BAB IV

HASIL DAN UJI COBA

IV.1. Uji Coba

Proses uji coba dari aplikasi ini adalah dengan melakukan pengujian langsung dengan memasukkan teks yang nantinya akan di enkrip dan di dekrip dengan menggunakan metode AES128 dan DES

Berikut adalah langkah-langkah yang dilakukan dalam proses pengujian aplikasi :

1. Melakukan instalasi eclipse.
2. Membuka aplikasi yang telah selesai dibuat dengan cara mengimport kedalam *workspace eclipse.*
3. Setelah proses import selesai dengan benar, langkah selanjutnya menginstal aplikasi wifi adb dari *playstore* ke *handphone android.*
4. Buka aplikasi wifi adb tersebut dan hubungkan dengan laptop atau komputer yang sudah terdapat program yang telah dibuat.
5. Lalu jalankan program aplikasi yang sudah dimasukkan tersebut.
6. Jika tidak terjadi kesalahan dalam aplikasi maka aplikasi tersebut akan berjalan sempurna dan terbuka jendela baru.

IV.2. Tampilan Layar

Pada bagian ini merupakan penjelasan dari hasil rancangan *interface* untuk administrator yang terdiri dari sebagai berikut :

1. *Interface Splash*

Adapun hasil dari rancangan *interface splash* dapat dilihat pada gambar IV.1 dibawah ini.

Gambar IV.1 Tampilan *Splash*

2. *Interface Menu*

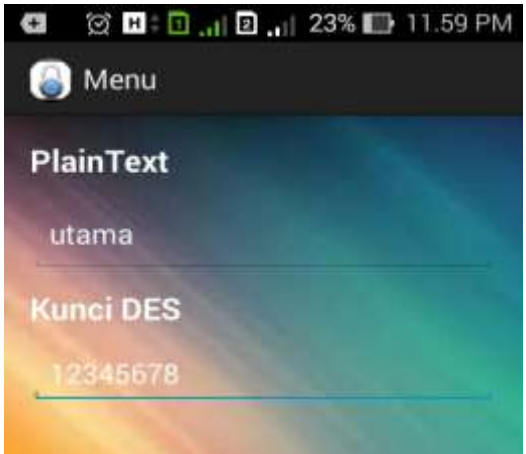
Pada *interface* menu ini menampilkan *form* yang disediakan untuk pengguna mengakses sistem atau untuk menjalankan aplikasi yang telah dirancang, dimana pada menu terdapat *form-form* yang mempunyai fungsi masing-masing. Adapun *interface* menu dapat dilihat pada gambar IV.2 berikut ini

Gambar IV.2 Tampilan Menu DES dan AES

Rancangan *form* ini dibuat untuk sebagai *form* utama dimana di *form* ini ada terdapat 3 *button* dan 1 *textview* yang akan membuka *form* lainnya seperti algoritma DES, algoritma AES128, *exit* dan *about.*

3. *Interface form* Enkrip DES Dan AES

Pada *interface* menu ini menampilkan *form* yang disediakan untuk pengguna dalam mengenkrip teks namun sebelumnya harus memasukan key dan plaintext dengan metode DES dan AES128. Adapun *interface* Enkrip DES dapat dilihat pada gambar IV.3 berikut ini :



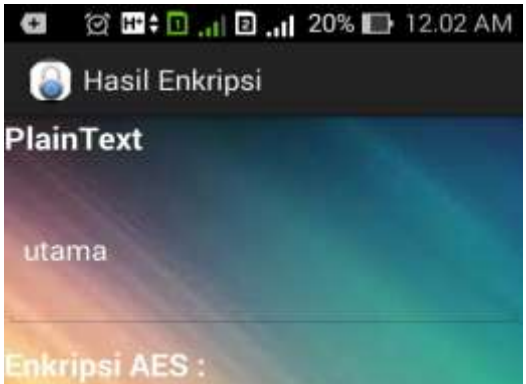
Gambar IV.3 Tampilan *Form* Enkripsi DES dan AES

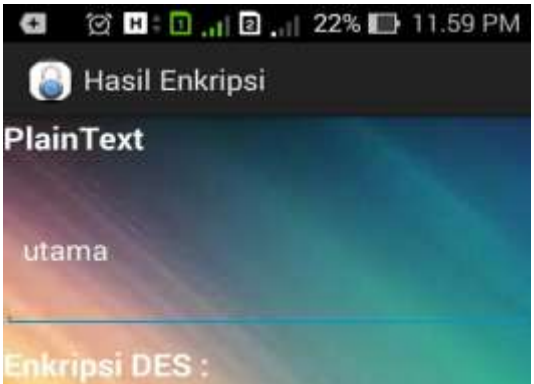
Dimana rancangan form ini dibuat untuk melakukan proses enkripsi teks dimana akan dijelaskan satu persatu tentang form tersebut. Berikut penjelasan form algoritama DES dan AES128.

1. *Text box* menunjukkan tempat untuk memasukkan teks yang nantinya akn di enkrip
2. *Textbox* Kunci tempat untuk memasukkan kunci yang sesuai dengan metode yang digunakan.
3. Tombol/ *button* Enkripsi merupakan tombol untuk memproses pengenkripsian text yang telah di input.
4. Tombol/ *button* kembali merupakan tombol untuk kembali ke menu utama

4. *Interface Form* Hasil Enkripsi DES dan AES

Interface ini merupakan *form* setelah proses enkripsi *file* berjalan lancar. Adapun *interface form* hasil enkripsi dapat dilihat pada gambar IV.4 berikut ini.





Gambar IV.4 Tampilan *Form* Hasil Enkripsi DES dan AES

Dimana rancangan form ini dibuat untuk menampilkan hasil enkripsi teks dimana akan dijelaskan satu persatu tentang form tersebut, berikut penjelasan form algoritama DES dan AES128.

1. *Text box* tempat untuk menampilkan teks hasil enkripsi

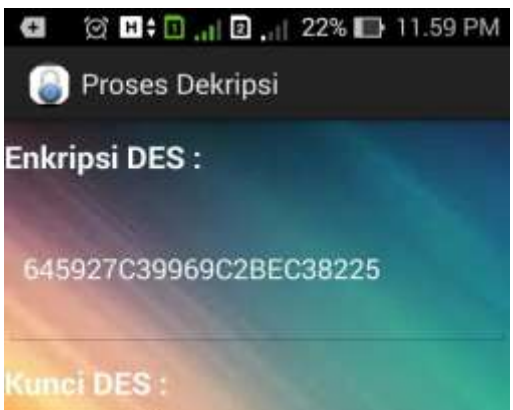
2. *Textbox* tempat untuk menampilkan kunci hasil enkripsi

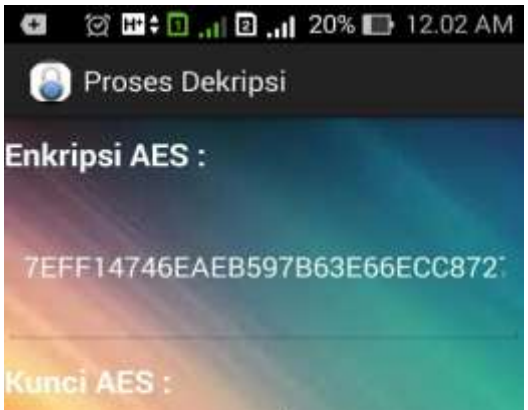
3. Tombol/*button Dekrip* merupakan tombol untuk memproses pengdekripsian selanjutnya.

4. Tombol/*button* kembali merupakan tombol untuk kembali ke menu utama

5. *Interface form* Dekripsi DES Dan AES

Pada *interface* menu ini menampilkan *form* yang disediakan untuk pengguna dalam mengdenkrip teks namun sebelumnya harus memasukan key DES dan AES128 sesuai dengan metode yang di pilih. Adapun *interface* Enkrip DES dapat dilihat pada gambar IV.5 berikut ini :





Gambar IV.5 Tampilan *Form* Dekripsi DES dan AES

Dimana rancangan form ini dibuat untuk melakukan proses dekripsi teks dimana akan dijelaskan satu persatu tentang form tersebut . berikut penjelasan form algoritama DES dan AES128.

1. *Text box* menunjukkan hasil enkrip yang nantinya akan di dekrip.
2. *Textbox* Kunci tempat untuk memasukkan kunci yang sesuai dengan metode yang digunakan awal.
3. Tombol/ *button* Enkripsi merupakan tombol untuk memproses pengdekripsian text yang telah di input.
4. Tombol/ *button* kembali merupakan tombol untuk kembali ke menu utama.

6. *Interface Form* Hasil Dekripsi DES Dan AES

Interface ini merupakan *form* setelah proses dekripsi *file* berjalan lancar. Adapun *interface form* hasil dekripsi dapat dilihat pada gambar IV.6 berikut ini :

Gambar IV.6 Tampilan *Form* Hasil Dekripsi DES dan AES

Dimana rancangan form ini dibuat untuk menampilkan hasil Dekripsi teks dimana akan dijelaskan satu persatu tentang form tersebut . berikut penjelasan form algoritama DES dan AES128.

1. *Text box* tempat untuk menampilkan teks hasil dekripsi
2. *Textbox* tempat untuk menampilkan kunci hasil dekripsi
3. Tombol/*button Dekrip* merupakan tombol untuk memproses pengdekripsian selanjutnya.
4. Tombol/*button* kembali merupakan tombol untuk kembali ke menu utama
5. *Textview* untuk menampilkan konsep kerja AES128 dan DES.

7. *Interface form* Konsep Kerja AES dan DES

Pada *interface* menu ini menampilkan *form* yang berfungsi untuk menjelaskan secara singkat cara kerja dari metode AES128 dan DES mulai dari hal pengengkripan dan pendeskripsian sebuah teks. Adapun *interface* form Konsep Kerja DES dapat dilihat pada gambar IV.7 berikut ini:

Gambar IV.7 Tampilan Konsep Kerja AES dan DES

Dimana rancangan form ini dibuat untuk menampilkan konsep kerja dari kedua metode. Berikut penjelasan form algoritama DES dan AES128.

1. Tombol/*button* kembali merupakan tombol untuk kembali ke menu utama
2. Tombol/*button next* untuk menunjukkan kehalaman selanjutnya.
3. Tombol/*button back* untuk mengembalikan kehalaman sebelumnya.
4. *Textview* untuk menampilkan konsep kerja AES128 dan DES.

8. Tampilan *Form About*

Pada Tampilan *form about* terdapat keterangan dari aplikasi yang dibuat oleh perancang. Adapun Tampilan *form about* dapat dilihat pada gambar IV.8 berikut ini.

Gambar IV.8 Tampilan About

IV.3. *Hardware/Software* Yang Dibutuhkan

Untuk menjalankan program ini dibutuhkan perangkat keras (*hardware*) dan perangkat lunak (*software*)sebagai berikut :

a. Perangkat Keras (*Hardware*)

1. Prosesor *Intel Core i3* atau diatasnya.
2. RAM dengan kapasitas 2Gb
3. *Keyboard, Mouse*
4. *Android Mobile Phone*

b. Perangkat Lunak (*Software*)

1. *SDK Java* sebagai mesin aplikasi *Java* pada aplikasi desktop
2. Sistem operasi *android* pada *mobile phone*

3. Wifi ADB dari *playstore*

IV.4. Analisa Hasil

IV.4.1 Pengujian

Aplikasi Sistem keamanan data dengan algoritma AES128 dan DES, dibangun dengan menggunakan Eclipce dan beberapa software lainya, dan menggunakan bahasa pemograman Java. Aplikasi keamanan data ini membandingkan AES128 dan DES dengan berupa data teks, kemudan teks tersebut akan di enkripsi dan di dekripsi.

Dalam pengujian perbandingan antara AES128 dan DES nantinya diketahui setelah melakukan kalkulasi terhadap kedua metode tersebut dimana perbedaan dari kedua metode tersebut berdasarkan langkah-langkahnya sampai menampilkan hasil dari masing-masing metode tersebut, adapun hasil dari pengujian, dapat dilihat pada tebel IV.1.

Tabel IV.1 Daftar Pengujian Perbandingan Enkripsi AES128 Dan DES

No	Metode	Plaintext	Key	Timer	Hasil Enkripsi	Ket
1	AES 128	123456789ABCDEF	ABCDEFGHIJKLMNOP	0,06 Second	5AD42F600746101FA1C7C2E18542CD6CE48C21ED35A40FACD929A1BF25FC595	Berhasil
		potensi	1234567890123456	0,01 Second	600E2240AC8003A50A56A12A7F5B527B	Berhasil
		utama	1234567890123456	0,00 Second	7EFF14746EAEB597B63E66ECC87276B6	Berhasil
2	DES	activity	2345678	0,5 Second	C2B7C2B1C3ACC2AAC2BBC2A8C3AC28D	Berhasil
		potensi	12345678	0,07 Second	4D58C2B6C28E040971C396	Berhasil

		utama	12345678	0,07 Second	645927C39969C2 BEC38225	Berhasil

Tabel IV.2 Daftar Pengujian Perbandingan Dekripsi AES128 Dan DES

No	Metode	Plaintext	Key	Timer	Hasil Dekripsi	Ket
1	**AES 128**	5AD42F6007 46101FA1C7 C2E18542C D6CE48C21 ED35A40FA CD929A1BF 25FC595	ABCDEF GHIJKLM NOP	0,00 Second	PASSWORD	Berhasil
		600E2240AC 8003A 50A56A12A 7F5B527B	12345678 90123456	0,01 Second	potensi	Berhasil
		7EFF14746E AEB597B63 E66ECC8727 6B6	12345678 90123456	0,00 Second	utama	Berhasil
2	**DES**	C2B7C2B1C 3ACC2AAC 2BBC2A8C3 AC28D	2345678	0,05 Second	activity	Berhasil
		4D58C2B6C 28E040971C 396	12345678	0,07 Second	potensi	Berhasil
		645927C399 69C2BEC38 225	12345678	0,08 Second	utama	Berhasil

IV.5. Uraian Tertulis DES dan AES128

IV.5.1 DES

1. Mengubah Plaintext dan key Menjadi bilangan biner.

   Contoh:

   Plaintext = **a c t i v i t y**

   Key = **13 34 57 79 9B BC DF F1**

Ubahlah *plaintext* dalam bentuk biner

a = 01000001

c = 01000011

t = 01010100

i = 01001001

v = 01010110

i =01001001

t =01010100

y =01011001

Ubah *key* dalam bentuk biner

13 =00010011

34 =00110100

57 =01010111

79 =01111001

9B=10011011

BC=10111101

DF=11011111

F1 =11110001

2. Setelah itu lakukan *Initial Permutation* (IP) pada bit *plaintext* menggunakan tabel IP berikut:

Tabel IV.3. Initial Permutation (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

Hasil *output*nya adalah sebagai berikut :

IP(x) : 11111111 11010100 01010100 10101011 00000000 00000000 10101000 10010010

Kemudian pecah bit pada IP(x) menjadi 2 bagian seperti berikut:

L_0 = 11111111 11010100 01010100 10101011

R_0 = 00000000 00000000 10101000 10010010

3. Langkah selanjutnya generate kunci yang digunakan untuk mengenkripsi *plaintext* dengan menggunakan tabel Permutasi Kompresi (PC-1) dengan membuang 1 bit masing-masing blok kunci dan 64 bit menjadi 58 bit.

Tabel VI.4. Permutasi Choice One (PC-1)

57	49	41	33	25	17	19
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	48	30	2
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Hasil outputnya adalah sebagai berikut :

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

Pecah CD(k) menjadi dua bagian sebagai berikut :

C_0 =1111000 0110011 0010101 0101111

D_0 =0101010 1011001 1001111 0001111

4. Lakukan pergeseran ke kiri (*Left Shift* pada) C_0, D_0 sebanyak 1 atau 2 kali berdasarkan urutan putaran sebagai berikut:

Geser 1 bit ke kiri

C_1 =1110000 1100110 0101010 1011111

D_1 =1010101 0110011 0011110 0011110

Geser 2 bit ke kiri

C_2 =1100001 1001100 1010101 0111111

D_2 =0101010 1100110 0111100 0111101

Geser 2 bit ke kiri

C_3 =1000011 0011001 0101010 1111111

D_3 =0101011 0011001 1110001 1110101

Geser 2 bit ke kiri

C_4 =0001100 1100101 0101011 1111100

D_4 =0101100 1100111 1000111 1010101

Sampai pengeseran ke-16

C_{15} = 1111100 0011001 1001010 1010111

D_{15} = 1010101 0101100 1100111 1000111

Geser 1 bit ke kiri

C_{16} = 1111000 01100110 010101 0101111

D_{16} = 0101010 1011001 1001111 0001111

5. Setiap hasil putaran digabungkan kembali menjadi C_i D_i dan di input kedalam tabel *Permutation Compression* 2(PC-2) untuk mengubah 56 bit menjadi 48 bit.

Tabel IV.5. Choice Permutation Two (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

Berikut hasil Outputnya sebagai berikut:

C_1D_1 = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

Sampai ke-16

C_{16} D_{16} = 1111000 01100110 010101 0101111 0101010 1011001 1001111 0001111

K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101

6. Setelah itu 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 dengan mengunakan tabel Ekspansi(E)

Tabel IV.6. Expantion Permutation (E)

32	**1**	**2**	**3**	**4**	**5**
4	**5**	**6**	**7**	**8**	**9**
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**

7. Kemudian hasil E (R_{i-1}) kemudian di XOR kan dengan K dan menghasilkan vektor Matriks A. Berikut hasil Outputnya:

Iterasi 1

E (R_{i-1}) -1 = 100000 000000 000000 000000 000000 001101 010000 000110

K1 = 000110 110000 001011 101111 111111 000111 000001 110010

----------XOR

A1 = 100110 110000 001011 101111 111111 001010 010001 110100

Sampai iterasi ke 16

E (16) -1 = 101101 011101 010100 000101 010101 010001 010110 100010

K16 = 110010 110011 110110 001011 000011 100001 011111 110101

----------XOR

A16 = 011111 110011 110110 001011 000011 100001 011111 110101

8. Kemudian vektor A disubsitusikan kedelapan buah S-Box (*Substitution S-Box*), dimana blok pertama disubsitusikan dengan S_1 blok kedua dengan S_2 dan seterus dan seterusnya sehingga menghasilkan *output* vektor B_i 32 bit.

   B_1 = 1000 0101 0100 1000 0011 0010 1110 1010

   B_2 = 1101 1100 0100 0011 1000 0000 1111 1001

   Sampai ke-16

   B_{15} = 0100 0001 0011 1001 1111 0111 0010 0111

   B_{16} = 1000 0001 0110 1010 1111 0111 0100 1011

9. Setelah didapatkan nilai dari vektor B, langkah selanjutnya adalah memutasikan bit vektor B menggunakan tabel *P-Box* kemudian dikelompokkan menjadi 4 blok dimana tiap-tiap blok memiliki 32 bit data.

Tabel IV.7. Fungsi Permutasi (P)

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

Sehingga hasil yang didapat adalah sebagai berikut:

$P(B_1)$ = 00101000 10110011 01000100 11010001

$P(B_2)$ = 10001011 11011001 10001100 00010011

Sampai Ke-16

$P(B_{15})$ =10100101

00100110 11101100

11101100

$P(B_{16})$ = 00101001 11110111 01101000 11001100

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 seperti berikut ini :

P(B1) = 00101000 10110011 01000100 11010001
L(1)-1 = 11111111 11010100 01010100 10101011
----------XOR
R1 = 11010111 01100111 00010000 01111010

Sampai dengan 16

P(B15)= 10100101 00100110 11101100 11101100
L(15)-1= 11001011 11101000 01100110 10100001
----------XOR
R15 = 01101110 11001110 10001010 01001101
P(B16)= 00101001 11110111 01101000 11001100
L(16)-1= 10011100 00001000 11101101 11010010
----------XOR
R16 = 10110101 11111111 10000101 00011110

10. L16 dan R16 digabungkan kemudian dipermutasikan untuk terakhir kali dengan tabel *Initial Permutation* (IP) seperti berikut:

Tabel IV.8. Initial Permutation (IP)

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	50	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

Input : 10110101 11111111 10000101 00011110 01101110 11001110 10001010 01001101

Output : 01010110 10111001 11110111 10111011 01010001 11010100 10110110

01111000

Chiper (dalam Hex) : **56B9F7BB51D4B678**

IV.5.2 AES128

proses enkripsi AES128 dengan Algoritma AES

Plaintext = **0 1 2 3 4 5 6 7 8 9 A B C D E F**

Dalam HEX = 30 31 32 33 34 35 36 37 38 39 40 41 42 43 45 46

Kunci = **A B C D E F G H I J K L M N O P**

Dalam HEX = 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50

Sediakan 2 buah matrix 4 x 4 yang setiap kolom metriksnya mewakili 8 bit data input yang berurutan adalah sebagai berikut :

Plaintext HEX :

$$\begin{pmatrix} 30 & 34 & 38 & 43 \\ 31 & 35 & 39 & 44 \\ 32 & 36 & 41 & 45 \\ 33 & 37 & 42 & 46 \end{pmatrix}$$

Kunci HEX :

$$\begin{pmatrix} 41 & 45 & 49 & 4D \\ 42 & 46 & 4A & 4E \\ 43 & 47 & 4B & 4F \\ 44 & 48 & 4C & 50 \end{pmatrix}$$

1. Dilakukan proses XOR pada *plaintext/state* dengan *roundkey*. Proses XOR antar kolom kedua matriks yang bersesuaian setiap kolom menjadi biner. Contoh biner HEX 30 adalah 0011 0000, dan HEX 41 adalah 0100 0001. Jadi, 0011 0000 XOR 0100 0001 = 01110001 HEX 71. Demikian seterusnya proses XOR untuk setiap kolom yang bersesuaian. Langkah ini disebut, yang menghasilkan matriks baru berikut ini :

$$\begin{pmatrix} 71 & 71 & 71 & 0E \\ 73 & 73 & 73 & 0A \\ 71 & 71 & 0A & 0A \\ 77 & 1F & 0E & 16 \end{pmatrix}$$

2. Pada matriks hasil XOR antara *plaintext* dengan *roundkey* dilakukan proses subsitusi dengan S-Box. Hasil subsitusinya sebagai berikut ini :

$$\begin{pmatrix} A3 & A3 & A3 & AB \\ 8F & 8F & 8F & 67 \\ A3 & A3 & 67 & 67 \\ F5 & C0 & AB & 47 \end{pmatrix}$$

3. Pada hasil subsitusi dengan S-Box di lakukan proses *ShiftRows*. Prosesnya sangat sederhana dengan melakukan penggeseran secara wrapping (siklik) pada 3 baris terakhir array state. Junmlah penggeseran bergantung pada nilai baris (r). Baris r=1 digeser ejauh 1 byte hasilnya, baris r=2 digeser 2 byte, dan baris r=3 digeser sejauh 3 byte. Baris r=0 tidak digeser. Hasilnya beruopa matriks seperti berikut ini :

$$\begin{pmatrix} A3 & A3 & A3 & AB \\ 8F & 8F & 67 & 8F \\ 67 & 67 & A3 & A3 \\ 47 & F5 & C0 & AB \end{pmatrix}$$

4. Langkah selanjutnya setelah hasil *ShiftRows* didapat adalah melakukan *MixCoulumns* dengan cara mengalikan matriks hasil *ShiftRows* dengan matriks Rijndael. Transformasi ini dinyatakan sebagai perkalian matriks :

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \times \begin{matrix} A3 \\ 8F \\ 67 \\ 47 \end{matrix} = \begin{matrix} 07 \\ 06 \\ 2B \\ FB \end{matrix}$$

semua state dilakukan dengan cara yang sama, sehingga didapatkan metriks hasil proses *MixColumns* sebagai berikut :

$$\begin{pmatrix} 07 & B5 & C7 & 96 \\ 06 & 41 & CC & 2A \\ 2B & B1 & 91 & 69 \\ FB & E3 & 9A & A8 \end{pmatrix}$$

5. Langkah 1-4 dilakukan sampai putaran ke 9 namun untuk putaran 10 tidak dilakukan *MixColumns* tapi langsung dilakukan proses XOR hasil state *ShiftRows* dengan kunci terakhir.
6. Proses ekspansi kunci 128 bit . ambil 4 byte terakhir dari kunci yaitu 4D 4E AF 50, lalu geser *byte* terakhir, 4E 4F 50 4D. Subsitusikan dengan S-Box, hasilnya adalah 2F 84 53 E3. Dan langkah selanjutnya, XOR-kan dengan konstanta nilai tertentu dari pengguna.

2F XOR 01 = 00101111 XOR 00000001 = 00101110 =2E

84 XOR 00 = 00000100 XOR 00000000 = 10000100 =84

53 XOR 00 = 01010011 XOR 00000000 = 01010011 =53

E3 XOR 00 = 11100011 XOR 00000000 = 11100011 =E3

Kemudian XOR kan 2E 84 53 E3 dengan 4 *byte* baris pertama kunci awal yaitu 41 42 43 44.

2E XOR 41 = 00101110 XOR 01000001 = 01101111 =6F

84 XOR 42 = 10000100 XOR 01000010 = 11000110 =C6

53 XOR 43 = 01010011 XOR 01000011 = 00010000 =10

E3 XOR 44 = 11100011 XOR 01000100 = 10100111 =A7

Hasil proses XOR diatas adalah 6F C6 10 A7 yang merupakan 4 *byte* pertama dari kunci yang baru untuk byte.

45 XOR 6F = 01000101 XOR 01101111 = 00101010 =6F

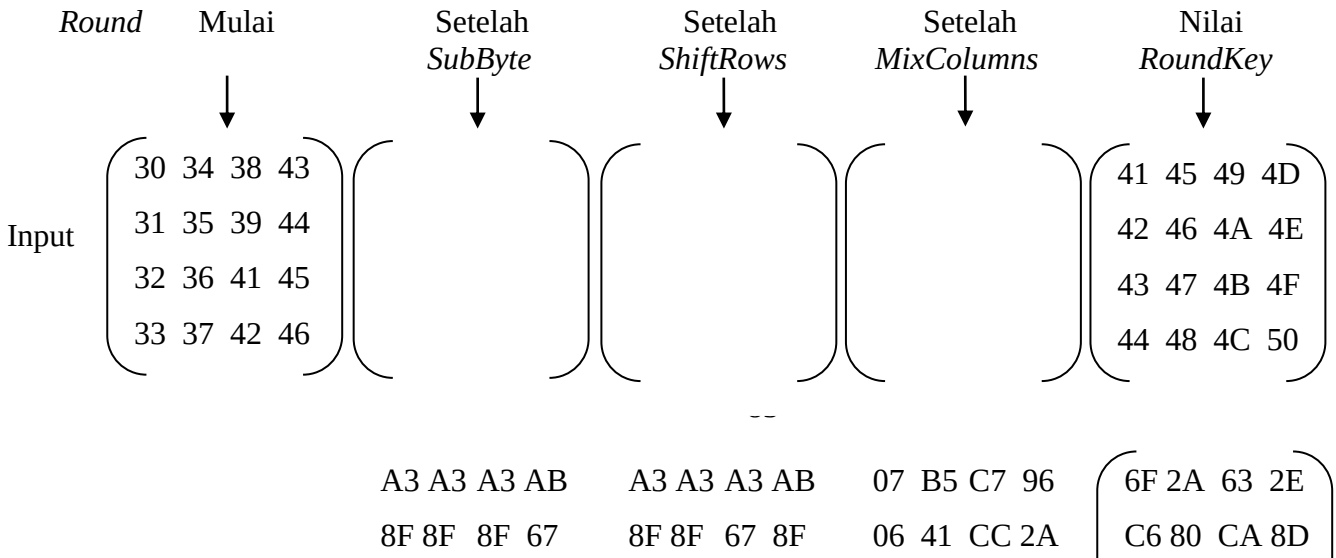
46 XOR C6 = 01000110 XOR 11000110 = 10000000 =C6

47 XOR 10 = 01000111 XOR 00010000 = 01010111 =10

48 XOR A7 = 01001000 XOR 10100111 = 11101111 =A7

Demikian seterusnya hingga didapatkan 16 *byte* set kunci yang baru.

7. Hasil matriks enkripsi menggunakan metode AES dengan kunci 128 bit berikut ini :



1

71 71 71 0E			
73 73 73 0A			
71 71 0A 0A			
71 1F 0E 16			

Sampai ke-10

10

A0 19 DB 75	E0 04 B9 9D	0E D4 B9 9D		BA D3 43 85
54 14 CF 99	20 FA 8A EE	FA 8A EE 20		2E CC F2 74
D4 15 D8 80	48 59 61 E7	61 E7 48 59		4E 56 34 75
58 E9 41 B4	6E 1E 83 8D	8D 6A 1E 83		ED 6B 3D 55

Untuk mendapatkan hasil *chiper text* dari keseluruhan *round* adalah sebagai berikut :

EO XOR BA = 11100000 10111010 = 01011010 = 5A
FA XOR 2E = 11111010 00101110 = 11010100 = D4
61 XOR 4E = 01100001 01001110 = 00101111 = 2F
8D XOR ED = 10001101 11101101 = 01100000 = 60
D4 XOR D3 = 11010100 11010011 = 00000111 = 07
8A XOR CC = 10001010 11001100 = 01000110 = 46
E7 XOR 56 = 11100111 01010110 = 10110001 = B1
6A XOR 6B = 01101010 01101011 = 00000001 = 01

B9 XOR 43 = 10111001 01000011 = 11111010 = FA
EE XOR F2 = 11101110 11110010 = 00011100 = 1C
48 XOR 34 = 01001000 00110100 = 00101111 = 7C
1E XOR 3D = 00011110 00111101 = 01100000 = 2E

9D XOR 85 = 10011101 10000101 = 00010001 = 18

20 XOR 74 = 00100000 01110100 = 01010100 = 54

59 XOR 75 = 01011001 01110101 = 00101100 = 2C

83 XOR 55 = 10000011 01010101 = 11010110 = D6

Maka di dapatkan hasil chiphertext dari keseluruhan *round* adalah sebagai berikut:

5A D4 2F 60 07 46 B1 01 FA 1C 7C 2E 18 54 2C D6

IV.5.3. Perbandingan Algoritma DES Dan AES128 Secara Mendasar

Beberapa perbedaan mendasar antara DES dan AES128 yang telah dijabarkan sebagai berikut:

Tabel IV.9. Perbandingan Antara DES dan AES

Faktor	AES	DES
Panjang Kunci	128.192 dan 256 bit	56 bit
Ukuran Blok	128.192 dan 256 bit	64 bit
Keamanan	Dianggap Aman	Kurang Aman
Chiper Teks	Simetris Block Chiper	Simetris Block Chiper
Pengukuran Waktu	Cepat	Lambat

IV.6. Kelebihan Dan Kekurangan

IV.6.1. Kelebihan

Adapun kelebihan dari aplikasi adalah sebagai berikut :

1. Aplikasi ini memberikan perbandingan dua metode dalam proses enkripsi dan dekripsi data teks.
2. Aplikasi ini juga memaparkan konsep kerja pada setiap metode yang digunakan di akhir proses enkripsi dan dekripsi.
3. Mudah digunakan karena *user interface* yang sederhana

IV.6.2. Kekurangan

Adapun kekurangan dari aplikasi adalah sebagai berikut :

a. Melihat perkembangan sistem pada aplikasi dengan desain yang dinamis, aplikasi ini masih perlu banyak pengembangan lagi dari segi desain dan tampilan untuk mempermudah user dalam memahami proses jalannya aplikasi.
b. Aplikasi ini dirancang hanya untuk membandingkan dua metode yakni algoritma AES128 dan algoritma DES.
c. Ketika mengenkripsi data text dengan jumlah karakter yang terlalu panjang maka akan memakan waktu yang cukup lama.