BAB IV

HASIL DAN UJI COBA

IV.1. Uji Coba

Proses uji coba dari aplikasi ini adalah dengan melakukan pengujian langsung terhadap *file* yang akan dienkripsi maupun didekripsi. Adapun *file* yang akan diproses oleh aplikasi ini adalah *file* yang berekstensi .txt.

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

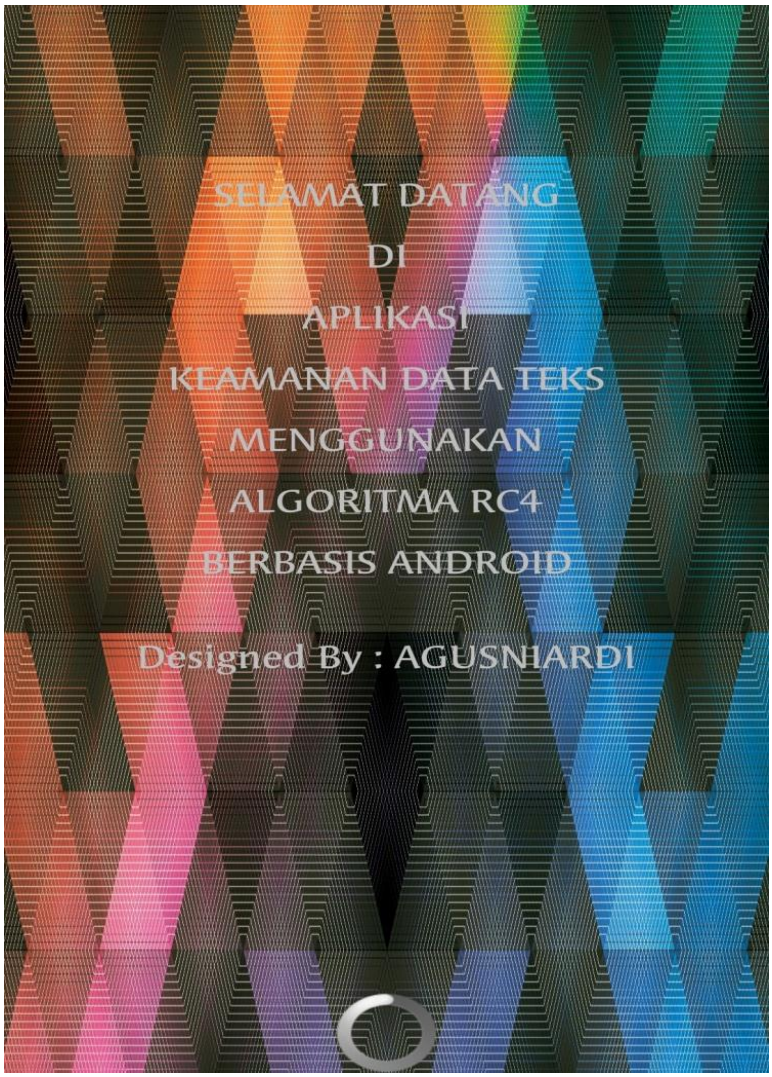
1. Melakukan instalasi *eclipse*.
2. Membuka aplikasi yang telah selesai dibuat dengan cara mengimpor kedalam *workspaceeclipse*.
3. Setelah proses *import* selesai dengan benar, langkah selanjutnya memastikan bahwa tidak ada yang *error* pada program tersebut.
4. Hubungkan *handphone android* ke laptop atau komputer yang sudah terdapat program yang telah dibuat dengan menggunakan kabel USB.
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. *InterfaceSplash*

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



Gambar IV.1 *Interface Splash*

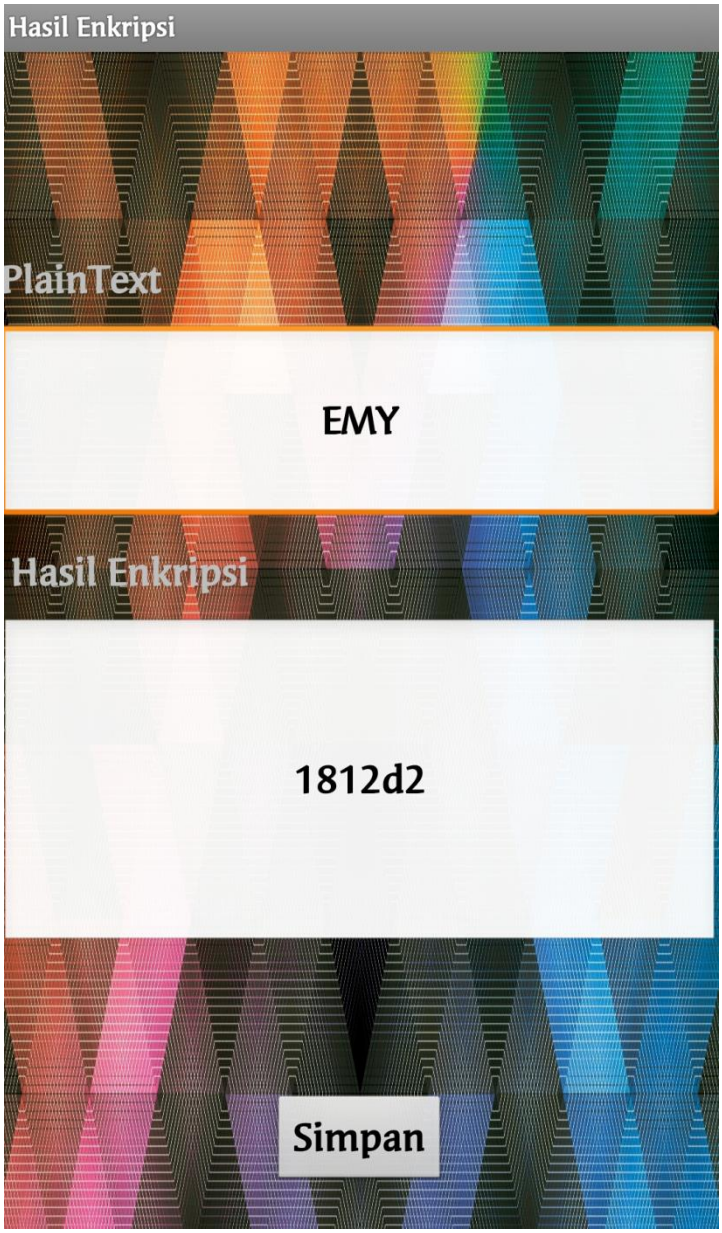
2. *Interface* Menu

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*Interface* Menu

3. *Interface Form* Hasil Enkripsi

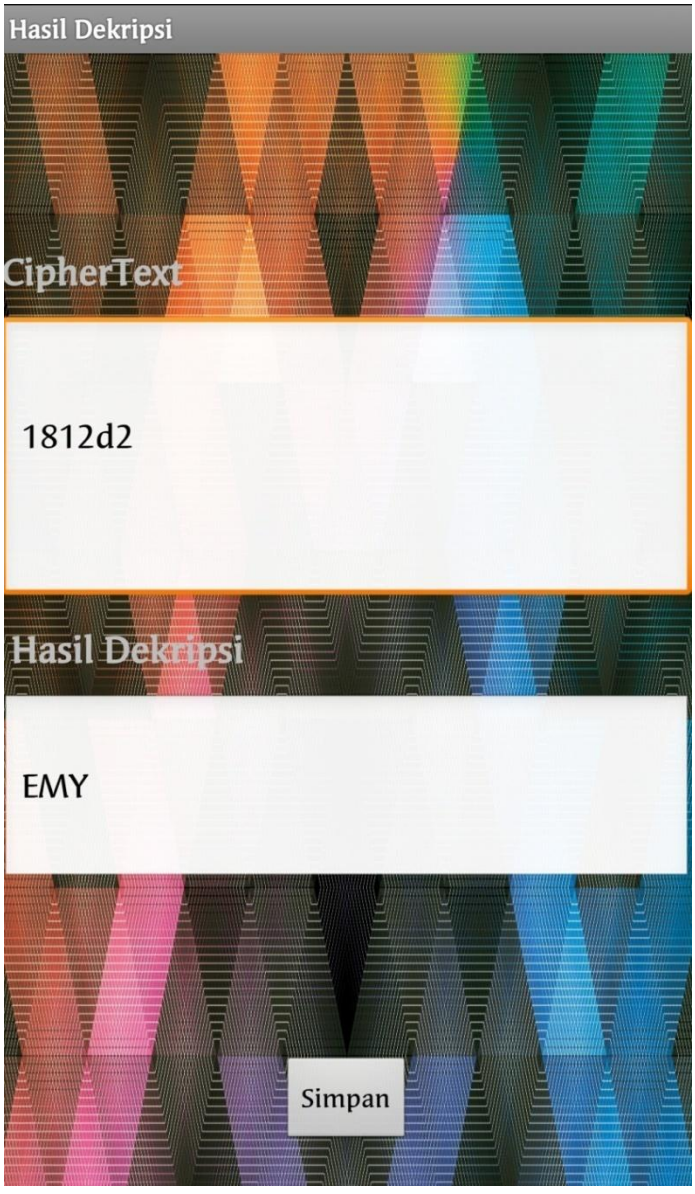
Interface ini merupakan *form* setelah proses enkripsi *file* berjalan lancar. Adapun *interface form* hasil enkripsi dapat dilihat pada gambar IV.3 berikut ini.



Gambar IV.3*Interface Form* Hasil Enkripsi

4. *Interface Form* Hasil Dekripsi

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



Gambar IV.4*Interface Form* Hasil Dekripsi

5. *Interface Form* Tentang

Pada *interface form* Tentang, terdapat keterangan dari aplikasi yang dibuat oleh perancang. Adapun *interface form* Tentang dapat dilihat pada gambar IV.5 berikut ini.

Gambar IV.5*Interface Form* Tentang

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 min. 2Gb
3. *Keyboard, Mouse. USB Cable*
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*

IV.4. Analisa Hasil

IV.4.1. Uji Coba Perhitungan

Berikut ini adalah contoh uji coba perhitungan enkripsi dan dekripsi *file*teks menggunakan algoritma RC4 dengan *plaintext* "EMY" dan kunci "2828282828282828"

a. Inisialisasi *array* S-*Box*dengan panjang 16 *byte*sehingga S-*Box array* S berbentuk S[0] = 0, S[1] = 1, S[2] = 2, S[3] = 3, S[4] = 4, S[5] = 5, S[6] = 6, S[7] = 7, S[8] = 8, S[9] = 9, S[10] = 10, S[11] = 11, S[12] = 12, S[13] = 13, S[14] = 14, S[15] = 15.

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

b. Inisialisasi *array* kunci (S-*Box* lain) sehingga S-*Boxarray* kunci (K) berbentuk K[0] = 2, K[1] = 8, K[2] = 2, K[3] = 8, K[4] = 2, K[5] = 8,K[6] = 2, K[7] = 8, K[8] = 2, K[9] = 8, K[10] = 2, K[11] = 8, K[12] = 2, K[13] = 8, K[14] = 2, K[15] = 8.

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

c. Lakukan permutasi terhadap nilai-nilai di dalam *array* S dengan cara menukar isi *array* S[i] dengan S[j], karena pada contoh digunakan algoritma RC4 dengan *mode* 16 *byte*, maka prosesnya adalah sebagai berikut.

j = 0

for i = 0 to 15

j = (j + S[i] + K[i]) mod 16

isi S[i] dan S[j] ditukar

Dengan menggunakan algoritma tersebut, untuk nilai i = 0 sampai i = 15 didapatkan nilai *array* S sebagai berikut.

1. Iterasi pertama, untuk nilai i = 0

j = (j + S[i] + K[i]) mod 16

j = (j + S[0] + K[0]) mod 16

j = (0 + 0 + 2) mod 16

j = 2

Dilakukan penukaran isi *array* S[0] dan S[2] sehingga *array* S berbentuk:

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

2. Iterasi kedua, untuk nilai i = 1 dan j = 2 (nilai j = 2 didapatkan dari iterasi pertama)

   j = (j + S[i] + K[i]) mod 16

   j = (j + S[1] + K[1]) mod 16

   j = (2 + 1 + 8) mod 16

   j = 11

   Dilakukan penukaran isi *array* S[1] dan S[11] sehingga *array* S berbentuk:

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

3. Iterasi ketiga, untuk nilai i = 2 dan j = 11 (nilai j = 11 didapatkan dari iterasi kedua)

   j = (j + S[i] + K[i]) mod 16

   j = (j + S[2] + K[2]) mod 16

   j = (11 + 0 + 2) mod 16

   j = 13

   Dilakukan penukaran isi *array* S[2] dan S[13] sehingga *array* S berbentuk:

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

4. Iterasi keempat, untuk nilai i = 3 dan j = 13 (nilai j = 13 didapatkan dari iterasi ketiga)

   j = (j + S[i] + K[i]) mod 16

   j = (j + S[3] + K[3]) mod 16

   j = (13 + 3 + 8) mod 16

   j = 8

   Dilakukan penukaran isi *array* S[3] dan S[8] sehingga *array* S berbentuk:

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

5. Iterasi kelima, untuk nilai i = 4 dan j = 8 (nilai j = 8 didapatkan dari iterasi keempat)

   j = (j + S[i] + K[i]) mod 16

   j = (j + S[4] + K[4]) mod 16

   j = (8 + 4 + 2) mod 16

   j = 14

   Dilakukan penukaran isi *array* S[4] dan S[14] sehingga *array* S berbentuk:

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

6. Iterasi keenam, untuk nilai i = 5 dan j = 14 (nilai j = 14 didapatkan dari iterasi kelima)

   j = (j + S[i] + K[i]) mod 16

j = (j + S[5] + K[5]) mod 16

j = (14 + 5 + 8) mod 16

j = 11

Dilakukan penukaran isi *array* S[5] dan S[11] sehingga *array* S berbentuk:

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

7. Iterasi ketujuh, untuk nilai i = 6 dan j = 11 (nilai j = 11 didapatkan dari iterasi keenam)

   j = (j + S[i] + K[i]) mod 16

   j = (j + S[6] + K[6]) mod 16

   j = (11 + 6 + 2) mod 16

   j = 3

   Dilakukan penukaran isi *array* S[6] dan S[3] sehingga *array* S berbentuk:

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

8. Iterasi kedelapan, untuk nilai i = 7 dan j = 3 (nilai j = 3 didapatkan dari iterasi ketujuh)

   j = (j + S[i] + K[i]) mod 16

   j = (j + S[7] + K[7]) mod 16

   j = (3 + 7 + 8) mod 16

   j = 2

Dilakukan penukaran isi *array* S[7] dan S[2] sehingga *array* S berbentuk:

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

9. Iterasi kesembilan, untuk nilai i = 8 dan j = 2 (nilai j = 2 didapatkan dari iterasi kedelapan)

   j = (j + S[i] + K[i]) mod 16

   j = (j + S[8] + K[8]) mod 16

   j = (2 + 3 + 2) mod 16

   j = 7

   Dilakukan penukaran isi *array* S[8] dan S[7] sehingga *array* S berbentuk:

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

10. Iterasi kesepuluh, untuk nilai i = 9 dan j = 7 (nilai j = 7 didapatkan dari iterasi kesembilan)

    j = (j + S[i] + K[i]) mod 16

    j = (j + S[9] + K[9]) mod 16

    j = (7 + 9 + 8) mod 16

    j = 8

    Dilakukan penukaran isi *array* S[9] dan S[8] sehingga *array* S berbentuk:

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

11. Iterasi kesebelas, untuk nilai i = 10 dan j = 8 (nilai j = 8 didapatkan dari iterasi kesepuluh)

    j = (j + S[i] + K[i]) mod 16

    j = (j + S[10] + K[10]) mod 16

    j = (8 + 10 + 2) mod 16

    j = 4

    Dilakukan penukaran isi *array* S[10] dan S[4] sehingga *array* S berbentuk:

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

12. Iterasi keduabelas, untuk nilai i = 11 dan j = 4 (nilai j = 4 didapatkan dari iterasi kesebelas)

    j = (j + S[i] + K[i]) mod 16

    j = (j + S[11] + K[11]) mod 16

    j = (4 + 1 + 8) mod 16

    j = 13

    Dilakukan penukaran isi *array*S[11] dan S[13] sehingga *array* S berbentuk:

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

13. Iterasi ketigabelas, untuk nilai i = 12 dan j = 13 (nilai j = 13 didapatkan dari iterasi keduabelas)

j = (j + S[i] + K[i]) mod 16

j = (j + S[12] + K[12]) mod 16

j = (13 + 1 + 2) mod 16

j = 0

Dilakukan penukaran isi *array* S[13] dan S[0] sehingga *array* S berbentuk:

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

14. Iterasi keempatbelas, untuk nilai i = 13 dan j = 0 (nilai j = 0 didapatkan dari iterasi ketigabelas)

    j = (j + S[i] + K[i]) mod 16

    j = (j + S[13] + K[13]) mod 16

    j = (0 + 1 + 8) mod 16

    j = 9

    Dilakukan penukaran isi *array* S[13] dan S[9] sehingga *array* S berbentuk:

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

15. Iterasi kelimabelas, untuk nilai i = 14 dan j = 9 (nilai j = 9 didapatkan dari iterasi keempatbelas)

    j = (j + S[i] + K[i]) mod 16

    j = (j + S[14] + K[14]) mod 16

    j = (9 + 4 + 2) mod 16

j = 15

Dilakukan penukaran isi *array* S[14] dan S[15] sehingga *array* S berbentuk:

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

16. Iterasi keenambelas, untuk nilai i = 15 dan j = 15 (nilai j = 15 didapatkan dari iterasi kelimabelas)

j = (j + S[i] + K[i]) mod 16

j = (j + S[15] + K[15]) mod 16

j = (15 + 4 + 8) mod 16

j = 11

Dilakukan penukaran isi *array* S[15] dan S[11] sehingga *array* S berbentuk:

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

Karena*plaintext*mempunyai panjang 3 *byte*, untuk mendapatkan *ciphertext*terlebih dahulu bangkitkan *keystream* sebanyak 3 *byte*. Proses untuk membangkitkan kunci enkripsinya adalah sebagai berikut.

i = j = 0

i = (i + 1) mod 16

j = (j + S[i]) mod 16

isi S[i] dan S[j] ditukar

t = (S[i] + S[j]) mod 16

K = S[t]

Dengan menggunakan algoritma tersebut, kita bisa membangkitkan kunci untuk proses enkripsi dengan cara berikut.

1. Itersai pertama, untuk nilai i = j = 0, maka:

   i = (i + 1) mod 16

   i = (0 + 1) mod 16

   i = 1

   j= (j + S[i]) mod 16

   j= (0 + S[1]) mod 16

   j= (0 + 0) mod 16

   j = 0

   Menukar isi S[1] dan S[0] sehingga *array* S berbentuk.

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

   t = (S[1] + S[0]) mod 16

   t = (1 + 11) mod 16

   t = 12

   K = S[t] = S[12] = 12

   Jadi, kunci pertama untuk enkripsi adalah 12.

2. Itersai kedua, untuk nilai i = 1 dan j = 0 (nilai i dan j didapat dari hasil iterasi pertama), maka:

   i = (i + 1) mod 16

   i = (1 + 1) mod 16

   i = 2

   j= (j + S[i]) mod 16

   j= (0 + S[2]) mod 16

   j= (0 + 7) mod 16

   j = 7

   Menukar isi S[2] dan S[7] sehingga *array* S berbentuk.

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

   t = (S[2] + S[7]) mod 16

   t = (3 + 7) mod 16

   t = 10

   K = S[t] = S[4] = 10

   Jadi, kunci kedua untuk enkripsi adalah 10.

3. Itersai ketiga, untuk nilai i = 2 dan j = 7 (nilai i dan j didapat dari hasil iterasi kedua), maka:

   i = (i + 1) mod 16

   i = (2 + 1) mod 16

   i = 3

j= (j + S[i]) mod 16

j= (7 + S[3]) mod 16

j= (7 + 6) mod 16

j = 13

Menukar isi S[3] dan S[13] sehingga *array* S berbentuk.

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

t = (S[3] + S[13]) mod 16

t = (5 + 6) mod 16

t = 11

K = S[t] = S[0] = 11

Jadi, kunci ketiga untuk enkripsi adalah 11.

4. Itersai keempat, untuk nilai i = 3 dan j = 13 (nilai i dan j didapat dari hasil iterasi ketiga), maka:

i = (i + 1) mod 16

i = (3 + 1) mod 16

i = 4

j= (j + S[i]) mod 16

j= (13 + S[4]) mod 16

j= (13 + 10) mod 16

j = 7

Menukar isi S[4] dan S[7] sehingga *array* S berbentuk.

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

t = (S[4] + S[7]) mod 16

t = (7 + 10) mod 16

t = 1

K = S[t] = S[1] = 1

Jadi, kunci keempat untuk enkripsi adalah 1.

5. Itersai kelima, untuk nilai i = 4 dan j = 7 (nilai i dan j didapat dari hasil iterasi keempat), maka:

   i = (i + 1) mod 16

   i = (4 + 1) mod 16

   i = 5

   j= (j + S[i]) mod 16

   j= (7 + S[5]) mod 16

   j= (7 + 13) mod 16

   j = 4

   Menukar isi S[5] dan S[4] sehingga *array* S berbentuk.

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

t = (S[5] + S[4]) mod 16

t = (7 + 4) mod 16

t = 11

K = S[t] = S[11] = 11

Jadi, kunci keempat untuk enkripsi adalah 4.

Proses enkripsinya : *plaintext* di-XOR-kan dengan kunci

Untuk *plaintext* "EMY" dan kunci "2828282828282828" dari hasil pembangkitan kunci *ciphertext*-nya adalah:

	Kode Biner					
	E	M	Y			
Plaintext (P)	01000101	01001101	01011001	00000000	00000000	00000000
Kunci (K)	01000100	01000101	01011000	00000010	01100100	00000010
P -xor- K	00000001	00001000	00000001	00000010	01100100	00000010
Ciphertext(C)	1	8	1	2	D	2

Cara mendapatkan *plaintext*-nya adalah:

	Kode Biner					
	1	8	1	2	D	2
Ciphertext(C)	00000001	00001000	00000001	00000010	01100100	00000010
Kunci (K)	01000100	01000101	01011000	00000010	01100100	00000010
C -xor- K	01000101	01001101	01011001	00000000	00000000	00000000
Plaintext (P)	E	M	Y			

IV.4.2. Hasil Uji Coba

Berikut ini adalah hasil pengujian *black box* aplikasi keamanan data teks menggunakan algoritma RC4.

Tabel IV.1 Hasil Pengujian *Black Box* Aplikasi Keamanan Data Teks Menggunakan algoritma RC4

No	Form	Keterangan	Hasil
1	Masuk ke aplikasi keamanan data teks menggunakan algoritma RC4	Sistem akan membuka aplikasi menampilkan layar pembuka aplikasi dan menuju menu pilihan	Valid
2	Memilih menu Enkripsi atau Dekripsi	Akan tampil menu untuk memilihdata teks yang akan dienkrip atau didekrip	Valid
3	Klik *file* teks yang dipilih untuk dienkrip atau didekrip	Menampilkan isi data teks dan perintah untuk memasukkan kunci sebanyak 16 karakter	Valid
4	Lanjut ke menu pilihan untuk menyimpan data teks hasil dari enkripsi/dekripsi	Menampilkan isi data teks/plainteks dan hasilnya setelah dienkripsi/dekripsi	Valid
5	Memilih menu "Tentang" untuk melihat rincian aplikasi	Akan tampil menu "Tentang"	Valid

IV.4.3. Kelebihan Dan Kekurangan

a) Kelebihan aplikasi yang dirancang adalah sebagai berikut :

1. Aplikasi dapat menjaga keamanan dan kerahasiaan *file* dari orang yang tidak bertanggung jawab.
2. Aplikasi ini bekerja dengan menggunakan kunci yang sama antara proses enkripsi dan dekripsinya sehingga mudah diingat.
3. Mempermudah *user* dalam mengamankan data.
4. Mudah digunakan karena *user interface* yang sederhana

b) Kekurangan aplikasi yang dirancang adalah sebagai berikut :

1. Tampilan dan *layout* dari aplikasi masih sederhana.
2. Proses enkripsi dan dekripsi hanya bisa dilakukan pada *file* teks yang ekstensinya txt.
3. Ketika mengenkripsi *file* teks dengan jumlah karakter yang terlalu panjang pada *file* akan memakan waktu yang cukup lama.
4. Proses enkripsi dan dekripsi hanya dapat dilakukan pada memori internal.