Jurnal CyberTech

Vol.x. No.x, Juli 2019, pp. xx~xx
P-ISSN : 9800-3456
E-ISSN : 2675-9802

Implemantasi Kriptografi Pengamanan Data Tamu Pada Hotel Sibayak Multi Menggunakan Metode Data Encryption Standard (DES)

Kristina Meldawati Mahulae [#1], Faisal Taufik[#2], Usti Fatimah Sari Sitorus[.#3]

[#1] Sistem Informasi, STMIK Triguna Dharma
[#2.3] Program Studi Dosen Pembimbing, STMIK Triguna Dharma

Article Info

Article history:
Received Jun 12th, 201x
Revised Aug 20th, 201x
Accepted Aug 26th, 201x

Keyword:
Implementasi Kriptoggrafi
Data Encryption Standard
Pengaman Data Tamu

ABSTRACT

Dalam sebuah hotel memiliki data atau informasi yang sangat penting dan perlu di lakukan untuk menjaga kerahasiaan dan keakuratan data tersebut. Oleh sebab itu berbagai hotel termaksud sibayak multi harus melakukan pengamanan data tamu agar data tersebut aman dan terjaga. Karena Kerahasiaan sebuah data merupakan aspek yang sangat penting bagi hotel untuk melindungi data atau privasi tamu hotel sibayak multi.

Data Encryption Standard (DES) adalah salah satu metode kriptografi cipher blok yang popular digunakan karena tingginya tingkat keamananinformasi dan dijadikan standard algoritma enkripsi kunci-simetri. DES adalah nama standard enkripsi simetri yang dahulu memiliki nama algoritma enkripsinya DEA (Data Encryption Algorithm), namun nama DES lebih populer dari pada DEA.

Perangkat lunak yang dibangun akan menggunakan metode Data Encryption Standard, pada konsep perancangan pengamanan pada hotel sibayak multi data yang digunakan adalah data tamu. Hasil dari pengamanan menggunakan metode Data Encryption Standard adalah berbasis desktop dan dapat membantu pihak hotel dalam mengamankan data tamu.

***Kata Kunci** : Kriptogrfi, Data Encryption Standard*

Corresponding Author: *First Author
Nama : Kristina Meldawati Mahulae
Program Studi Sistem Informasi
STMIK Triguna Dharma
Email: Kristina Mahulae18@gmail.com

1. PENDAHULUAN

Hotel Sibayak Multi didirikan pada tahun 2002 yang berlokasi diberastagi, Hotel ini memberikan fasilitas dan pelayanan dikarenakan adanya kegiatan pariwisata disekitar lokasi tersebut, Hotel Sibayak Multi sebagai penyedia tempat tinggal sementara bagi para wisatawan dan menyediakan fasilitas positif bagi pengunjung, Hotel Sibayak Multi juga memiliki pengaruh yang cukup besar dalam meningkatkan perkembangan perekonomian daerah, namun persaingan antar Hotel yang ketat membutuhkan fasilitas dan keamanan bagi pengunjung diera digitalisasi saat ini.

Dengan kemajuan digitalisasi yang ada pada saat ini berbagai perusahaan-perusahaan termaksud Hotel Sibayak Multi sudah menggunakan media komputer dalam menunjang pekerjaan dihotel itu sendiri yang bersifat pengolahan data, dan oleh sebab itu Hotel Sibayak Multi itu juga harus memperhatikan aspek keamanan data yang ada didalam hotel tersebut, dikarenakan data-data yang ada didalam hotel sibayak multi itu sendiri ada data yang bersifat pribadi dan rahasia sehingga jika data tersebut jatuh kepihak yang tidak semestinya akan berdampak buruk terhadap reputasi hotel itu sendiri, dan bahkan juga dapat berdampak buruk bagi tamu-tamu hotel tersebut, misalnya bocornya data diri tamu yang sifatnya rahasia dan apabila data tersebut disalah gunakan orang yang tidak bertanggung jawab akibatnya sangat fatal. seperti penipuan yang mengatas namakan hotel, sehingga Hotel akan kehilangan reputasi, untuk mengatasi permasalahan tersebut, Maka di butuhkan sebuah metode penyandian, ilmu sekaligus seni guna menjaga file yang disebut juga dengan kriptografi [1]**.**

2. METODE PENELITIAN

Metodologi penelitian merupakan pengumpulan inMenuasi terhadap objek yang akan diteliti serta melakukan investigasi pada data yang di dapatkan tersebut. Dalam melakukan penelitian harus terjun langsung kelapangan untuk mendapatkan data sesuai dengan yang ingin diteliti. Pada proses penelitian terdapat tiga teknik yang digunakan yaitu wawancara, observasi dan studi kepustakaan [2].

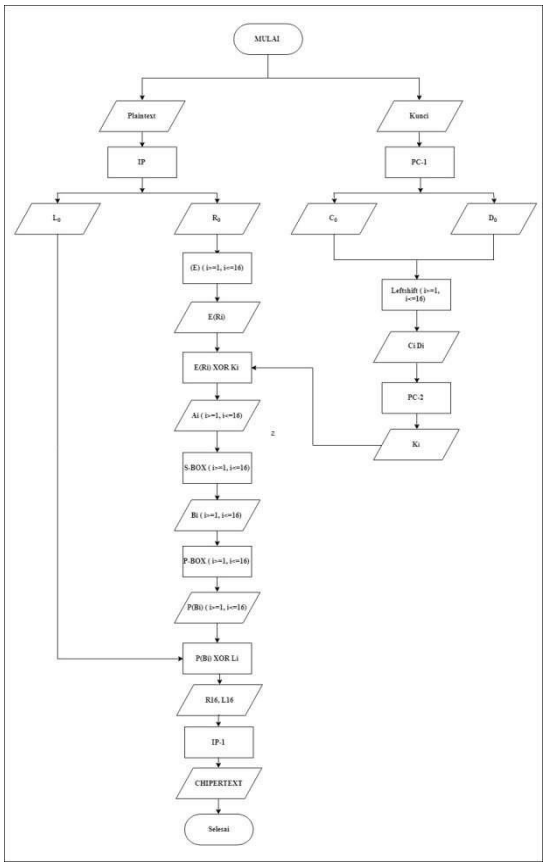
1. Observasi
   Observasi merupakan salah satu teknik dalam pengumpulan data yang kompleks. Dalam penelitian ini observasi dilakukan untuk mendapatkan data di Hotel Sibayak Multi. Hal ini bertujuan untuk memperoleh inMenuasi tentang data yang akan digunakan dalam penelitian ini. Data yang didapatkan di Hotel Sibayak Multi adalah data tamu.
2. Wawancara
   Teknik wawancara ini dilakukan untuk mendapatkan informasi tambahan dari pihak-pihak yang memiliki wewenang dan berinteraksi langsung dengan *front office manager* (Ivan Surbakti).

2.2 Algoritma Sistem

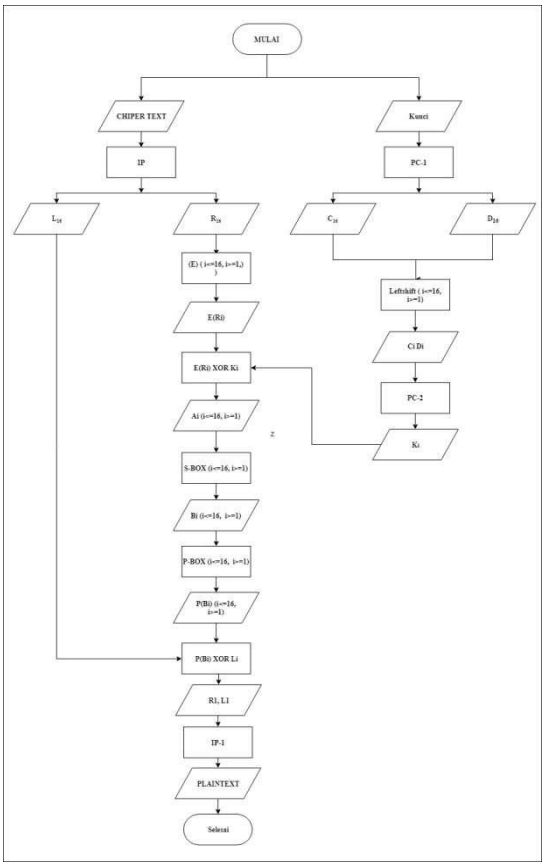
Algoritma sistem merupakan penjelasan langkah-langkah dalam penyelesaian masalah dalam perancangan sistem keamanan data produksi harian dengan menggunakan algoritma DES. Hal ini dilakukan untuk meningkatkan keamanan data tamu tersebut [3].

2.3 Flowchart dari metode penyelesaian

Berikut ini adalah *flowchart* dari proses enkripsi dan dekripsi dari algoritma DES yaitu sebagai berikut:



Gambar 3.1 *flowchart* proses enkripsi



Gambar 3.2 *flowchart* Proses Dekripsi

2.4 Proses Enkripsi

Proses enkripsi adalah mengubah suatu data plaintext ke chippertext. Dalam proses enkripsi terdapat beberapa langkah-langkah berikut:

1. Mengubah *Plaintext* dan *Key* menjadi biner

   Mengubah plaintext kedalam biner berdasarkan tabel ASCII.

Tabel 3.1 Konversi plaintext ke biner

PLAINTEXT			
	DEC	hexa	biner
M	77	4D	01001101
a	97	61	01100001
i	105	69	01101001
s	115	73	01110011
a	97	61	01100001
r	114	72	01110010
a	97	61	01100001
h	104	68	01101000

Mengubah *key* ke dalam biner berdasarkan tabel ASCII

KEY			
	Dec	Hexa	Biner
K	75	4B	01001011
R	82	52	01010010
I	73	49	01001001
S	83	53	01010011
T	84	54	01010100

KEY			
I	73	49	01001001
N	78	4E	01001110
A	65	41	01000001

2. *Initial Permutation* (IP) *Plaintext*

Lakukan *initial permutation* (IP) pada bit plaintext menggunakan tabel IP seperti berikut:

Tabel 3.2 Initial permutation

PLAINTEXT (X)							
0	1	0	0	1	1	0	1
0	1	1	0	0	0	0	1
0	1	1	0	1	0	0	1
0	1	1	1	0	0	1	1
0	1	1	0	0	0	0	1
0	1	1	1	0	0	1	0
0	1	1	0	0	0	0	1
0	1	1	0	1	0	0	0

IP1							
58	50	42	34	26	18	10	2
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
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Keterangan pada tabel *initial permutation* dan tabel IP(X):
Angka 0 dan 1 merupakan bilangan biner
Angka 1,2,3 dan seterusnya yang adalah urutan posisi bit
Urutan bit ke-58 pada tabel *plaintext* (X), di letakan pada posisi 1 pada tabel IP,
Urutan bit ke-50 pada tabel *plaintext* (X), di letakan pada posisi 2 pada tabel IP,
Urutan bit ke-42 pada tabel *plaintext* (X), di letakan pada posisi 3 pada tabel IP,
Demikian seterusnya dan menghasilkan Tabel IP(X).

Tabel 3.3 IP(X)

TABEL IP (X)								
1	1	1	1	1	1	1	1	L0
0	0	1	0	1	0	0	0	
0	0	0	0	0	0	0	1	
0	1	0	1	1	1	1	1	
0	0	0	0	0	0	0	0	R0
1	1	1	1	1	1	1	0	
1	0	0	0	0	1	0	1	
0	0	1	0	1	0	0	0	

Selanjutnya bit pada IP(X) di pecah menjadi dua bagian yaitu L0 dan R0 sehingga hasilnya dapat di lihat pada tabel

3. Melakukan Permutasi *Key* Kompresi PC-1

Kunci yang sudah diubah menjadi bilangan biner, lalu di permutasikan dengan menggunakan tabel permutasi kompresi PC-1, pada langkah ini terjadi kompresi 64 bit menjadi 56 bit.

Tabel 3.4 Permutasi Kompresi PC-1

KEY							
0	1	0	0	1	0	1	1
0	1	0	1	0	0	1	0
0	1	0	0	1	0	0	1
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	0

PC1						
57	49	41	33	25	17	9
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

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

7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Keterangan pada tabel Permutasi Kompresi PC-1
Angka 0 dan 1 merupakan bilangan biner
Angka 1,2,3 dan seterusnya adalah urutan posisi bit
Urutan bit ke-57 pada tabel key, diletakan pada posisi 1 pada Tabel PC-1,
Urutan bit ke-49 pada tabel key, diletakan pada posisi 2 pada Tabel PC-1 dst, dan hasil permutasi key dapat di lihat pada tabel 3.7.

Tabel 3.5 Permutasi Kompresi PC-1

Tabel PC-1							
0	0	0	0	0	0	0	**C0**
0	1	1	1	1	1	1	
1	1	0	0	0	0	0	
0	0	0	0	0	0	1	
0	1	0	0	1	0	1	**D0**
1	0	1	0	1	0	0	
0	0	0	1	1	0	0	
1	0	1	1	0	1	0	

Selanjutnya bit pada Tabel hasil permutasi PC-1 di pecah menjadi dua bagian yaitu C0 dan D0 sehingga hasilnya sebagai berikut.
C0: 0000000 0111111 1100000 0000001
D0: 0100101 1010100 0001100 1011010

4. Melakukan Pergeseran Kiri (*Left Shift Operation*)

Lakukan pergeseran kiri (*left Shift Operation*) pada C0 dan D0 sebanyak satu atau dua kali berdasarkan putaran yang ada pada tabel putaran sebagai berikut:

Tabel 3.6 Left shif

Putaran ke – i	**Jumlah Pergeseran Bit (Left Shift)**
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Keterangan:
Untuk putaran ke-1, dilakukan pergeseran 1bit ke kiri,
Untuk putaran ke-2, dilakukan pergeseran 1 bit ke kiri,
Untuk putaran ke-3, dilakukan pergeseran 2 bit ke kiri, dan seterusnya hingga putaran yang ke-16.
Berikut hasil dari *left shift*:
C0 : 0000000 0111111 1100000 0000001
D0 : 0100101 1010100 0001100 1011010
Putaran ke-1, di geser 1 bit ke kiri.

C1 : 0000000 1111111 1000000 0000010
D1 : 1001011 0101000 0011001 0110100
Putaran ke-2, di geser 1 bit ke kiri.
C2 : 0000001 1111111 0000000 0000100
D2 : 0010110 1010000 0110010 1101001
Putaran ke-3, di geser 2 bit ke kiri.
C3 : 0000111 1111100 0000000 0010000
D3 : 1011010 1000001 1001011 0100100
Putaran ke-4, di geser 2 bit ke kiri.
Dst..
C16 : 0000000 0111111 1100000 0000001
D16 : 0100101 1010100 0001100 1011010
Setiap hasil putaran digabungkan kembali menjadi CiDi dan diinput kedalam tabel Permutation Compression 2 (PC-2) dan terjadi kompresi data CiDi 56 bit menjadi CiDi 48 bit dan mengkasilkan Ki.

Tabel 3.7 Permutation Compression 2 (PC-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

Keterangan:
Urutan bit pada C_iD_i yang ke-14, diletakan di posisi 1 pada tabel PC-2,
Urutan bit pada C_iD_i yang ke-17, diletakan di posisi 2 pada tabel PC-2,
Urutan bit pada C_iD_i yang ke-11, diletakan di posisi 3 pada tabel PC-2, dan seterusnya.
Berikut hasil *output*nya:
C_1D_1 : 0000000 1111111 1000000 0000010 1001011 0101000 0011001 0110100
K_1 : 101000 001001 001001 001010 010100 001100 011010 100011
C_2D_2 : 0000001 1111111 0000000 0000100 0010110 1010000 0110010 1101001
K_2 : 101000 000001 001011 010010 001000 001111 100111 001100
C_3D_3 : 0000111 1111100 0000000 0010000 1011010 1000001 1001011 0100100
K_3 : 001101 000101 001001 010000 001000 001001 010010 110111
Dst..
$C_{16}D_{16}$: 0000000 0111111 1100000 0000001 0100101 1010100 0001100 1011010
K_{16} : 101000 011001 001000 100010 010011 110010 000001 101100

5. Melakukan Ekspansi Data

Pada langkah ini, kita akan meng-ekspansi data $R_{i-}1$ 32 bit menjadi R_i 48 bit sebanyak 16 kali putaran dengan nilai perputaran 1>= i <=16 menggunakan Tabel Ekspansi (E).

Tabel 3.8 Ekspansi

Tabel Ekspansi					
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

Hasil $E(R_{i-1})$ kemudian di XOR dengan K_i dan menghasilkan Vektor Matriks A_i. Berikut hasil *output*nya:

Iterasi 1

E(R1)-1) : 000000 000001 011111 111101 010000 001010 100101 010000
K1 : 101000 001001 001001 001010 010100 001100 011010 100011
---------- XOR
A1 : 101000 001000 010110 110111 000100 000110 111111 110011

Pada iterasi satu (1) diatas didapat A1 dari hasil XOR E(R1-1) dan K1, setelah itu maka proses selanjutnya langsung ke langkah ke-6 terlebih dahulu, dimana $A_{i,}$ akan dimasukan ke dalam S-BOX dan menghasilkan PB1 yang kemudian di XOR kan dengan L0 dan menghasilkan nilai R_i. Nilai R_i ini digunakan untuk melanjutkan iterasi ke-2.

6. Memasukan data ke dalam S-BOX

A1 : 101000 001000 010110 110111 000100 000110 111111 110011

Tabel 3.9 Subtitusi S1

	0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Kemudian kita ambil sampel blok bit pertama yaitu **101000**, pisahkan menjadi 2 blok yaitu:

1. Bit pertama dan terakhir yaitu 1 dan 0, digabungkan menjadi 10
2. Bit kedua hingga kelima yaitu 0100

Selanjutnya dibandingkan dengan memeriksa perpotongan antara kedua di dapatkan nilai 13 (warna kuning) lalu dibinerkan menjadi **1101**

Tabel 3.10 Subtitusi S2

	0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
00	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
01	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
10	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
11	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Kemudian kita ambil sampel blok bit kedua yaitu **001000**, pisahkan menjadi 2 blok yaitu:

1. Bit pertama dan terakhir yaitu 0 dan 0, digabungkan menjadi 0
2. Bit kedua hingga kelima yaitu 0100

Selanjutnya dibandingkan dengan memeeriksa perpotongan antara kedua di dapatkan nilai 6 (warna kuning) lalu dibinerkan menjadi **0110.** Dan seterusnya untuk blok ketiga hingga blok kedelapan dibandingkan dengan S3 dan S8. Berdasrkan cara diatas diperoleh hasil sebagai berikut:

B1 : 11010110 01111011 01001111 11001100

7. Memutasikan bit vector B_i

Setelah didapatkan nilai vektor B_i, langkah selanjutnya adalah memutasikan bit vektor B_i menggunakan tabel P-*BOX*, lalu dikelompokkan menjadi 4 blok dimana setiap blok memiliki 32 bit data.

Tabel 3.11 Matrik Permutasi P (P-*box*)

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 hasil yang didapatkan sebagai berikut:

P(B-1) : 11011100 11110101 10100000 01111111

Hasil P(B_i) kemudian di XOR kan dengan L_i-1 untuk mendapatkan nilai R_i. Sedangkan nilai L_i sendiri diperoleh dari nilai R_i-1 untuk nilai 1>=i<=16.

P(B1) : 11011100111101011010000001111111
L0 : 11111111001010000000000101011111
---XOR
R1 : 00100011110111011010000100100000
Untuk mencari R2 sampai R16, lakukan langkah yang sama dari langkah 5 sampai 7 dan dituliskan dalam bentuk itrasi.

Iterasi 2

E(R1) : 00010000011111101111101111010000001010010000000
K2 : 10100000000100101101001000100000111110011100 1100
---XOR
A2 : 10110000011011000010100111110000011010000011001100
P(B2) : 01011100010010100010101101010100
L1 : 00000000111111101000010100101000
---XOR
R2 : 01011100101101001010111001111100

Iterasi 3

E(R2) : 00101111100101011010100101010101110000111111 1000
K3 : 00110100010100100101000000100000100101001011 0111
---XOR
A3 : 00011011110001111111100101110101010101110100 1111
P(B3) : 01010101000000000011000001101011
L2 : 00100011110111011010000100100000
---XOR
R3 : 01110110110111011001000101001011
Dst..

Iterasi 16

E(R15) : 11001010000000100000011010111111100000111010 1011
K16 : 10100001100100100010001001001111001000000110 1100
---XOR
A16 : 01101011100100000010010011110000101000111100 0111
P(B16) : 10001000101000000000010101000111
L15 : 01111001000100010011110001100011
---XOR
R16 : 11110001101100010011100100100100
L16 : 10010000010000110111110001110101

8. Menggabungkan R16 dan L16

Langkah terakhir adalah menggabungkan R_{16} dengan L_{16} kemudian dipermutasikan dengan tabel *initial permutation* (IP-^{1}).

Tabel 3.12 Permutasi R16 dan L16 dengan Tabel IP^{-1}

R16 dan L16								
R16	1	1	1	1	0	0	0	1
	1	0	1	1	0	0	0	1
	0	0	1	1	1	0	0	1
	0	0	1	0	0	1	0	0
L16	1	0	0	1	0	0	0	0
	0	1	0	0	0	0	1	1
	0	1	1	1	1	1	0	0
	0	1	1	1	0	1	0	1

TABEL IP-1							
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 3.13 Chipertext

CHIPERTEKS							
0	1	1	1	0	1	1	0
0	0	1	0	0	0	0	0
0	0	0	0	1	0	1	1
0	0	0	0	1	1	0	0
1	1	0	1	1	1	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	0
1	1	0	1	0	0	0	0

Menghasilkan *output*:
Chiper dalam biner : 01110110 00100000 00001011 00001100 11011110 01011111 01101010 11010000
Atau dalam *chiper* hexa : 76 20 B C DE 5F 6A D0

2.5 Proses Dekripsi

Untuk dapat mengetahui isi pesan sebenarnya, perlu dilakukan konversi *ciphertext* menjadi bentk biner untuk mendapatkan bit *chipertext*. Dekripsi dapat dilakukan sebagai berikut:

1. Melakukan permutasi terhadap *chipper*

Chiper dalam biner : 10111110 10100100 10101010 00101111 11010101 01110101 01100010 11100011
Atau dalam *chiper* hexa : BE A4 AA 2F D5 75 62 E3
Atau dalam *chiper* plaintext : ¾ ¤ ª / Õ u b ã

Tabel 3.14 *Initial permutation chipper* (IP)

Ciphertext							
0	1	1	1	0	1	1	0
0	0	1	0	0	0	0	0
0	0	0	0	1	0	1	1
0	0	0	0	1	1	0	0
1	1	0	1	1	1	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	0
1	1	0	1	0	0	0	0

Tabel IP							
58	50	42	34	26	18	10	2
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
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Tabel 3.15 Hasil initial permutation chipper *(IP)*

IP(Cipher)								
1	1	1	1	0	0	0	1	L0
1	0	1	1	0	0	0	1	
0	0	1	1	1	0	0	1	
0	0	1	0	0	1	0	0	
1	0	0	1	0	0	0	0	R0
0	1	0	0	0	0	1	1	
0	1	1	1	1	1	0	0	
0	1	1	1	0	1	0	1	

Selanjutnya bit pada IP (*Chiper*) dipecah menjadi 2 bagian yaitu L0 dan R0,
Sehingga menghasilkan sebagai berikut:
L0 : 11110001101100010011100100100100
R0 : 10010000010000110111110001110101

Iterasi 16
P(B16) : 10001000101000000000010101000111
L15 : 11110001101100010011100100100100
---------- XOR
R16 : 01111001000100010011110001100011

Iterasi 15
P(B15) : 10010100111001111111000100010100
L14 : 10010000010000110111110001110101

---XOR
R15 : 00000100101001001000110101100001
Iterasi 14
P(B14) : 10100111000011010110101110010111
L13 : 01111001000100010011110001100011
---XOR
R14 : 11011110000111000101011111110100
Iterasi 1
P(B1) : 11011100111101011010000001111111
L0 : 00100011110111011010000100100000
---XOR
R1 : 11111111001010000000000101011111
L1 : 00000000111111101000010100101000

2. Melakukan permutasi R1 dan L1 dengan table IP-1

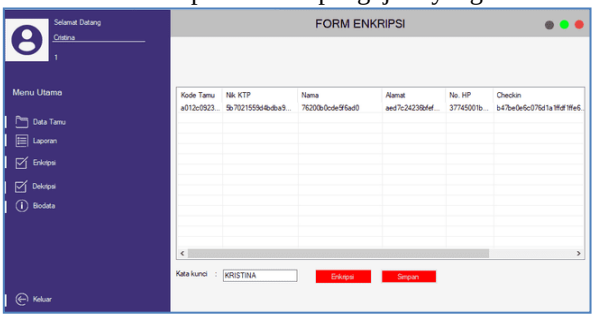
Kemudian R_1 dan L_1 di permutasikan kembali dengan tabel *inverse initial permutation* sehingga menghasilkan *output*:

Plaintext dalam biner : 01001101 01100001 01101001 01110011 01100001 01110010 01100001 01101000
Atau dalam bentuk hexa: 4D 61 69 73 61 72 61 68
Dan dalam bentuk *plaintext:* Maisarah

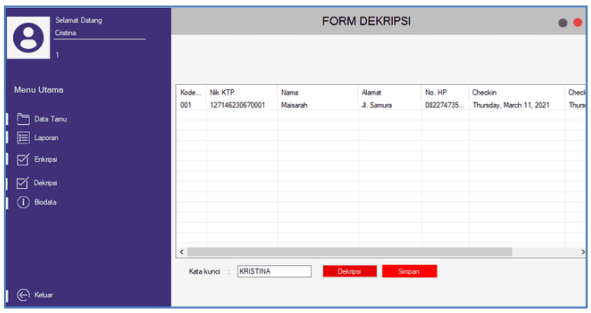
3. ANALISA DAN HASIL

Uji coba sistem bertujuan untuk membuktikan bahwa *input, proses, output* yang dihasilkan oleh sistem aplikasi *Visual Studio* 2012 telah benar dan sesuai dengan yang diinginkan. Pengujian sistem dengan cara memasukkan data ke dalam sistem dan memperhatikan *output* yang dihasilkan. Jika *input,* proses dan *output* telah sesuai, maka sistem telah benar. Berikut merupakan tahapan untuk pengujian sistem yaitu:

1. Melakukan *input* data tamu yang kemudian sistem akan menampilkan data tamu yang tersimpan di *database*.
2. Menggunakan bahasa pemrograman *Microsoft Visual Studio* 2012 dalam pengolahan data yang disimpan dalam *database Microsoft Office Access* 2010 . Penggunaan sistem pengamanan data tamu pada Hotel Sibayak Multi, agar dapat berjalan dengan baik *file* aplikasi *Visual Studio* 2012 harus ditempatkan pada satu *folder* dan dilengkapi dengan *input* data dari analisa sistem. Lokasi *folder* yang telah ditentukan adalah tempat untuk menyimpan *file-file* yang telah dikumpulkan, untuk menghindari kesalahan sebaiknya data tidak diletakkan kedalam *folder* yang berbeda. Selanjutnya untuk menerapkan metode dalam mengamankan data tamu, maka data tersebut akan *diinput* ke aplikasi lalu simpan data tersebut ke dalam *database Access*. Jalankan aplikasi *Visual Studio* 2012 yang telah terinstal dikomputer. Berikut ini merupakan hasil pengujian yang dilakukan pada sistem.



Gambar 5.1 Pengujian untuk data tamu enkripsi



Gambar 5.2 Pengujian untuk data tamu dekripsi

3.2 Kelemahan dan kelebihan sistem

Berikut ini diuraikan kelemahan dan kelebihan dari sistem:

1. Kelemahan Sistem
   Dalam sistem tentunya masih ada kekurangan dan kelemahan. Adapun kelemahan yang ada di dalam sistem adalah:
   a. Sistem yang dibangun tidak dapat diakses secara online, sehingga sistem hanya dapat digunakan secara lokal saja.
   b. Hasil ini hanya digunakan pada kasus di Hotel Sibayak Multi, tidak di perusahaan lain.
2. Kelebihan Sistem
   Hasil yang didapat dari pengujian sistem ini mempunyai kelebihan- kelebihan antara lain :
   a. Proses Pengamanan Data
      Bagi pengguna sistem khususnya pada Hotel Sibayak Multi yang ingin menggunakan sistem ini, cukup meng*input* data tamu yang akan dijadikan sebagai objek pengamanan data, kemudian melakukan proses enkripsi, maka hasil yang di dapat yaitu sebuah *cipherteks*, data tamu tersebut diamankan dengan menggunakan kombinasi kriptografi algoritma *data encryption standard* sehingga sulit untuk mengetahui dan membaca data tamu tersebut.
   b. Menjalankan Program
      Program yang dibangun bersbasis *desktop programming*, walaupun tidak terhubung jaringan ataupun internet sistem tetap dapat untuk dijalankan dan dapat membantu pihak karyawan pada Hotel Sibayak Multi dalam mengamankan data tamu.

4. KESIMPULAN

Berdasarkan pembahasan dan evaluasi dari bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut :

1. Dalam menganalisa masalah yang terjadi terkait dengan pengamanan data tamu pada Hotel Sibayak Multi algotritma *Data Encryption Standard* (DES) maka dilakukan proses enkripsi untuk data tamu.
2. Perancang sistem kriptografi yang mengadopsi algoritma DES (*Data Encryption Standard)* dengan metode sistem *Block Cipher* di dalam menyelesaikan masalah terkait pengamanan data tamu pada Hotel Sibayak Multi menggunakan pemrograman yang berbasis desktop.
3. Pengujian sistem ini dilakukan sebelum nantinya dapat dicoba untuk membantu Hotel Sibayak Multi terkait di dalam pengamanan data tamu pada Hotel Sibayak Multi.

REFERENSI

[1] Deny Adhar, "Implementasi Algoritma DES (Data Encryption Standard) pada enkripsi dan dekripsi berbasis android ," vol. 3, pp. 53-57, Juli 2019.

[2] Dina Fitra Murad, Fachruddin, "nalisaDanImplementasiAlgoritmaEnkripsiSimetrisDataEncryptionStandard (DES) Pada Raspberry Pi," vol. XI, pp. 55-56, Mei 2019.

[3] F.H Habibie, "Pembangunan Sistem Informasi Penerimaan Calon Tenaga Kerja Secara Online Berbasis Website Pada Bursa Kerja Khusus SMK Ganesha Tama Boyolali," *J. tekno*, vol. 5, pp. 77-83, 2014

[4] Sejati Waluyo, "Sistem Keamanana Management File Menggunakan Algoritma Advanced Encryption Standard," pp. 639-640, 2018.

[5] S. Mathur, "Analysis and design of enhanced RSA algorithm to improve the security," pp. 3-7, 2017.

[6] H. Pandiangan, *Perancangan Media Pengiriman Pesan Teks Dengan Penyandian Pesan Menggunakan Algoritma Rc4 Berbasis Web*, vol. 1, pp. 62-71, 2016.

BIBLIOGRAFI PENULIS

Nama : Kristina Meldawati Mahulae
Tempat Lahir : Pekanbaru
Tanggal Lahir : 17 Desember1998
Jenis Kelamin : Perempuan
Agama : Kristen Protestan
Warga Negara : Indonesia
Status : Lajang
Alamat : Pusuk 2 Simaningggir
Email : kristinamahulae18@gmail.com
Bidang Keahlian : Sistem Informasi

Nama : Faisal Taufik S.Kom.,Kom
NIDN : 0104038603
Tempat Lahir : Kisaran
Tanggal Lahir : 04 Maret 1986
Agama : Islam
Program Studi : Sistem Informasi
Email : faisal.taufik@trigunadharma.ac.id
Deskripsi : Dosen Tetap STMIK TRIGUNA DHARMA yang aktif mengajar dan fokus pada bidang keilmuan pemrograman komputer.

Nama : Usti Fatimah Sari Sitorus Pane S. Kom., M.kom
NIDN : 0120089101
Tempat lahir :Lingga Tiga
Tanggal Lahir :20 Agustus 1991
Agama :Islam
Program Studi : Sistem Informasi
Email : ustipaneee@gmamil.com
Riwayat pendidikan:
=> S1: STMIK TRIGUNA DHARMA (2009-2013)
=> S2: UPI YPTK PADANG (2014-2016)
Prestasi : Dosen Terbaik tahun 2019
Deskripsi : Dosen tetap STMIK TRIGUNA DHARMA yang aktif mengajar dan fokus di bidang ilmu komputer dengan bidang keilmuan embedded system dan sistem digital.