	5	
63	55	47	39	31	23	15	7	

X	0	1	0	1	1	0	0	0
–	0	1	0	1	1	1	1	1
U	0	1	0	1	0	1	0	1
N	0	1	0	0	1	1	1	0
G	0	1	0	0	0	1	1	1
G	0	1	0	0	0	1	1	1
U	0	1	0	1	0	1	0	1
L	0	1	0	0	1	1	0	0

Sehingga menghasilkan biner baru sebagai berikut.

$L0 = 1111111101000111111111001110110$

$R0 = 00000000000000001000101100111010$

b) Melakukan Iterasi Hingga 16 Rounds

Kemudian $R0$ akan melakukan ekspansi dengan menggunakan tabel *Expansi*.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Proses *Expansi* dilakukan dengan cara bit ke 32 pindah ke kolom 1 baris 1. Adapun posesnya sebagai berikut:

[illegible]

R0 = 000000000000000000001010001

p	1	1	0	0
q	1	0	1	0
Pq	0	1	1	0

1	0	1	0	0	0
---	---	---	---	---	---

Sehingga akan memberikan hasil sebagai berikut.

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	4	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7

3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4 :																
Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5:																
Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6 :																
Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	2	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7 :																
Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8:																
Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	5	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	3	15	12	9	0	3	5	6	11

B1 1101 1111 0011 1000 1001 1101 0110 0101

Setelah mendapatkan hasil tersebut, maka dilanjutkan dengan melakukan *Permutation Box* (P-BOX) dengan B1.

Tabel 7. *Permutation Box* (P-Box)

Tabel P-Box							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Sehingga menghasilkan P-BOX(B1) sebagai berikut.

P-BOX(B1) = 11110101100110001110110000111111

Setelah mendapatkan P-Box(B1) kemudian akan di XOR dengan L0, dan akan menghasilkan R1 dan L1 didapat dari R0.

P-Box(B1) 11110101100110001110110000111111

L0 111111110100011111111111001110110 XOR

R1 00001010110111110001001001001001

Tahapan ini akan terus berlanjut hingga mendapatkan R16 dan L16.

R0	0000000000000001000101100111010	L0	111111110100011111111111001110110
R1	00001010110111110001001001001001	L1	0000000000000001000101100111010
R2	11110101100110000110011100000101	L2	00001010110111110001001001001001
R3	11111111010001111111111001110110	L3	11110101100110000110011100000101
R4	0000000000000001000101100111010	L4	11111111010001111111111001110110
R5	00001010110111110001001001001001	L5	0000000000000001000101100111010
R6	11110101100110000110011100000101	L6	00001010110111110001001001001001

R7	111111101000111111111001110110	L7	11110101100110000110011100000101
R8	0000000000000001000101100111010	L8	11111111010001111111111001110110
R9	00001010110111110001001001001001	L9	00000000000000001000101100111010
R10	11110101100110000110011100000101	L10	00001010110111110001001001001001
R11	1111111010001111111111001110110	L11	11110101100110000110011100000101
R12	00000000000000001000101100111010	L12	11111111010001111111111001110110
R13	00001010110111110001001001001001	L13	00000000000000001000101100111010
R14	11110101100110000110011100000101	L14	00001010110111110001001001001001
R15	1111111010001111111111001110110	L15	11110101100110000110011100000101
R16	00000000000000001000101100111010	L16	11111111010001111111111001110110

Setelah mendapatkan R16 dan L16 maka selanjutnya adalah pengembalian biner dengan melakukan *Initial Permutation Invers* (IP^{-1}), kemudian pengembalian biner menjadi karakter.

Tabel 8. *Initial Permutation Invers*

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Tabel *Initial permutation invers* di permutasi dengan hasil R16 dan L16

R16 = 00000000000000001000101100110101

L16 = 1111111101000111111111001110110

Setelah proses selesai maka mendapatkan hasil (IP^{-1}) dan kemudian biner hasil tersebut menjadi karakter

Tabel 9. Hasil IP^{-1}

1	0	1	0	0	1	0	1
1	0	1	0	1	1	1	0
1	0	1	0	1	0	1	1
1	0	0	0	1	1	0	0
1	0	0	0	1	0	1	1
1	0	0	0	1	0	1	1
1	0	1	0	1	0	1	0
1	0	0	0	1	1	0	0

Tabel 10. Hasil biner diubah menjadi karakter

P	10100101	10101110	10101011	10001100	10001011	10001011	10101010	10001100
Dec	165	174	171	140	139	139	170	140
Char	¥	®	«	8C	8B	8B	^a	8C

Sehingga menghasilkan pra-ciphertext 1 yaitu ¥ ® « 8C 8B 8B ^a 8C. Pra-ciphertext 1 akan menjadi *plaintext* untuk proses enkripsi kedua hingga menghasilkan *ciphertext*. Hasil dari tahap-tahap tersebut dapat dilihat pada table berikut.

Tabel 11. Hasil Enkripsi

Proses Enkripsi ke-i	Hasil Enkripsi Char
Pra-ciphertext 1	¥ ® « 8C 8B 8B ^a 8C
Ciphertext	] \ Z] W L G G U L

Setelah mengalami proses enkripsi 2DES maka akan berubah menjadi 5A5D574C4747554C.

NIS	Nama	Kelas
123456789	Indra	Z]WLGGUL
987654321	Sinaga	Z]WLGGUL
787878455	Williamsyah	Z]WLGGUL
232345456	Rangga	Z]WLGGUL
121212234	Tri Anggi	Z]WLGGUL

Gambar 7. Tabel *record database* setelah di enkripsi

3. Proses Dekripsi

Pada Proses dekripsi proses pembangkitan kunci dilakukan seperti proses pembangkitan kunci proses Enkripsi. Proses Dekripsi memiliki tahapan-tahapan untuk mencari sebuah *Plaintext*, proses ini diawali dengan menginput *plaintext*, kemudian mengkonversi plaintext ke biner dan menjadikan pengelompokkan sebarang 2 bagian L_i dan R_i , melakukan *Initial Permutation* (IP) pada biner plaintext hingga menghasilkan biner baru, kemudian melakukan iterasi hingga 16 round, R_i akan melakukan ekspansi dengan menggunakan tabel *Expansi*, Kemudian hasil tersebut akan menghasilkan $E(R_i)$ yang akan di Xor dengan K_1 dari kunci 2 sehingga menghasilkan A_1 . Kemudian biner dari A_1 dibagikan menjadi 8 kelompok yang masing-masing terdiri dari 6 biner. Setiap kelompok di substitusi kedalam S-BOX hingga mendapatkan B_1 . Kemudian B_1 melakukan permutasi dengan tabel P-BOX hingga menghasilkan P-BOX B_1 , Kemudian melakukan Xor dengan L_0 dan akan menghasilkan R_1 dan L_1 hingga menghasilkan L_{16} dan R_{16} . Setelah mendapatkan hasil L_{16} dan R_{16} biner dipermutasi dengan *Initial permutation Invers* (IP^{-1}), kemudian hasil biner dikembalikan dalam bentuk karakter, Hingga menghasilkan *Plaintext*.

a) Merubah *Plaintext* Dan *Key* Menjadi Bentuk Biner

b) Melakukan Permutasi Biner Plaintext

Biner dari *plaintext* diacak berdasarkan tabel IP dan akan menghasilkan 2 kelompok yaitu L_i dan R_i .

c) Melakukan Iterasi Hingga 16 Rounds

Tabel 12 Hasil Biner di ubah menjadi karakter

P	10100101	10101110	10101011	10001100	10001011	10001011	10101010	10001100
Dec	165	174	171	140	139	139	170	140
Char	¥	®	«	8C	8B	8B	^a	8C

Sehingga menghasilkan *ciphertext* yaitu ¥ ® « 8C 8B 8B ^a 8C. Pra-*plaintext* 1 akan menjadi *ciphertext* pada proses dekripsi kedua hingga menghasilkan *plaintext*. Hasil dari tahap-tahap tersebut dapat dilihat pada table berikut.

Tabel 13. Hasil Dekripsi

Proses Dekripsi ke-i	Hasil Char
Ciphertext	Z]WLGGUL
Pra-plaintext 1	¥ ® « 8C 8B 8B ^a 8C
Plaintext	X_UNGGUL

Sehingga menghasilkan *Plaintext* yaitu X_UNGGUL, berdasarkan algoritma 2DES.

NIS	Nama	Kelas
123456789	Indra	X_UNGGUL
987654321	Sinaga	X_UNGGUL
787878455	Williamsyah	X_UNGGUL
232345456	Rangga	X_UNGGUL
121212234	Tri Anggi	X_UNGGUL

Gambar 8. *Reocrd database* setelah di dekripsi

4. KESIMPULAN

Setelah mengimplementasikan perangkat lunak pengamanan data nilai pada aplikasi *e-raport* berdasarkan algoritma 2DES, maka dapat diambil beberapa kesimpulan dari bab-bab sebelumnya. Adapun beberapa kesimpulan sebagai berikut:

1. Pengamanan data nilai pada aplikasi *e-raport* dilakukan dengan mengenkripsi data nilai yang akan di input, sehingga data yang tersimpan di dalam *database* merupakan *cipher* dari data nilai. Proses dekripsi dilakukan pada saat pemanggilan *record* pada laporan data nilai.
2. Algoritma 2DES diterapkan pada proses *input* sehingga akan menghasilkan data nilai yang berisi *ciphertext* pada *database*.

5. REFERENCE

- [1] Dony Ariyus, Pengantar Ilmu Komputer. Yogyakarta: Graha Ilmu, 2008.
- [2] J. T. Elektro, U. K. Indonesia, A. Euclid, and C. B. Chaining, "Vol.8, No. 1," vol. 8, no. 1, pp. 77–90.
- [3] R. Munir, Kriptografi. Bandung: Informatika, 2006.