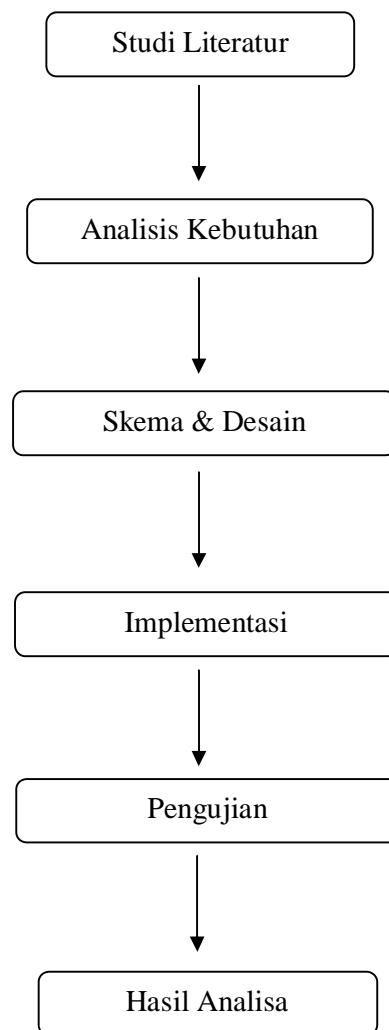


BAB III

METODE PENELITIAN

Dalam pengerjaan Tugas Akhir ini diperlukan langkah-langkah kegiatan penelitian untuk mendapatkan hasil yang maksimal. Untuk itu penulis merencanakan suatu langkah-langkah yang dapat memaksimalkan dalam pengerjaan Tugas Akhir ini. Langkah-langkah itu adalah sebagai berikut :



Gambar 10 Tahapan Penelitian

3.1 Studi Literatur

Studi literatur adalah mencari referensi teori yang relevan dengan kasus atau permasalahan yang ditemukan. Referensi tersebut berisikan tentang :

- a. Kriptografi
- b. DES
- c. 3DES
- d. VB.Net

Referensi ini dapat dicari dari buku, jurnal, artikel laporan penelitian, situs-situs di internet dan tutorial. Output dari studi literatur ini adalah terkoleksinya referensi yang relevan dengan perumusan masalah.

3.2 Analisis Kebutuhan dan spesifikasi perangkat

Menurut (Munir, 2006) Pertukaran informasi setiap detik di internet membuat pihak tidak bertanggung jawab menyalahgunakan informasi tersebut. Oleh karena itu agar data yang dikirim aman dari orang yang tidak bertanggung jawab, data tersebut harus disembunyikan dengan cara menyandikan data tersebut menggunakan kriptografi algoritma 3DES dengan metode penyandian EEE. Pertukaran data baik di jaringan lokal maupun di jaringan internet membawa informasi berupa pesan (*message*) yaitu suatu data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan disebut juga plainteks (*plaintext*). Pesan dapat berupa data atau informasi yang dikirimkan atau disimpan di dalam media perekaman. Pesan yang tersimpan tidak hanya berupa teks, tetapi dapat juga berbentuk gambar (*image*), suara (*audio*), dan juga video.

Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke bentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut *chipertext*. *Chipertext* harus dapat ditransformasikan kembali menjadi plainteks agar pesan yang diterima bisa dibaca. Dalam kriptografi, *Triple DES* adalah sebuah *cipher* blok yang dibentuk oleh DES dengan menggunakannya tiga kali. Ketika diketahui bahwa kunci berukuran 56 bit dari DES tidak cukup kuat untuk menjaga dari *brute force attacks*, *Triple DES* dengan tiga kunci berbeda memiliki kunci berukuran 168bit (3 kali kunci 56bit dari DES). Penggunaan tiga kunci tersebut penting untuk mencegah *meet-in-the-middle attacks*.

3.2.1 Spesifikasi Perangkat Keras

3.2.1.1 Personal Computer (PC) atau laptop

Pada penelitian ini dibutuhkan beberapa komputer yang digunakan sebagai server dan client, server yang nantinya bertugas sebagai server yang mengatur mangle dan client sebagai penyerang/ attacker. Berikut tabel kebutuhan *personal computer*:

No	Faktor	Deskripsi
1	Prosesor	Intel® Core™ i3-2310M CPU @ 2.10GHz
2	RAM	4 GB DDR3 / 4096 MB RAM
3	HDD	500

Tabel 1 Spesifikasi Laptop

3.2.2 Spesifikasi Perangkat Lunak

3.2.2.1 Microsoft .Net Framework

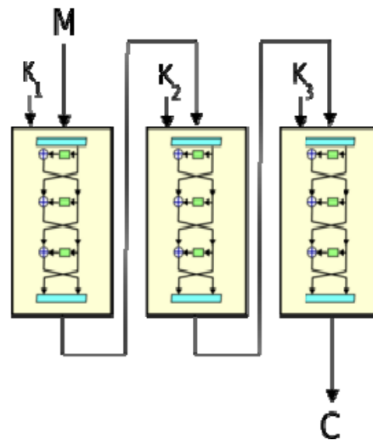
Microsoft .NET Framework (dibaca *Microsoft Dot Net Framework*) atau lebih dikenal dengan singkatan dot net (tidak berhubungan dengan domain .net) merupakan sebuah perangkat lunak kerangka kerja yang berjalan utamanya pada sistem operasi Microsoft Windows, saat ini .NET Framework umumnya telah terintegrasi dalam distribusi standar Windows (mulai dari Windows Server 2003 dan versi-versi Windows yang lebih baru). Kerangka kerja ini menyediakan sejumlah besar pustaka pemrograman komputer dan mendukung beberapa bahasa pemrograman serta interoperabilitas yang baik sehingga memungkinkan bahasa-bahasa tersebut berfungsi satu dengan lain dalam pengembangan sistem. Berbeda halnya dengan tipikal aplikasi konvensional umumnya, program yang ditulis dengan memanfaatkan .NET Framework berjalan pada lingkungan perangkat lunak melalui Common Language Runtime, dan bukan perangkat keras secara langsung. Hal ini memungkinkan aplikasi yang dibuat di atas .NET secara teoretis dapat berjalan pada perangkat keras apapun yang didukung oleh .NET Framework. Perangkat lunak ini adalah kunci penawaran utama dari Microsoft, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk platform Windows.

3.2.2.2 VB.Net

adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET Framework, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para *programmer* dapat membangun aplikasi Windows Forms, Aplikasi web berbasis ASP.NET, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti Microsoft Visual C++, Visual C#, atau Visual J#), atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio .NET. Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas .NET Framework. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft, dan versi baru ini tidak kompatibel dengan versi terdahulu.

3.3 Skema & Desain

3DES memiliki tiga buah kunci yang berukuran 168-bit (tiga kali kunci 56-bit dari DES). Pada algoritma 3DES dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma DES. Berikut ini adalah skema global 3DES.



Gambar 11 Algoritma 3DES

Ada dua pemilihan kunci eksternal algoritma 3DES yaitu :

- a. K_1, K_2, K_3 adalah kunci yang saling berbeda

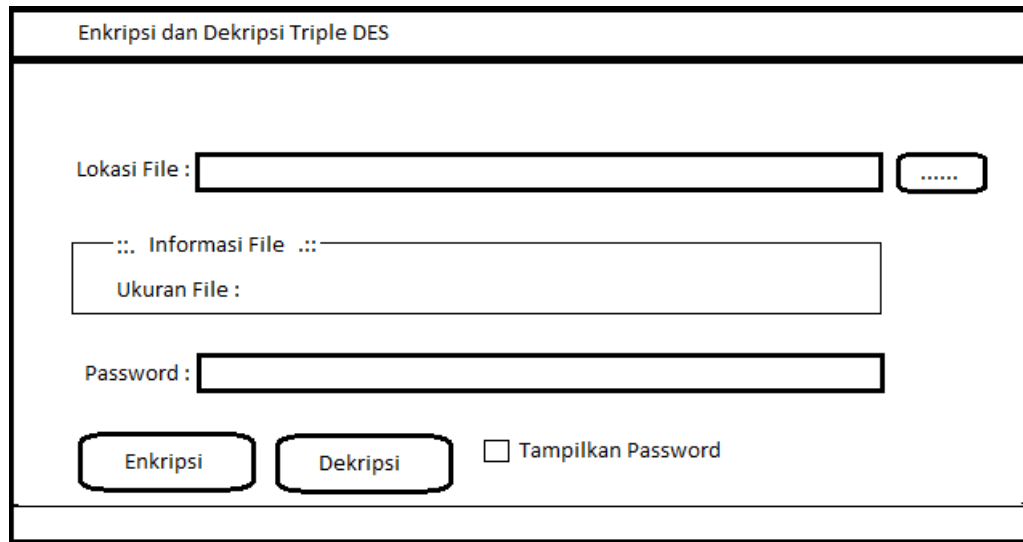
$$K_1 \neq K_2 \neq K_3 \neq K_1$$

- b. K_1 dan K_2 adalah kunci yang berbeda, dan k_3 sama dengan K_1

$$K_1 \neq K_2 \text{ dan } K_3 = K_1$$

3.3.1 Desain *Interfaces* Aplikasi 3DES

Desain aplikasi untuk enkripsi dan dekripsi pada algoritma 3DES yang akan dibuat seperti berikut ini :



Gambar 3.3.1 Rancangan dari tampilan *interfaces*

Dibawah ini merupakan keterangan dari gambar 3.3.1 diatas, yaitu :

1. *Frame* atas, merupakan judul atau nama rancangan yang akan dibuat
2. *Button* ... untuk mencari file doc yang akan di enkripsi ataupun dekripsi
3. *Password text field* merupakan text password yang akan digunakan untuk melakukan enkripsi dan dekripsi
4. *Button* enkripsi digunakan untuk meng-enkripsi file doc
5. *Button* dekripsi digunakan untuk meng-dekripsi file doc
6. *Checklist* tampilkan *password* untuk menampilkan password yang kita ketik di *password text field*

3.4 Implementasi

Dalam fase ini, rancangan pada fase perancangan digunakan untuk melakukan proses enkripsi dan dekripsi pada file .doc dengan menggunakan aplikasi yang sudah dibuat. Aktivitas yang dilakukan pada tahap ini diantaranya adalah melakukan enkripsi dan dekripsi pada file .doc.

3.5 Pengujian

Pengujian dilakukan dengan meng-enkripsi file doc dan dekripsi file doc dengan menggunakan metode 3DES.

3.6 Hasil Analisa

Hasil analisa adalah adanya sebuah pengujian enkripsi file doc dan dekripsi file doc dengan menggunakan aplikasi yang sudah disediakan.

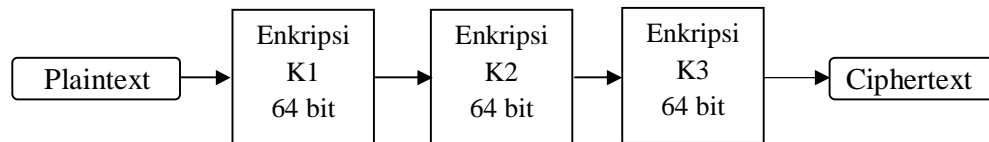
3.7 Pembahasan Metode 3 DES

Metode 3 DES menggunakan DES sebanyak 3 kali. Bentuk sederhana perhitungan untuk enkripsi dan dekripsi 3 DES adalah :

Enkripsi : $C = E_{K3}(E_{K2}(E_{K1}(P)))$

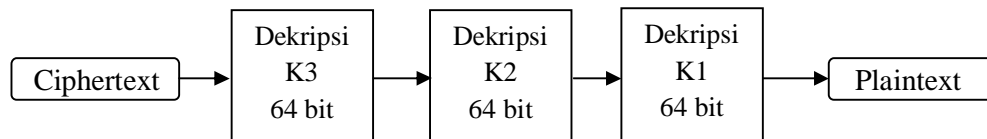
Dekripsi : $P = D_{K1}(D_{K2}(D_{K3}(C)))$

Bentuk ini dikenal dengan mode EEE karena untuk memperoleh chiperteks dilakukan proses enkripsi sebanyak 3 kali, seperti yang digambarkan dalam gambar skema berikut ini.



Gambar 13 Enkripsi EEE 3DES

Sedangkan skema untuk dekripsinya digambarkan pada gambar 3.7.2 .



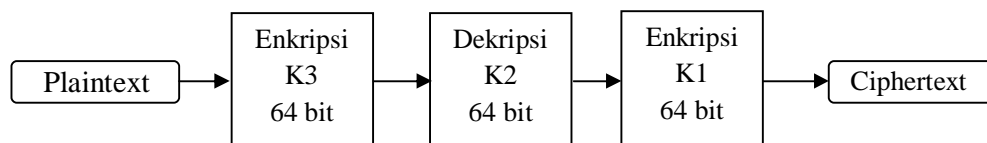
Gambar 14 Dekripsi EEE 3DES

Bentuk perhitungan 3DES lainnya yang menggunakan 3 buah kunci yang berbeda adalah :

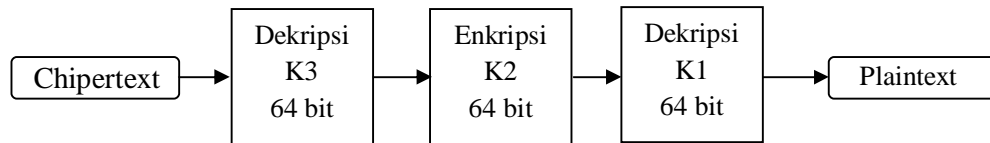
Enkripsi : $C = E_{K3}(D_{K2}(E_{K1}(P)))$

Dekripsi : $P = D_{K1}(E_{K2}(D_{K3}(C)))$

Untuk skema enkripsi dan dekripsinya terlihat seperti gambar dibawah ini :



Gambar 15 Enkripsi EDE 3 DES



Gambar 16 Dekripsi EDE 3 DES

Selain menggunakan 3 kunci, metode 3 DES juga dapat dibuat varian lainnya dengan menggunakan hanya 2 kunci. Proses perhitungannya yaitu sebagai berikut :

$$\text{Enkripsi : } C = E_{K1}(D_{K2}(E_{K1}(P)))$$

$$\text{Dekripsi : } P = D_{K1}(E_{K2}(D_{K1}(C)))$$

3.7.1 Proses 3 DES

Pada proses enkripsi pada algoritma 3DES ini menggunakan rumus sebagai berikut.

$$\text{Enkripsi : } C = E_{K3}(E_{K2}(E_{K1}(P)))$$

$$\text{Dekripsi : } P = D_{K1}(D_{K2}(D_{K3}(C)))$$

Rumus diatas menggunakan 3 kunci yang berbeda dan dilakukan 3x proses enkripsi. Seperti studi kasus dibawah ini yang akan menjelaskan bagaimana proses enkripsi pada algoritma 3DES.

Plaintext(x) : computer (636f6d7075746572)

Key (k1) : belajarr (62656c616a617272)

Key (k2) : enkripsi (656e6b7269707369)

Key (k3) : dekripsi (64656b7269707369)

Setelah menentukan kunci yang akan digunakan untuk enkripsi maka akan dilakukan conversi ke biner

Plaintext ke biner :

computer : 01100011 01101111 01101101 01110000 01110101 01110100 01100101

*Key*1 ke biner :

belajarr : 01100010 01100101 01101100 01100001 01101010 01100001 01110010
01110010

Key2 ke biner :

enkripsi : 01100101 01101110 01101011 01110010 01101001 01110000 01110011
01101001

Key3 ke biner :

dekripsi : 01100100 01100101 01101011 01110010 01101001 01110000 01110011
01101001

selanjutnya melakukan inisial permutasi (IP) pada bit *plaintext* menggunakan tabel IP berikut :

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 2 Inisial Permutasi (IP)

Urutan pada bit *plaintext* urutan ke 58 diratuh pada posisi 1, urutan pada bit *plaintext* urutan ke 50 ditaruh pada posisi 2, urutan pada bit *plaintext* urutan ke 42 ditaruh pada posisi 3,dst.

Sehingga hasil output adalah

Ip(x) : 11111111 10111000 01110110 01010111 00000000 11111111 00000110
10000011

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

L₀ : 11111111 10111000 01110110 01010111 (tabel berwarna hijau)

R₀ : 00000000 11111111 00000110 10000011 (tabel berwarna biru)

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

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	46	37	29
21	13	5	28	20	12	4

Tabel 3 PC-1

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

CD(k) adalah kunci 1 yang telah di permutasi

CD(k) : 00000000 11111111 11111111 1100110 0001000 0110000 0100000

Pecah CD(k) menjadi 2 bagian kiri dan kanan, sehingga menjadi

C_0 : 00000000 11111111 11111111 1100110 (tabel PC-1 warna hijau)

D_0 : 1100110 0001000 0110000 0100000 (tabel PC-1 warna biru)

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

Putaran ke - i	Jumlah Pergeseran(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

Tabel 4 tabel *Left Shift*

Untuk putaran ke 1, dilakuakn pergeseran 1 bit ke kiri, putaran ke 2, dilakukan pergeseran 1bit kekiri dan putaran ke 3, dilakukan pergeseran 2 bit kekiri, dst.

Berikut hasil output *left shift* ;

C_0 : 0000000 1111111 1111111 1100110

D_0 : 1100110 0001000 0110000 0100000

Digeser 1 bit ke kiri

C_1 : 0000001 1111111 1111111 1001100

D_1 : 1001100 0010000 1100000 1000001

Digeser 1 bit ke kiri

C_2 : 0000011 1111111 1111111 0011000

D_2 : 0011000 0100001 1000001 0000011

.....

Digeser 2 bit ke kiri

C_{14} : 1100000 0001111 1111111 1111100

D_{14} : 0001100 1100001 0000110 0000100

Digeser 2 bit ke kiri

C_{15} : 0000000 0111111 1111111 1110011

D_{15} : 0110011 0000100 0011000 0010000

Setiap hasil putaran digabungkan kembali menjadi C_iD_i dan diinput kedalam tabel Permutation Compression 2 (PC-2) dan terjadi kompresi data C_iD_i 56 bit menjadi C_iD_i 48 bit.

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

Tabel 5 PC-2

C_1D_1 : 00000001 1111111 11111111 11000101 000100000 11000010 10000001

K_1 : 11110000 10111110 01100110 10101000 00000101 00010010

C_2D_2 : 00000011 11111111 11111111 00000100 01000001 10000101 00000011

K_2 : 11110000 10111110 01110110 10000111 00010001 10100000

.....

$C_{15}D_{15}$: 00000000 01111111 11111111 11100110 10001000 00110000 10100000

K_{15} : 111100 011011 111000 101110 011000 110100 000010 100000

$C_{16}D_{16}$: 00000000 11111111 11111111 11001101 00010000 01100001 01000000

K_{16} : 11110000 10111110 10100110 00000010 00010000 00011111

Pada langkah berikutnya, kita akan meng-ekspansi data R_{i-1} 32 bit menjadi R_i 48 bit sebanyak 16 kali putaran dengan nilai perputaran $1 \leq i \leq 16$ 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

Tabel 6 Ekspansi (E)

Hasil $E(R_{i-1})$ kemudian di XOR dengan K_i dan menghasilkan Vektor Matriks A_i .

Berikut hasil outputnya :

Iterasi 1

$E(R(1)-1) = 10010101\ 01010111\ 10100100\ 00001111\ 01111100\ 10101110$

$K1 = 11110000\ 10111110\ 01100110\ 10101000\ 00000101\ 00010010$

----- XOR

$A1 = 01110000\ 10101001\ 10011000\ 00101000\ 11010001\ 00010100$

Iterasi 2

$E(R(2)-1) = 10010101\ 01010111\ 10100100\ 00001111\ 01111100\ 10101110$

$K2 = 11110000\ 10111110\ 01110110\ 10000111\ 00010001\ 10100000$

----- XOR

$A2 = 01100101\ 11101001\ 11010010\ 10001000\ 01101101\ 00001110$

.....

Iterasi 15

$E(R(15)-1) = 01101111\ 00111111\ 11110010\ 10101111\ 10111101\ 00000001$

$K15 = 111100\ 011011\ 111000\ 101110\ 011000\ 110100\ 000010\ 100000$

----- XOR

$A15 = 10011111\ 10000001\ 01011100\ 10011101\ 00010001\ 00000001$

Iterasi 16

$E(R(16)-1) = 01111010\ 10000011\ 00001110\ 10011010\ 01101010\ 01010101$

$K16 = 11110000\ 10111110\ 10100110\ 00000010\ 00010000\ 00011111$

----- XOR
A16 = 10001010 00111101 10101000 10011000 01111010 01001010

Langkah selanjutnya 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 vektor B_i 32 bit.

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

Tabel 7 S-BOX 1

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

Tabel 8 S-BOX 2

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

Tabel 9 S-BOX 3

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	8	9	4	5	11	12	7	2	14

Tabel 10 S-BOX 4

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

Tabel 11 S-BOX 5

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

Tabel 12 S-BOX 6

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

Tabel 13 S-BOX 7

	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

Tabel 14 S-BOX 8

Setiap angka pada tabel S-BOX convert menjadi biner, hasil convert seperti pada tabel berikut :

	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	1010	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0001	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

Tabel 15 S-BOX 1 biner

Menghitung A_1 dengan S-BOX dengan cara :

A_1 : 01110000 10101001 10011000 00101000 11010001 00010100

Ambil 6 bit dari A_1 yaitu 011100 kemudian pisahkan blok menjadi 2 yaitu :

- Bit pertama dan terakhir 0 dan 0 digabungkan 00
- Bit kedua hingga lima 1110

Kemudian bandingkan dengan memeriksa perpotongan antara keduanya didapatkan nilai 0000 dan seterusnya untuk blok kedua hingga ke enam kita bandingkan dengan S-BOX 2 hingga S-BOX 8.

Berdasarkan cara diatas diperoleh hasil sebagai berikut :

B_1 : 00001011 10011011 10101001 00100011

B_2 : 10011010 00000010 00101111 01100001

B_3 : 01100010 11000101 11000011 01111100

B_4 : 01110001 11011100 10111011 10000101

B_5 : 11100001 01100100 10101110 01001011

B_6 : 01101000 01110111 10111101 01111000

B_7 : 11110011 10100000 11000101 01110100

B_8 : 00110001 10001011 01011110 00101101

B_9 : 11111011 01001000 00010100 01010110

B_{10} : 01011000 01100001 00100100 11101011

B_{11} : 10101110 11000100 01110100 01011101

B_{12} : 11110001 00110011 10110110 01000001

B_{13} : 10001010 11110000 11010011 10110000

B_{14} : 00101000 11110101 00101101 10010100

B_{15} : 00101001 00000100 01110110 00100001

B_{16} : 00011000 11001100 10110010 00011111

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

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

Tabel 16 P-BOX

Sehingga hasil yang diperoleh seperti berikut ini :

$P(B_1)$: 11010101 01001010 011011011 1000000

$P(B_2)$: 01010000 11111000 00101100 10001010

$P(B_3)$: 11001011 00110101 10110111 00100000

$P(B_4) : 00110101\ 00100001\ 11111011\ 11100011$

$P(B_5) : 00011001\ 10110011\ 11011010\ 10001100$

$P(B_6) : 10111111\ 01011001\ 10110110\ 10001100$

$P(B_7) : 01000011\ 10010100\ 11100111\ 00101110$

$P(B_8) : 10111000\ 01100100\ 01001111\ 01101010$

$P(B_9) : 01100010\ 10011011\ 11000010\ 01101010$

$P(B_{10}) : 10001000\ 00011011\ 10001100\ 10001111$

$P(B_{11}) : 01101010\ 10011101\ 00011011\ 10111000$

$P(B_{12}) : 10100101\ 11110000\ 11001010\ 10001110$

$P(B_{13}) : 01100111\ 10101101\ 00100101\ 00000101$

$P(B_{14}) : 10010110\ 00001001\ 00110011\ 10101101$

$P(B_{15}) : 00100000\ 00101100\ 01011110\ 10001000$

$P(B_{16}) : 00101011\ 00101011\ 00011001\ 11100010$

Hasil $P(B_i)$ kemudian di XOR dengan L_{i-1} untuk mendapatkan nilai R_i , sedangkan nilai L_i sendiri diperoleh dari nilai R_{i-1} untuk nilai $1 \leq i \leq 16$.

$L_0 : 11111111\ 10111000\ 01110110\ 01010111$

$R_0 : 00000000\ 11111111\ 00000110\ 10000011$

$P(B_1) = 11010101\ 01001010\ 01101101\ 11000000$

$L(1)-1 = 11111111\ 10111000\ 01110110\ 01010111$

-----XOR

$R_1 = 00101010\ 11110010\ 00011011\ 10010111$

$P(B_2) = 01010000\ 11111000\ 00101100\ 10001010$

$L(2)-1 = 00000000\ 11111111\ 00000110\ 10000011$

-----XOR

$R_2 = 01010000\ 00000111\ 00101010\ 00001001$

.....

P(B15) = 00100000 00101100 01011110 10001000

L(15)-1 = 11010100 01001011 01101101 11000010

-----XOR

R15 = 11110100 01100111 00110010 01001010

P(B16) = 00101011 00101011 00011001 11100010

L(16)-1 = 11011001 11111001 01011101 10100000

-----XOR

R16 = 11110010 01100111 00110011 01001010

Gabungkan R₁₆ dan L₁₆ kemudian permutasikan untuk terakhir kalinya dengan tabel invers inisial permutasi (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 17 IP⁻¹

Sehingga input :

R₁₆L₁₆ : 11110010 01100111 00110011 01001010 11110100 01100111 01001010

Menghasilkan output :

Chiper (dalam biner) = 00101000 01111011 10100100 00000010 11011000 11101000
11110111 11010000

Chiper (dalam hex) = 287BA402D8E8F7D0

Gunakan chiper (dalam hex) key1 sebagai plaintext dan dienkrpsi menggunakan key2 dengan cara yang sama dan hasil yang diperoleh dari enkripsi key1 sebagai plaintext dan key2 sebagai berikut :

Chiper 2(dalam biner) = 11011110 00001111 10111000 11000101 01001101 00011101
11010101 01111111

Chiper 2(dalam hex) = DE0FB8C54D1DD57F

Enkripsi terakhir yaitu mengenkripsi chiper 2(dalam hex) dengan kunci 3 yang sudah ditentukan yang memperoleh hasil seperti berikut :

Chiper 3(dalam biner) = 01011110 00100101 00010010 10000111 01111111 11000111
10100000 11011110

Chiper 3(dalam hex) = 5E2512877FC7A0DE

Hasil enkripsi yang diperoleh dari :

Plaintext : computer

Kunci 1 : belajarr

Kunci 2 : enkripsi

Kunci 3 : dekripsi

Mendapatkan hasil akhir enkripsi yaitu 5E2512877FC7A0DE.