IMPLEMENTASI ENKRIPSI DAN DESKRIPSI DATA SIAK (SISTEM INFORMASI ADMINISTRASI KEPENDUDUKAN) MENGGUNAKAN ALGORITMA DES ,AES DAN MD5

Putra Adi wijaya[1], Meilani Damanik[2],Puspita Hartati[3],Indra Gunawan[4]
Email : putraadiwijaya265@gmail.com
Sistem Informasi STIKOM Tunas Bangsa Pematangsiantar

ABSTRAK.

Perkembangan teknologi yang semakin berkembang pesat setiap saatnya mengakibatkan terbukanya cela bagi oknum oknum yang tidak betanggung jawab untuk masuk kedalam suatu program atau data,guna untuk mengetahui suatu data /program yang bersifat rahasia. Dari sekian banyak manfaat atau dampak positif yang di timbulkan oleh perkembangang teknologi, Banyak Juga Dampak-Dampak negatif yang di timbulkan oleh perkembangan teknologi tersebut.

Yang di antaranya adalah tentang keamanan yang semakin mudah di retas oleh setiap orang di karenakan akses untuk masuk ke suatu program atau data seseorang semakin muda, karna adanya dukungan akses internet dan teknologi lainnya.

Oleh sebab itu kita harus mengantisipasi secara tepat agar tidak menjadi masala yang berkesinambungan di masa yang akan datang.

Pada penelitian ini dilakukan pengamanan data siak menggunakam 3 metode yang berbeda yaitu metode DES (*data encryption standart),*AES *(advanced encryption standart),MD5(Message Digest 5)* .ketiga metode ini berguna untuk meng enkripsi data. DES Lebih dulu di perkenalkan di dunia komputer, sementara AES adalah metode yang di ciptakan guna untuk meminimalisir kekurangan yang terdapat di dalam algoritma des Sementara MD5 adalah algoritma yang di buat guna untuk mempermuda dalam enkripsian data.

Kata-kunci : komputer,des,aes,md5

The development of technology that is growing rapidly every time results in the opening of reproach for unscrupulous individuals who are not responsible for entering into a program or data, in order to find out a data / program that is confidential. Of the many benefits or positive impacts caused by the development of technology, there are also many negative impacts caused by the development of these technologies.

Which of them is about security that is increasingly easy to crack by everyone because access to enter a program or data is getting younger, because of the support of internet access and other technologies.

Therefore we must anticipate precisely so as not to become a sustainable problem in the future.

In this research, siak data security is done using 3 different methods, namely DES (data encryption standard) method, AES (advanced standard encryption), MD5 (Message Digest 5). These three methods are useful for encrypting data. DES First introduced in the world of computers, while AES is a method created to minimize the deficiencies contained in the des algorithm

While MD5 is an algorithm that is made to make it easier in data encryption.

Keywords: computer, des, aes,md5

PENDAHULUAN

Pada dasarnya sistem administrasi kependudukan merupakan sub sistem dari sistem administrasi Negara, yang mempunyai peranan penting dalam pemerintahan dan pembangunan penyelenggaraan administrasi kependudukan.rahasia yang terdapat di dalam data tersebut bersifat rahasia, oleh karna itu perlu pengamanan yang khusus, guna untuk mengamankan data tersebut agar tidak di ketahui oleh oknum-oknum yang tidak betanggung jawab.

Pada jurnal ini kami mencoba melakukan sebuah penelitian untuk mengamankan suatu data administrasi kependudukan. Di mana kami megambil sampel dari data penduduk kabupaten simalungun. Dengan menggunakan algoritma DES (*data encryption standart),* AES *(advanced encryption standart),*dan MD5(*Message Digest 5)* kami mencoba mengamankan data siak itu tersebut dengan tujuan agar data tersebut tidak di ketahui oleh orang-orang yang tidak bertanggung jawab.

Pengamanan data ini kami lakukan guna untuk memperkecil tingkat kejahatan yang memanfaatkan data diri penduduk, yang merupakan suatu data yang tidak seharusnya di ketahui oleh banyak orang. Kurangnya penanggulangan untuk menanggulangi masala ini,mengakibatkan semakin banyaknya data kependudukan yang di salah gunakan,contohnya sebagai data palsu untuk orang yang bukan prmilik data tersebut untuk menjalankan niat dan tujuannya.

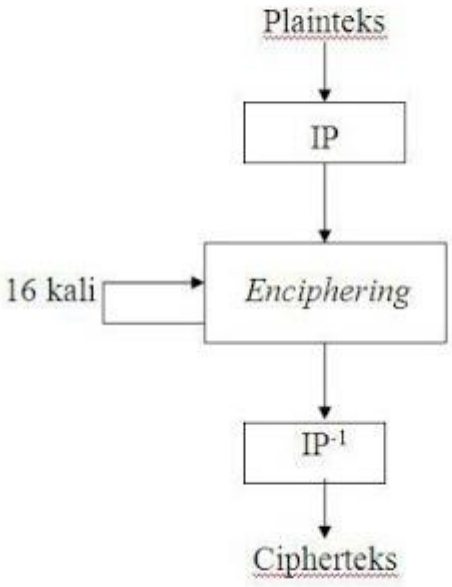
Dengan menggunakan 3 algoritma yang berbeda yaitu Algoritma DES (*data encryption standart),*AES *(advanced encryption standart) dan MD5(Message Digest 5)* data siak yang bersifat rahasi tersebut dapat teramankan secara efisien.

METODELOGI

Data Encryption Standard (DES) adalah salah satu algoritma kriptografi simetris, artinya kunci yang digunakan untuk proses enkripsi sama dengan kunci yang digunakan untuk proses dekripsi. Algoritma DES ini juga merupakan algoritma enkripsi block-chiperdengan panjang blok 64 bit dan dengan panjang kunci 56 bit yang bersifat rahasia yang dibagi (shared secret). Shared secret sendiri merupakan sepenggal data yang hanya diketahui oleh pihak-pihak yang melakukan komunikasi, dalam hal ini yaitu pengirim pesan dan penerima pesan. Yang dimaksud sepenggal data di sini dapat berupa kata sandi (password), passphrase, atau kunci pada algoritma enkripsi. Saat ini DES sudah hampir tidak digunakan lagi karena panjang kunci yang hanya 56 bit itu amat dengan mudah dibongkar dengan serangan Brute Force. Menggunakan prosesor tercepat saat tulisan ini dibuat, DES dapat dibongkar hanya dalam waktu beberapa menit. Algoritma lain yang dianggap sebagai ganti dari algoritma DES ialah algoritma AES (Advanced Encryption Standard).

CARA KERJA DATA ENCRYPTION STANDARD

Cara kerjanya adalah dengan mengubah pesan asli yang dapat dimengerti/dibaca manusia (plainteks) ke bentuk lain yang tidak dapat dimengerti/dibaca oleh manusia (cipherteks). Proses transformasi plainteks menjadi chiperteks diistilahkan dengan enkripsi



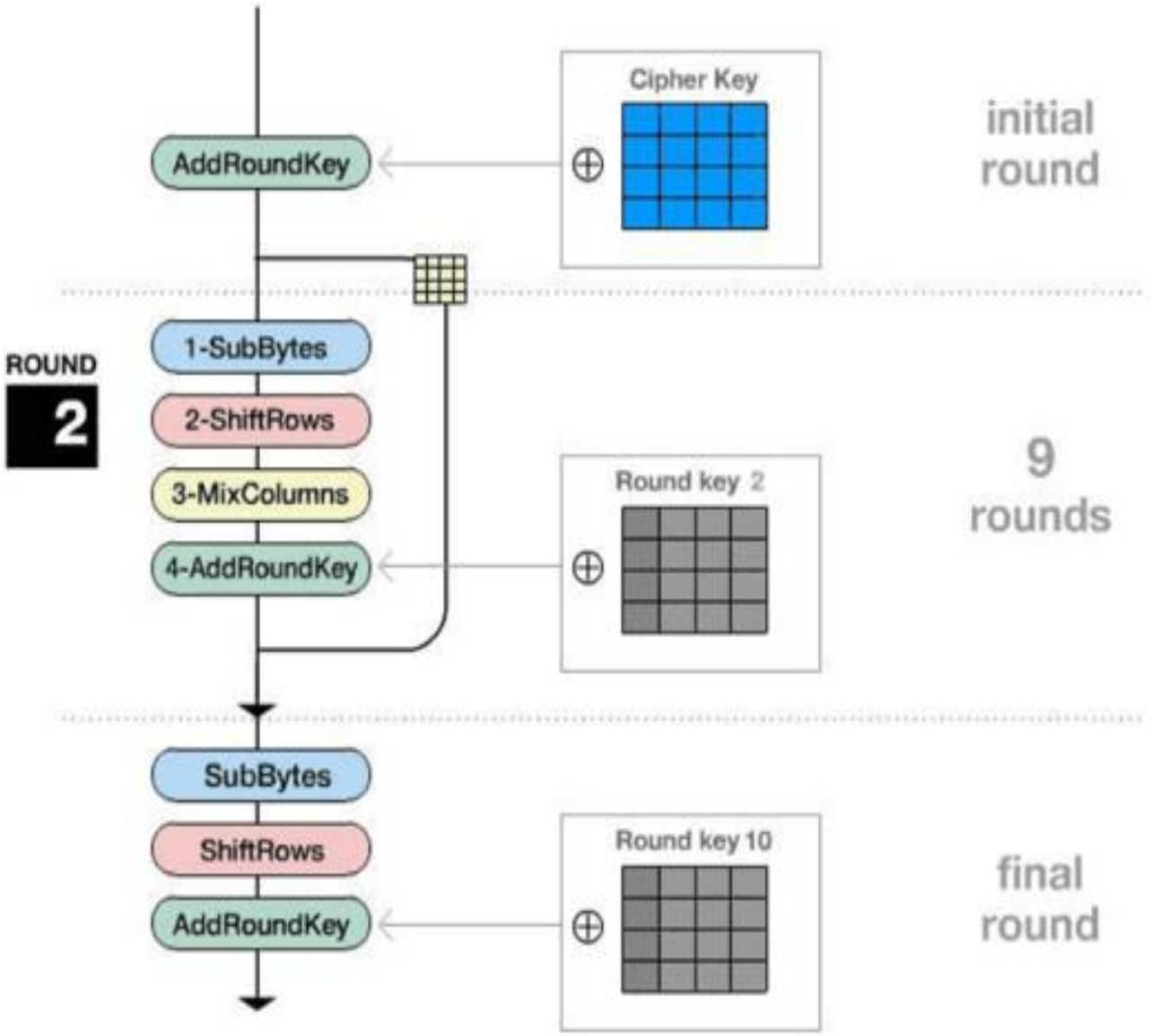
Advance Encryption Standard (AES)

Advanced Encryption Standard (AES) merupakan algoritma cryptographic yang dapat digunakan untuk mengamankan data. Algoritma AES adalah blok chipertext simetrik yang dapat mengenkripsi (encipher) dan dekripsi (decipher) info rmasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut ciphertext; sebaliknya dekripsi adalah merubah ciphertext data menjadi bentuk semula yang kita kenal sebagai plaintext.

AES (Advanced Encryption Standard) adalah lanjutan dari algoritma enkripsi standar DES (Data Encryption Standard) yang masa berlakunya dianggap telah usai karena faktor keamanan. Kecepatan komputer yang sangat pesat dianggap sangat membahayakan DES, sehingga pada tanggal 2 Maret tahun 2001 ditetapkanlah algoritma baru Rijndael sebagai AES.

AES memiliki ukuran block yang tetap sepanjang 128 bit dan ukuran kunci sepanjang 128, 192, atau 256 bit. Berdasarkan ukuran block yang tetap, AES bekerja pada matriks berukuran 4x4 di mana tiap-tiap sel matriks terdiri atas 1 byte (8 bit). Sedangkan **Rijndael** sendiri dapat mempunyai ukuran matriks yang lebih dari itu dengan menambahkan kolom sebanyak yang diperlukan.

Blok chiper tersebut dalam pembahasan ini akan diasumsikan sebagai sebuah kotak. Setiap plainteks akan dikonversikan terlebih dahulu ke dalam blok-blok tersebut dalam bentuk heksadesimal. Barulah kemudian blok itu akan diproses dengan metode yang akan dijelaskan. Secara umum metode yang digunakan dalam pemrosesan enkripsi dalam algoritma ini dapat dilihat melalui gambar berikut:



2. PEMBAHASAN

SAMPEL DATA SIAK (SISTEM INFORMASI ADMINISTRASI KEPENDUDUKAN)

DATA SIAK (SISTEM INFORMASI ADMINISTRASI KEPENDUDUKAN)						
NO		NO KK	NAMA	NIK	JENIS KELAMIN	TANGGAL LAHIR
1	1	1208021402080713	PAIDI	1208020704650001	LAKI-LAKI	4-Jul-1965
	2		ROMINAH	1208027112660055	PEREMPUAN	31-12-1966
	3		DARA ELZA PUTRI	1208026701080002	PEREMPUAN	27-01-2008
	4		SUKILAH	1208027112380008	PEREMPUAN	31-12-1938
2	5	1208021402080666	WAKIDI	1208021202520001	LAKI-LAKI	2-Dec-1952
	6		MESINEM	1208024911690001	PEREMPUAN	11-Sep-1969
	7		ERWIN GUNAWAN	1208021912000001	LAKI-LAKI	19-12-2000
3	8	1208023008160003	ANDI FARIKO	1208022306900001	LAKI-LAKI	23-06-1990
	9		TIKA SILPIYA	1208217006950004	PEREMPUAN	30-06-1995
	10		ANDRIAN ANUGRAH RAMADHAN	1208020307160002	LAKI-LAKI	3/7/2016
4	11	1208021602170004	BILLY AZHAR ZEIN	1208021807890002	LAKI-LAKI	18-07-1989
	12		RIA ANJELA	1208025708970003	PEREMPUAN	17-08-1997
	13		REYFALDI AZHAR	1208020602160002	LAKI-LAKI	2-Jun-2016
5	14	1208022507130001	MARIA	1208027112690034	PEREMPUAN	31-12-1969
6	15	1208021210110004	JUNI AGUSTINA	1208026608770003	PEREMPUAN	26-06-1977
	16		YELSY PUIGA UTARI	1208026408980003	PEREMPUAN	24-08-1998
7	17	1208021402080633	JUMINA	1208027112590005	PEREMPUAN	31-12-1959
8	18	1208022912140006	SUMINEM	1208025512370001	PEREMPUAN	16-12-1937
9	19	1208021501150010	ERNI JULIANA	1208024411900001	PEREMPUAN	11-Apr-1990
	20		MUHAMMAD ADITYA	1208020904140004	LAKI-LAKI	4-Sep-2014
10	21	1208020607170005	WAGINI	1208027006710001	PEREMPUAN	30-06-1971
	22		WIDIA PIRAMITA	1208024509010001	PEREMPUAN	9-May-2001
	23		MUZAKI PRASETYO	1208022101060002	LAKI-LAKI	21-01-2006
	24		HASRIRA KHAIFANI	1208025701100002	PEREMPUAN	17-01-2010
11	25	1208021602160004	YANTI	1208027112660054	PEREMPUAN	31-12-1966
	26		RANI	1208026401970002	PEREMPUAN	24-01-1997
12	27	1208022008100018	BERTA TAMBUNAN	1208027112470014	PEREMPUAN	31-12-1947
	28		ADELIA PUTRI NAIPOSPOS	1208025710900001	PEREMPUAN	17-10-1990

PROSES ENKRIPSI MENGGUNAKAN ALGORITMA DES , AES , DAN MD5

Login	
User id	Password
admin	Admin
computer	1334 57799BBCDFF1
teknik	**74 65 6b 6e 69 6b**

Dari table di atas kami akan mencoba mengambil bebarapa contoh untuk meng enkripsi data untuk login agar keamanan data kependudukan yang kita amankan terjamin dan tidak dapat di ketahui oleh banyak orang melainkan hanyalah orang-orang yang berhak mengetahuinya saja.

Data yang berwarna kuning adalah data yang akan kami coba enkripsi menggunakan ketiga metode di atas, di mana ini bertujuan untuk mengamankan dan memperkecil celah untuk masuk yaitu orang-orang yang tidak bertanggung jawab dan ingin menggunakan data penduduk guna untuk melaksanakan niat yang ingin dia wujudkan,

LANGKAH PERTAMA MENGGUNAKAN ALGORITMA DES

Ubahlah plaintext kedalam bentuk biner
C : 01000011
O : 01001111
M : 01001101
P : 01010000
U : 01010101
T : 01010100
E : 01000101
R : 01010010

Ubahlah key kedalam bentuk biner

13 : 00010011
34 : 00110100
57 : 01010111
79 : 01111001
9B:10011011
BC:10111100
DF:11011111
F1:11110001

Langkah Kedua:

Lakukan Initial Permutation (IP) pada bit plaintext menggunakan tabel IP berikut:

Tabel Initial Permutation (IP)

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

Urutan bit pada plaintext urutan ke 58 ditaruh diposisi 1,

Urutan bit pada plaintext urutan ke50 ditaruh di posisi 2,

Urutan bit pada plaintext urutan ke42 ditaruh di posisi 3, dst

Sehingga hasil outputnya adalah:

IP(x) : 11111111101110000111011001010111 00000000000000000000011010000011

Pecah bit pada IP(x) menjadi 2 bagian yaitu:

L^0:11111111101110000111011001010111(tabel IP dengan warna kuning)

R^0:00000000000000000000011010000011(tabel IP dengan warna hijau)

Langkah Ketiga:

Generate kunci yang akan digunakan untuk mengenkripsi plaintext dengan menggunakan tabel permutasi kompresi PC-1, pada langkah ini terjadi kompresi dengan membuang 1bit masing-masing blok kunci dari 64bit menjadi 56bit.

Tabel PC-1

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

Dapat kita lihat pada tabel diatas, tidak terdapat urutan bit 8, 16, 24, 32, 40, 48, 56, 64 karena telah dikompress. Berikut hasil outpunya:

CD(k) : 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

Pecah CD(k) menjadi dua bagian kiri dan kanan, sehingga menjadi

C_0:11110000110011001010101111 (tabelPC-1 warna kuning)

D_0:0101010101100110011110001111 (tabelPC-1 warna hijau)

Langkah Keempat:

Lakukan pergeseran kiri (LeftShift) pada C_0 dan D_0, sebanyak 1 atau 2 kali berdasarkan kali putaran yang ada pada tabel putaran sebagai berikut:

Tabel Left Shift

Putaran ke- i	Jumlah Pergeseran (Left
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

Untuk putaran ke1, dilakukan pegeseran 1bit kekiri

Untuk putaran ke 2, dilakukan pergeseran 1 bit kekiri.

Untuk putaranke 3, dilakukan pergeseran 2bit kekiri,dst

Berikut hasil outputnya:
C_0:11110000110011001010101011111
D_0:01010101011001100111100011111
Digeser 1bit kekiri
C_1:11100001100110010101010111111
D_1:10101010110011001111000111110
Digeser bit kekiri
C_2:11000011001100101010101111111
D_2:01010101100110011110000111101
Digeser 2bit kekiri
C_3:00001100110010101010111111111
D_3:01010110011001111000111101010
Digeser bit kekiri
C_4:00110011001010101011111111100
D_4:01011001100111100011110101010
Digeser 2bit kekiri
C_5:11001100101010101111111110000
D_5:01100110011110000111101010101
Digeser 2bit kekiri
C_6:00110010101010111111111000011
D_6:10011001111000111101010101010
Digeser 2bit kekiri
C_7:11001010101011111111100001100
D_7:01100111100011110101010110
Digeser 2bit kekiri
C_8:00101010101111111110000110011
D_8:10011110001111010101011001
Digeser 1bit kekiri
C_9:01010101011111111100001100110
D_9:00111100011110101010110011
Digeser 2bit kekiri
C_{10}:01010101111111110000110011001
D_{10}:11110001111010101011001100
Digeser 2bit kekiri
C_{11}:01010111111110000110011001 01
D_{11}:11000111101010101100110011
Digeser 2bit kekiri
C_{12}:01011111111000011001100101 01
D_{12}:00011110101010101100110011 11
Digeser 2bit kekiri
C_{13}:01111111100001100110010101 01
D_{13}:01111010101010110011001111 00
Digeser 2bit kekiri
C_{14}:11111110000110011001010101 01
D_{14}:11101010101011001100111100 01
Digeser 2bit kekiri
C_{15}:11110000110011001010101 11
D_{15}:10101010101100110011110001 11
Digeser 1bit kekiri
C_{16}:11110000110011001010101011 11
D_{16}:01010101011001100111100011 11
Setiap hasil putaran digabungkan kembali menjadi C_iD_i dan diinput kedalam table Permutation Compression 2(PC-2) dan terjadi kompresi data C_iD_i

56bit menjadi CiDi 48bit.

TabelP -2

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

Berikut hasil outputnya:

C_1D_1 = 1110000 1100110 0101010 1011111

10101010110011001111000111
10
K_1 = 000110 110000 001011 101111 111111

0001110000011100
10
K_4 = 011100101010 110111 010110 110110

1100110101000111
01

C_5D_5 = 1100110 0101010 1011111 1110000
01100110011110000111101010101
K_5 = 011111001110 110000 000111 111010
11010100111010100
C_6D_6 = 0011001 0101010 1111111 1000011
10011001111000111101010101010
K_6 = 011000111010 010100 111110 010100
00011110110010111
C_7D_7 = 1100101 0101011 1111110 0001100
01100111100011110101010110

K_7 = 111011001000 010010 110111 111101
100001100010111100

C_8D_8 = 0010101 0101111 1111000 0110011
10011110001111010101011001

K_8 = 111101111000 101000 111010 110000
010011101111111011

C_9D_9 = 0101010 1011111 1110000 1100110
00111100011110101010110011

K_9 = 111000001101 101111 101011 111011
011110011110000001
$C_{10}D_{10}$ = 01010101111111 1000011 0011001
11110001111010101011001100

K_{10} = 101100011111 001101 000111 101110
100100011001001111

$C_{11}D_{11}$ = 01010111111110 0001100 1100101
11000111101010101100110011

K_{11} = 001000010101 111111 010011 110111
101101001110000110

$C_{12}D_{12}$ = 01011111111000 0110011 0010101
00011110101010110011001111

K_{12} = 011101010111 000111 110101 100101
000110011111101001

$C_{13}D_{13}$ = 01111111100001 1001100 1010101
01111010101011001100111100

K_{13} = 100101111100 010111 010001 111110
101011101001000001

$C_{14}D_{14}$ = 11111110000110 0110010 1010101
11101010101011001100111110001

K_{14} = 010111110100 001110 110111 111100
101110011100111010

$C_{15}D_{15}$ = 11111000011001 1001010 1010111
10101010101100110011111000111

K_{15} = 101111111001 000110 001101 001111
0100111111000010
10

$C_{16}D_{16}$ = 11110000110011 0010101 0101111
01010101011001100111100011
11

K_{16} = 110010110011 110110 001011 000011
1000010111111101
01

Langkah Kelima:

Pada langkah ini ,kita akan meng-ekspansi data R_{i-1}

32bit menjadi R_i 48bit sebanyak 16 kali putaran dengan nilai perputaran 1< = I < =16 menggunakan Tabel Ekspansi (E).

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 $E(_{Ri-1})$ kemudian di XOR dengan Kidan menghasilkan Vektor Matriks Ai.

Berikut hasil

outputnya: Iterasi 1

$E(R_{(1)-1})$=10000000000000000000000000000000
0011010100000001
10

K_1

=00011011000000101110111111111
1
00011100000111001
0

---------- XOR

A_1 =10011011000000101110111111111111
00101001000111010
0

Berhubung bagian dibawah ini yang paling ribet, maka saya tambahkan keterangan ditengah-tengah proses iterasi. Bisa kita lihat pada iterasi 1 diatas setelah kita dapat kan hasil XOR antara E(R(1)-1) dengan K1 dan menghasilkan A1, maka proses berikutnya langsung masuk ke LANGKAH KEENAM terlebih dahulu, dimana A1 akan dimasukan kedalam S-Box dan menghasilkan output B1.

B1 kemudian akan dipermutasikan lagi dengan tabel P-Box dan menghasilkan nilai PB1 yang kemudian di XOR-kan dengan L0 dan menghasilkan nilai R1. Nilai R1 ini digunakan untuk melanjutkan iterasi ke-2
.

Iterasi–
2

E(R(2)-1) =
01101010111010000101011010011
0
10010101000000110
1

K2 =
01111001101011101101100111011
0
11110010011110010
1

----------XOR

A2 = 00010011010001101000111101000
0
01100111011110100
0

Iterasi–
3

E(R(3)-
1)=01000101011111101110011110001
0
01010101001010000
1

K3 =
01010101111111001000101001000
0
10110011111001100
01

XOR A3 =

00010000100000100111100110000
1
11100110110011100
00

Iterasi
–4

E(R(4)-
1)=01011111000101011111001111010
1
01110000111111000
01

K4 =
01110010101011011101011011011
0
11001101010001111
01

XOR A4 =

00101101101110000010010100001
1
10111101101110
11

Iterasi
–5

E(R(5)-1)=11011010100101110000010101100
1
01101010011010001
1

K5
=01111100111011000000011111110
1
0
11010100111010100
0

XOR

A5 =10100110011110110000001010001
1
10111110100000101
1

Iterasi–6

E(R(6)-1)=10010101101111000101011010111
0
10110000011111101
0

K6 =011000111010010100111110010100
000111101100101111

----------XOR A6

=111101100001100101101000111010
101011101011010101

Iterasi–7

E(R(7)-1)=110010100001011111110010100111
111101011001010011

K7 =111011001000010010110111111101
100001100010111100

----------XOR

A7 = 001001101001001101000101011010
011100111011101111

Iterasi–8

E(R(8)-1)=111100001010101001010101010011
110000001010100011

K8 = 111101111000101000111010110000
010011101111111011

----------XOR

A8 = 000001110010000001101111100011
100011100101011000

Iterasi–9

E(R(9)-1)=010010101111111000000000000010
101111110101010001

K9 = 111000001101101111101011111011
011110011110000001

----------XOR

A9 =101010100010010111101011111001

110001101011010000

Iterasi– 10

E(R(10)-1)= 100111 111000 001110 100010
100111110111111000001010

K10 =101100011111001101000111101110

100100011001001111

----------XOR

A10 =001011100111000011100101001001
010011100001000101

Iterasi– 11

E(R(11)-1)= 010011 110111 111010 101010
101111110011110001011001

K11 =001000010101111111010011110111
101101001110000110

----------XOR

A11 = 011011100010000101111001011000
011110111111011111

Iterasi– 12

E(R(12)-1)= 001001 011010 101001 011111
110001010111110010101100

K12 =011101010111000111110101100101
000110011111101001

----------XOR

A12 =010100001101101110101010010100
010001101101000101

Iterasi– 13

E(R(13)-1)= 100110 100111 110111 111011
111110101110101100001010

K13 =100101111100010111010001111110
101011101001000001

----------XOR

A13 =000011011011100000101010000000
000101000101001011

Iterasi– 14
E(R(14)-1)= 111001 010111 110000 001000
001000001000001011111011
K14 =010111110100001110110111111100
101110011100111010

----------XOR

A14 =101110100011111110111111110100
100110010111000001
Iterasi– 15
E(R(15)-1)= 000110 101100 001100 000001
011001011010100101010100

K15
=101111111001000110001101001111
010011111100001010

----------XOR

A15 = 101001010101001010001100010110
001001011001011110

Iterasi– 16

E(R(16)-1) = 101101 011101 010100 000101
010101010001010110100010

K16 =110010110011110110001011000011
100001011111110101

----------XOR

A16 = 011111101110100010001110010110
110000001001010111

Langkah Keenam:

Setiap Vektor A_i disubstitusikan kedelapan buah S-Box(Substitution Box), dimana blok pertama disubstitusikan dengan S_1, blok kedua dengan S_2 dan seterusnya dan menghasilkan output vector B_i 32bit.

S4 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
01	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
11	3	15	0	6	10	1	13	18	9	4	5	11	12	7	2	14

S5 :

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

S6 :

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

S7 :

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

S8 :

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

Cara menggunakan S-Box :
(Untuk detail penggunaan S-Box, silahkan lihat dihalaman http://en.wikipedia.org/wiki/S-box)

Kita ambil contoh S1, kemudian konversi setiap angka didalam tabel S1 yang berwarna putih menjadi biner, sehingga menjadi bentuk seperti dibawah:

S1 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

Kemudian kita ambil sampel blok bit pertama dari A1 yaitu 100110

Kita pisahkan blok menjadi 2 yaitu:

- Bitpertamadanterakhiryaitu1dan0digabungkan menjadi 10
- Bitkeduahingga ke lima0011

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

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

S3 :

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

Kemudian dibandingkan dengan memeriksa perpotongan antara keduanya didapatkan nilai 1000 (warna merah) dan seterusnya untuk blok kedua hingga blok kedelapan kita bandingkan dengan S2 hingga S8.

Berdasarkan cara diatas diperoleh hasil sebagai berikut:

B1= 1000 0101 0100 1000 0011 0010 1110 1010

B2= 1101 1100 0100 0011 1000 0000 1111 1001

B3= 1101 0110 0011 1100 1011 0110 0111 1111

B4= 0010 1001 1101 0000 1011 1010 1111 1110

B5= 0100 0001 0011 1101 1000 1010 1100 0011

B6= 0110 1101 1101 1100 0011 0101 0100 0110

B7= 1110 0011 0110 1011 0000 0101 0010 1101

B8= 0000 1000 1101 1000 1000 0011 1101 0101

B9= 0110 1110 1110 0001 1010 1011 0100 1010

B10=00100001011100000100000101101101

B11=01011110000011001101101111000010

B12=01101000000010110011011010101101

P(B5)=10010101001100101101100001000000

P(B6)= 00100100000110111111001111111000

P(B7)= 11001000110000011110111001101100

P(B8)= 00000111001110010010100101100001

P(B9)= 11011001001110111010001110010100

P(B10)= 00001100000101010110111000100100

P(B11)= 01110001001111101011000001010011

P(B12)= 10101000011010001000111011101001

P(B13)= 10000110110010111100111111001011

P(B14)= 00000101110111010011101001001111

P(B15)= 10100101001001101110110011101100

P(B16)= 00101001111101110110100011001100

B13=11111001110110110010010010110011

B14=10111000011111101100010111000001

B15=01000001001110011111011100100111

B16=10000001011010101111011101001011

Langkah Ketujuh:

Setelah didapatkan nilai vector Bi, langkah selanjutnya adalah memutasikan bit vector Bi menggunakan tabel P-Box, kemudian dikelompokkan menjadi 4 blok dimana tiap-tiap blok memiliki 32 bit data.

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 hasil yang didapat adalah sebagai berikut:

P(B1)= 00101000101100110100010011010001

P(B2)= 10001011110110011000110000010011

P(B3)= 01101111101100101001110011111110

P(B4)= 00111111001110111010001111010001

Hasil P(Bi) kemudian di XOR kan dengan Li-1 untuk mendapatkan nilai Ri.

Sedangkan nilai Li sendiri diperoleh dari Nilai Ri-1 untuk nilai 1 < = I < = 16.

L0 = 11111111101110000111011001010111

R0 = 00000000000000000000011010000011

P(B1) = 00101000101100110 1000100 11010001

L(1)-1 =11111111101110000111011001010111

----------XOR

R1 = 11010111000010110011001010000110

P(B2) = 10001011110110011 0001100

00010011

L(2)-1 =00000000000000000000011010000011

---------- XOR

R2 = 10001011110110011000101010010000

P(B3) = 0110111110110010 10011100
11111110

L(3)-1 =11010111000010110011001010000110

----------XOR R3 =

10111000101110011010111001111000

P(B4) = 0011111100111011 01000111
10100001

L(4)-1 =10001011110110011000101010010000

----------XOR R4 =

10110100111000101100110100110001

P(B5) = 1001010100110010 11011000
01000101

L(5)-1 =10111000101110011010111001111000

----------XOR R5 =

00101101100010110111011000111101

P(B6) = 0010010000011011 11110011
11111000

L(6)-1 =10110100111000101100110100110001

----------XOR R6 =

10010000111110010011111011001001

P(B7) = 1100100011000001 11101110
01101100

L(7)-1 =00101101100010110111011000111101

----------XOR R7 =

11100101010010101001100001010001

P(B8) = 0000011100111001 00101001
01100001

L(8)-1 =10010000111110010011111011001001

----------XOR R8

= 10010111110000000001011110101000

P(B9) = 1101100100111011 10100011
10010100

L(9)-1 =11100101010010101001100001010001

----------XOR R9

= 00111100011100010011101111000101

P(B10)=00001100000101010110111000100100

L(10)-1 = 10010111 11000000 00010111
10101000

----------XOR R10

= 10011011110101010111100110001100

P(B11)=01110001001111101011000001010011

L(11)-1 = 00111100 01110001 00111011
11000101

----------XOR R11

= 01001101010011111000101110010110

P(B12)=10101000011010001000111011101001

L(12)-1 = 10011011 11010101 01111001
10001100

----------XOR R12
= 00110011101111011111011101100101

P(B13)=10000110110010111100111111001011

L(13)-1 = 01001101 01001111 10001011
10010110

----------XOR R13
= 11001011100001000100010001011101

P(B14)=00000101110111010011101001001111

L(14)-1 = 00110011 10111101 11110111
01100101

----------XOR R14 = 00110110011000001100110100101010
P(B15)=10100101001001101110110011101100

L(15)-1 = 11001011 10000100 01000100

01011101

----------XOR R15 =01101110101000101010100010110001

P(B16)=00101001111101110110100011001100

L(16)-1 = 00110110 01100000 11001101

00101010

----------XOR R16 =00011111100101111010010111100110
L16 =01101110101000101010100010110001

Langkah terakhir adalah menggabungkan R_{16} dengan L_{16} kemudian dipermutasikan untuk terakhir kali dengan tabel Invers Initial Permutasi (IP^{-1}).

TabelIP^{-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

Sehingga Input:

$R_{16}L_{16}$= 00011111100101111010010111100110 01101110101000101010100010110001

Menghasilkan Output:

Cipher (dalam biner) = 01010110 11110001 11010101110010000101001010101111100000001 00111111 atau

Cipher(dalamhexa)= 56f1d5c8 52af813f

ENKRIPSI MENGGUNAKAN METODE AES DAN MD5

Sebelum mengenkripsi pesan, terlebih dulu harus dicari berapa fungsi hash dari “teknik” agar bisa digunakan sebagai kunci untuk mengenkripsi pesan “unib”. Perhitungan algoritma MD5 adalah sebagai berikut :

1. Tentukan pesan yang akan di *hashing*

   Pesan (String) : teknik

   Pesan (Hex) : **74 65 6b 6e 69 6b**

2. Penambahan bit

   Pesan ditambahkan dengan bit pengganjal hingga kongruen dengan 448 modulo 512 bit. Pesan : **74 65 6b 6e 69 6b**

   Pesan terdiri dari 6 karakter atau 6 byte = 6 x 8 = 48 bit, sehingga perlu ditambahkan sebesar 448 – 48 = 400 bit.

3. Pada bit terakhir sepanjang 64 bit diisi dengan nilai panjang pesan, dimana panjang pesan adalah 6 byte = 6 x 8 bit = 48 bit atau dalam hex = 30

4. Pengolahan pesan di dalam blok 16 word

   Pada MD-5 juga terdapat 4 (empat) buah fungsi nonlinear yang masing - masing digunakan pada tiap operasinya (satu fungsi untuk satu blok), yaitu:

   F(X,Y,Z) = XY v not(X) Z G(X,Y,Z) = XZ v Y not(Z) H(X,Y,Z) = X xor Y xor Z I(X,Y,Z) = Y xor (X v not(Z))

   Pengolahan pesan dilakukan sebanyak 4 putaran, tiap putaran dilakukan sebanyak 16 operasi dasar, yaitu sebagai berikut :

   Round (1,16)

   A = 2249092517 dalam Hex = 860E6DA5 B = 2931300306 dalam Hex = AEB817D2 C = 712877834 dalam Hex = 2A7DA70A D = 229885237 dalam Hex = 0DB3C535 Round (2,16)

   A = 3914487489 dalam Hex = E95256C1
   B = 2441348034 dalam Hex = 918403C2
   C = 3652998760 dalam Hex = D9BC5668 D = 28302740 dalam Hex = 01AFDD94 Round (3,16)

A = 4131823715 dalam Hex = F646A063

B = 3065820097 dalam Hex = B6BCB3C1

C = 1815224134 dalam Hex = 6C321F46

D = 73124886 dalam Hex = 045BCC16

Round (4,16)

A = 1331224407 dalam Hex = **4F58DF57 (AA)**

B = 2605068873 dalam Hex = **9B463249 (BB)**

C = 3006515519 dalam Hex = **B333C93F(CC)**

D = 2079023797 dalam Hex = **7BEB62B5 (DD)**

5. Inisialisasi penyangga/buffer MD5,

Pada perhitungan ini menggunakan penyangga berikut (dalam hex)

A = 67452301

B = efcdab89

C = 98badcfe

D = 10325476

Hasil akhir pengolahan pesan dengan 4 putaran yang masing-masing terdiri dari 16 operasi kemudian ditambahkan ke nilai penyangga MD5,

A = A +AA B

= B + BB C

= C + CC D

= D + DD

Hasilnya adalah

A = 3063808600 dalam Hex = B69E0258 B

= 2333334994 dalam Hex = 8B13DDD2 C

= 1273931325 dalam Hex = 4BEEA63D D

= 2350757675 dalam Hex = 8C1DB72B

6. Output operasi MD5 diperoleh dari langkah nomor 6 dengan urutan seperti pada gambar 4

A	4	3	2	1
B	8	7	6	5
C	12	11	10	9
D	16	15	14	13

Gambar 4 Output Operasi MD5

A = B6 9E 02 58

B = 8B 13 DD D2

C = 4B EE A6 3D D = 8C

1D B7 2B

Nilai digest pertama adalah 58, digest kedua 02, digest ketiga 9e dan seterusnya sesuai urutan pada tabel, hingga diperoleh 32 karakter digest MD5 yaitu

58029EB6D2DD138B3DA6EE4B2BB71D8C

Setelah mendapatkan nilai hash, langkah selanjutnya adalah mengenkripsi isi dokumen. perhitungan algoritma AES Enkripsi adalah sebagai berikut :

1. Input

- Plainteks = ‘unib’
- Password = md5(‘teknik’) = 58029eb6d2dd138b3da6ee4b2bb71d8c

2. Inisialisasi

Konversi tiap karakter plainteks dari char ke hex. Plainteks dibagi perblok dengan kapasitas blok 16 karakter. Apabila plainteks tidak cukup 16 karakter atau juga blok terakhir tidak cukup 16 karakter maka sisanya dipadding atau ditambahkan karakter tertentu hingga cukup 16 karakter. Karakter tambahan dapat berupa bit kosong atau bit dengan nilai sisa bloknya. Konversi dari char ke hex dilakukan dengan mengambil nilai ASCII karakter dengan perintah ord(‘karakter’), hasilnya berupa nilai desimal yang kemudian dapat diubah ke hex. Sebagai contoh, karakter ‘u’ dengan perintah ord(‘u’) diperoleh nilai 117, nilai ini bila dikonversi ke hex menjadi 75. Nilai ASCII karakter juga dapat dilihat dari tabel ASCII.

Untuk kata ‘unib’, tidak cukup 16 karakter, sehingga perlu ditambah bit dengan nilai sisa bloknya yaitu 12 (padding). Sehingga plainteks satu blok secara lengkap adalah seperti pada tabel C-3, Plainteks= ‘unib’+padding

□756e69620c0c0c0c0c0c0c0c0c0c0c0c Blok dengan 16 karakter ini kemudian dibagi 4 sehingga menghasilkan 4 bagian dan masing - masing dikonversi ke tipe data word, seperti pada tabel 1.

Tabel 1Konversi Hex ke Word

Bagian	Hexadesimal	Word
1	756e6962	1970170210
2	0c0c0c0c	202116108
3	0c0c0c0c	202116108
4	0c0c0c0c	202116108

3. Penambahan *Round Key*

Proses enkripsi AES 256 dilakukan sebanyak 14 *round*/putaran, setiap putaran membutuhkan Roundkey yang dihasilkan dari ekspansi kunci enkripsi.

Roundkey pertama diperoleh dari konversi kunci/password ke hex. Kunci AES 256 terdiri dari 32 karakter atau 64 karakter setelah dikonversi menjadi hex, setelah dibagi 8 akan menghasilkan 8 bagian. Tiap bagian ini selanjutnya dikonversi ke word.

Kunci = 58029eb6d2dd138b3da6ee4b2bb71d8c Kunci ini kemudian dikonversi ke hex seperti pada tabel 2

Tabel 2 Konversi ke Hex

No	Karakter	Desimal	Hexadesimal
1	5	53	35
2	8	56	38
3	0	48	30
4	2	50	32

Dan seterusnya sehingga menghasilkan : Kunci = 35383032 39656236 64326464 31333862 33646136 65653462 32626237 31643863

Konversi ke word dapat dilihat pada Tabel 3 :

Tabel 3 Konversi ke Word

Bagian	Hexadesimal	Word
1	35383032	892874802
2	39656236	962945590
3	64326464	1681024100
4	31333862	825440354
5	33646136	862216502
6	65653462	1701131362
7	32626237	845308471
8	31643863	828651619

Kunci ini kemudian diekspansi sehingga dapat digunakan untuk 14 putaran AES 256. Adapun proses penambahan Roundkey adalah dengan menggunakan fungsi **XOR** antara plainteks dengan Roundkey yang bersesuaian. Hasilnya adalah seperti pada tabel 4.

Tabel 4 Round Key

Bagian	Word Plaintext	Word Roundkey	PT xor RK
1	1970170210	892874802	1079400784
2	202116108	962945590	896101946
3	202116108	1681024100	1748920424
4	202116108	825440354	1027552366

4. Proses Round AES 256

   Lakukan proses shiftRows + subWord + mixColumns + addRoundKey sebanyak 14 putaran, untuk 13 putaran dihasilkan:

   Round Ke-13
   Hasilnya adalah:

   Bagian 1 => -1356996027
   Bagian 2 => 676864778
   Bagian 3 => 60773198
   Bagian 4 => -473863477

5. SubWord

   Hasil dari ke-13 putaran kemudian di SubWord, yang hasilnya adalah:

   Bagian 1 => 2040830062
   Bagian 2 => 879371879
   Bagian 3 => 2078010671
   Bagian 4 => 293077535

6. ShiftRows dan AddRoundKey

   Hasil SubWord kemudian dilakukan shiftRowdan penambahan RoundKey kembali yang hasilnya adalah :

   Bagian 1 => -1561917226
   Bagian 2 => -201071109
   Bagian 3 => -749947920
   Bagian 4 => -644676683

7. Hasil Enkripsi AES

   Hasil Enkripsi adalah gabungan dari keempat bagian terakhir, yaitu :

   a2 e7 08 d6 f4 03 e5 fb d3 4c b3 f0 d9 93 03 b5

Setelah berhasil mengenkripsi pesan, selanjutnya pengguna akan mendekripsikan kembali pesan yang telah di enkripsi tersebut menjadi *plaintext* kembali agar bisa dibaca. Langkahnya sebagai berikut :

1. Input
   - Cipherteks
     = a2 e7 08 d6 f4 03 e5 fb d3 4c b3 f0 d9 93 03 b5
   - Password
     = md5(‘teknik’)
     =58029eb6d2dd138b3da6ee4b2bb71d8c
2. Inisialisasi

   Kemudian dibagi menjadi 4 bagian dan masing – masing dikonversi ke tipe data word, seperti pada Tabel 5.

Tabel 5 Konversi ke Word

Bagian	Hexadesimal	Word
1	a2e708d6	-1561917226
2	f403e5fb	-201071109
3	d34cb3f0	-749947920
4	d99303b5	-644676683

3. Penambahan Round Key

Proses dekripsi AES 256 juga dilakukan sebanyak 14 *round*/putaran, setiap putaran membutuhkan invRoundkey yang dihasilkan dari ekspansi kunci enkripsi.

InvRoundkey pertama diperoleh dari RoundKey pada proses enkripsi, dimana invRoundeKey pertama = RoundKey pertama, untuk invRoundKey berikutnya dilakukan invMixColumn.

Kunci ini kemudian diekspansi sehingga dapat digunakan untuk 14 putaran AES 256. Adapun proses penambahan invRoundkey juga dilakukan dengan menggunakan fungsi

XOR antara cipherteks dengan Roundkey yang bersesuaian. Pada proses dekripsi round dimulai dari 14, sehingga menggunakan invRoundKey yang ke-14. Hasilnya adalah seperti Tabel 5.

Tabel 5 InvRoundKey

Bagian	Word Cipherteks	Word invRoundkey	CT xor iRK
1	-1561917226	-611457591	2037050655
2	-201071109	-1059526763	886768238
3	-749947920	-1472975977	2071500903
4	-644676683	-935909990	295970351

4. Proses Round Dekripsi AES 256

Lakukan proses shiftRows invShiftRows + invSubBytes + invMixColumns + addRoundKey sebanyak 14 putaran, untuk 13 putaran dihasilkan :

Round Ke-1 Hasilnya adalah:

Bagian 1 => 167331231

Bagian 2 => -1766713261

Bagian 3 => 1165347712

Bagian 4 => 665952069

5. Lakukan invShiftRows + invSubWord + addRoundKey

Hasil round kemudian dilakukan invShiftRows + invSubWord + addRoundKey yang hasilnya :

Bagian 1 => 1970170210

Bagian 2 => 202116108

Bagian 3 => 202116108

Bagian 4 => 202116108

6. Hasil Dekripsi

Hasil Dekripsi adalah gabungan dari keempat bagian terakhir, dalam hexadesimal adalah **756e69620c0c0c0c0c0c0c0c0c0c0c0c, d**alam bentuk string adalah **unib.**

KESIMPULAN

Berdasarkan hasil penelitian, pengujian, implementasi serta pembahasan mengenai Algoritma *Advanced Encryption Standard*, Algoritma data encryption standard dan Algoritma *Message Digest 5* dalam enkripsi dokumen SIAK:

1. File yang dihasilkan oleh proses enkripsi bisa menjadi 2 kali lipat bahkan lebih dari ukuran dan jumlah karakter file aslinya, ini dikarenakan hasil enkripsi dibuat dalam hex. Satu karakter diwakili dua karakter hex, misalkan karakter “U” dalam hex menjadi “75” begitu juga karakter simbol dan spasi juga ada hex-nya. Jadi file hasil enkripsi bisa 2 kali bahkan lebih dari ukuran dan jumlah karakter file aslinya.
2. Dari hasil penelitian, telah dibuktikan bahwa ukuran file hasil enkripsi dan kebutuhan waktu proses dipengaruhi oleh ukuran file asli, namun tidak dipengaruhi oleh jenis format file. Semakin besar ukuran file asli maka semakin besar pula ukuran file hasil enkripsi dan kebutuhan waktu prosesnya.
3. Untuk Menjaga data siak (Sistem Informasi Administrasi kependudukan) Dapat Di Gunakan berbagai macam metode yang di antaranya adalah ketiga metode di atas

REFRENSI

Jurnal Rekursif, Vol. 4 No. 3 September 2016, ISSN 2303 0755.

Kurniawan,Ivan.2012.http://studyinformatics.blogspot.co.id/2012/07/des-data-encryption- standard.html

Practical Approach to Design, Implementation and Management". 1998.

D.a. Rijmen, "AES submission document on Rijndael," 1998.

E. Asliana, "The Procurement of Government Goods and Services in Indonesia," 2012.

F.Widyaningrum, "Implementasi dan Analisis Aplikasi Transfer File Antar PC Menggunakan Algoritma RC4 128 BIT dan AES 128 BIT," 2008.

Inayatullah, "Analisis Penerapan Algoritma MD5 Untuk Pengamanan Password," 2009.

Jogiyanto, H. "*Analisis dan Perancangan Sistem Informasi"*. 2001.

L.UNIB,"http://lpse.unib.ac.id/eproc/tent angkami : diakses pada tanggal 05 Desember 2015."

Pender, T. A. (2002). *UML Weekend Crash Course.* Canada: Wiley Publishing, Inc.

Rosa & Shalahuddin, "Metode Pengembangan Sistem," 2011.

S.Bahri, "Implementasi Pengamanan Basis Data Menggunakan Metode Enkripsi MD5," 2012.

T.T. N. Duc H. Nguyen, Tan N. Duong, Phong H. Pham, "Cryptanalysis of MD5 on GPU Cluster," 2008.

V.Luisiana, "Implementasi Kriptografi Pada Fle Dokumen Menggunakan Algoritma AES-128," 2001.

V.Yuniati, "Enkripsi Dan Dekripsi Dengan Algoritma AES 256 Untuk Semua Jenis File," 2009.