**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 :

Studi Literatur

Analisis Kebutuhan

Skema & Desain

Implementasi

Pengujian

Hasil Analisa

Gambar 10Tahapan Penelitian

* 1. **Studi Literatur**

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

1. Kriptografi
2. DES
3. 3DES
4. VB.Net

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

* 1. **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 kunci56bit dari DES). Penggunaan tiga kunci tersebut penting untuk mencegah *meet-in-the-middle* *attacks*.

* + 1. **Spesifikasi Perangkat Keras**

3.2.1.1Personal Computer (PC) atau laptop

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

|  |  |  |
| --- | --- | --- |
| No | Faktor | Deskripsi |
| 1 | Prosesor | Intel® CoreTMi3-2310M CPU @ 2.10GHz |
| 2 | RAM | 4 GB DDR3 / 4096 MB RAM |
| 3 | HDD | 500 |
|  | | |

Tabel 1 Spesifikasi Laptop

* + 1. **Spesifikasi Perangkat Lunak**
       1. Microsoft .Net Framework

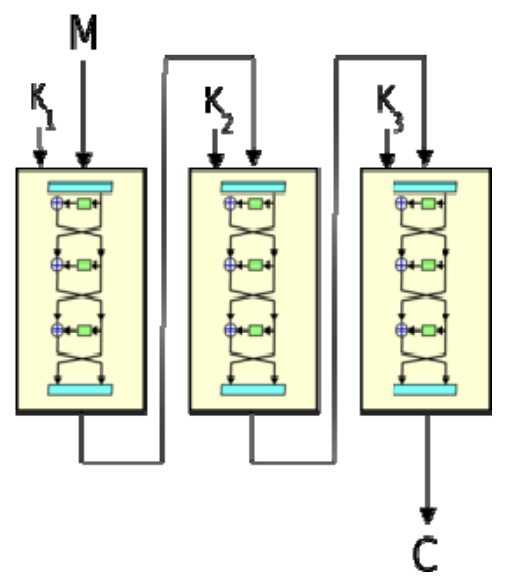
*Microsoft .NET Framework* (dibaca *Microsoft Dot Net Framework*) atau lebih dikenal dengan singkatan dot net (tidak berhubungan dengan domain [.net](https://id.wikipedia.org/wiki/.net" \o ".net)) merupakan sebuah [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) [kerangka kerja](https://id.wikipedia.org/wiki/Kerangka_kerja) yang berjalan utamanya pada [sistem operasi](https://id.wikipedia.org/wiki/Sistem_operasi) [Microsoft Windows](https://id.wikipedia.org/wiki/Microsoft_Windows), saat ini .NET Framework umumnya telah terintegrasi dalam distribusi standar Windows (mulai dari [Windows Server 2003](https://id.wikipedia.org/wiki/Windows_Server_2003" \o "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](https://id.wikipedia.org/wiki/Common_Language_Runtime" \o "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.

* + - 1. VB.Net

adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem [.NET Framework](https://id.wikipedia.org/wiki/.NET_Framework" \o ".NET Framework), dengan menggunakan [bahasa](https://id.wikipedia.org/wiki/Bahasa_pemrograman" \o "Bahasa pemrograman) [BASIC](https://id.wikipedia.org/wiki/BASIC). Dengan menggunakan alat ini, para *[programmer](https://id.wikipedia.org/wiki/Programmer" \o "Programmer)* dapat membangun aplikasi [Windows Forms](https://id.wikipedia.org/wiki/Windows_Forms), Aplikasi [web](https://id.wikipedia.org/wiki/Web) berbasis [ASP.NET](https://id.wikipedia.org/wiki/ASP.NET), dan juga aplikasi *[command-line](https://id.wikipedia.org/w/index.php?title=Command-line&action=edit&redlink=1" \o "Command-line (halaman belum tersedia))*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti [Microsoft Visual C++](https://id.wikipedia.org/w/index.php?title=Microsoft_Visual_C%2B%2B&action=edit&redlink=1" \o "Microsoft Visual C++ (halaman belum tersedia)), [Visual C#](https://id.wikipedia.org/wiki/Microsoft_Visual_C_Sharp), atau [Visual J#](https://id.wikipedia.org/wiki/Microsoft_Visual_J_Sharp" \o "Microsoft Visual J Sharp)), atau juga dapat diperoleh secara terpadu dalam [Microsoft Visual Studio .NET](https://id.wikipedia.org/w/index.php?title=Microsoft_Visual_Studio_.NET&action=edit&redlink=1" \o "Microsoft Visual Studio .NET (halaman belum tersedia)). Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari [Microsoft Visual Basic](https://id.wikipedia.org/wiki/Microsoft_Visual_Basic" \o "Microsoft Visual Basic) versi sebelumnya yang diimplementasikan di atas [.NET Framework](https://id.wikipedia.org/wiki/.NET_Framework" \o ".NET Framework). Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft, dan versi baru ini tidak kompatibel dengan versi terdahulu.

* 1. **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 :

* 1. K1, K2, K3 adalah kunci yang saling berbeda

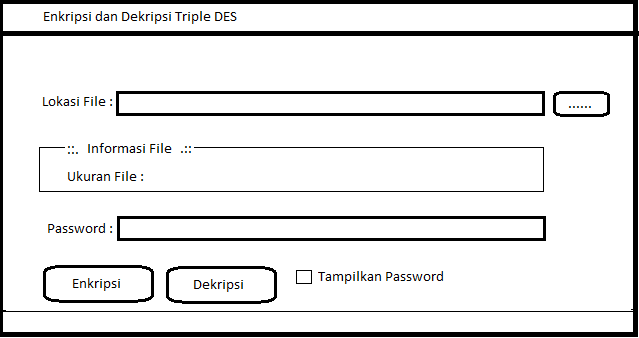
K1≠K2≠K3≠K1

* 1. K1 dan K2 adalah kunci yang berbeda, dan k3 sama dengan K1

K1≠K2 dan K3=K1

* + 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*
  1. **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.

* 1. **Pengujian**

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

* 1. **Hasil Analisa**

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

* 1. **Pembahasan Metode 3 DES**

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

Enkripsi : C = EK3(EK2(EK1(P)))

Dekripsi : P = DK1(DK2(DK3(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.

EnkripsiK3 64 bit

EnkripsiK2 64 bit

EnkripsiK1 64 bit

Ciphertext

Plaintext

Gambar 13 Enkripsi EEE 3DES

Sedangkan skema untuk dekripsinya digambarkan pada gambar 3.7.2 .

Ciphertext

DekripsiK3 64 bit

Plaintext

DekripsiK1 64 bit

DekripsiK2 64 bit

Gambar 14 Dekripsi EEE 3DES

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

Enkripsi : C = EK3(DK2(EK1(P)))

Dekripsi : P = DK1(EK2(DK3(C)))

Untuk skema enkripsi dan dekripsinya terlihat seperti gambar dibawah ini :

Plaintext

EnkripsiK3 64 bit

Ciphertext

DekripsiK2 64 bit

EnkripsiK1 64 bit

Gambar 15 Enkripsi EDE 3 DES

Chipertext

DekripsiK3 64 bit

Plaintext

DekripsiK1 64 bit

EnkripsiK2 64 bit

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 :

Enkripsi : C = EK1(DK2(EK1(P)))

Dekripsi : P = DK1(EK2(DK1(C)))

**3.7.1 Proses 3 DES**

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

Enkripsi : C = EK3(EK2(EK1(P)))

Dekripsi : P = DK1(DK2(DK3(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*1ke biner :

belajarr : 01100010 01100101 01101100 01100001 01101010 01100001 01110010 01110010

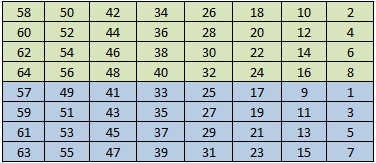
*Key*2ke biner :

enkripsi : 01100101 01101110 01101011 01110010 01101001 01110000 01110011 01101001

*Key*3ke biner :

dekripsi : 01100100 01100101 01101011 01110010 01101001 01110000 01110011 01101001

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



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

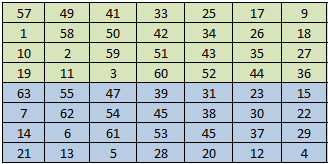
Ip(x) : 11111111 10111000 01110110 01010111 00000000 11111111 00000110 10000011

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.



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

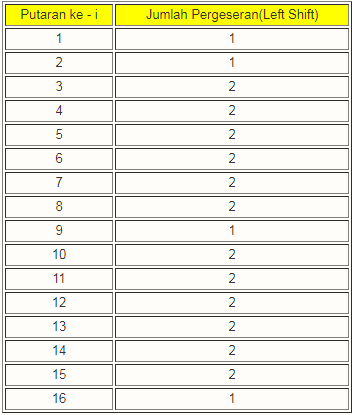
CD(k) : 0000000 1111111 1111111 1100110 0001000 0110000 0100000

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

C0 : 0000000 1111111 1111111 1100110 (tabel PC-1 warna hijau)

D0 : 1100110 0001000 0110000 0100000 (tabel PC-1 warna biru)

Lakukan pergeseran kiri (*Left shift)* pada C0 D0, sebanyak 1 atau 2 kali berdasarkan kali putaran yang ada pada tabel putaran sebagai berikut :



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* ;

C0 : 0000000 1111111 1111111 1100110

D0 : 1100110 0001000 0110000 0100000

Digeser 1 bit ke kiri

C1 : 0000001 1111111 1111111 1001100

D1 : 1001100 0010000 1100000 1000001

Digeser 1 bit ke kiri

C2 : 0000011 1111111 1111111 0011000

D2 : 0011000 0100001 1000001 0000011

………..

Digeser 2 bit ke kiri

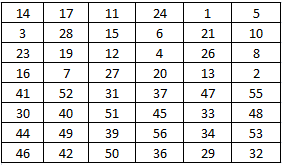
C14 : 1100000 0001111 1111111 1111100

D14 : 0001100 1100001 0000110 0000100

Digeser 2 bit ke kiri

C15 : 0000000 0111111 1111111 1110011

D15 : 0110011 0000100 0011000 0010000

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

Tabel 5 PC-2

C­1D1 : 00000001 1111111 11111111 11000101 000100000 11000010 10000001

K1 : 11110000 10111110 01100110 10101000 00000101 00010010

C­2D2 : 00000011 11111111 11111111 00000100 01000001 10000101 00000011

K2 : 11110000 10111110 01110110 10000111 00010001 10100000

……

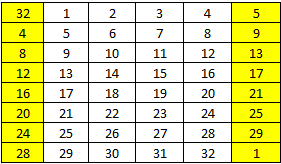
C­15D15 : 00000000 01111111 11111111 11100110 10001000 00110000 10100000

K15 : 111100 011011 111000 101110 011000 110100 000010 100000

C­16D16: 00000000 11111111 11111111 11001101 00010000 01100001 01000000

K16 : 11110000 10111110 10100110 00000010 00010000 00011111

Pada langkah berikutnya, kita akan meng-ekspansi data Ri-1 32 bit menjadi Ri 48 bit sebanyak 16 kali putaran dengan nilai perputaran 1<= i <=16 menggunakan Tabel Ekspansi (E).



Tabel 6 Ekspansi (E)

Hasil E(Ri-1) kemudian di XOR dengan Ki dan menghasilkan Vektor Matriks Ai.

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

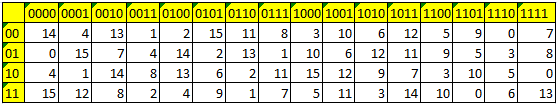
……

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

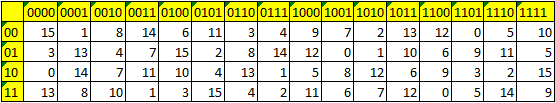
Langkah selanjutnya Setiap Vektor Ai disubstitusikan kedelapan buah S-Box (Substitution Box), dimana blok pertama disubstitusikan dengan S1, blok kedua dengan S2 dan seterusnya dan menghasilkan output vektor Bi32 bit.

S1



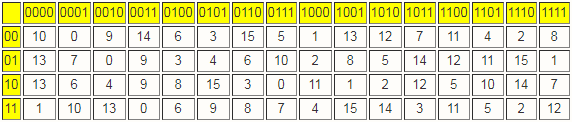
Tabel 7 S-BOX 1

S2



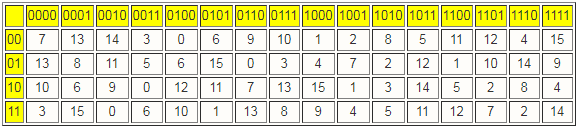
Tabel 8 S-BOX 2

S3



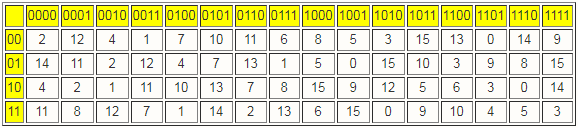
Tabel 9 S-BOX 3

S4



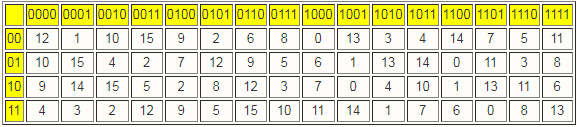
Tabel 10 S-BOX 4

S5



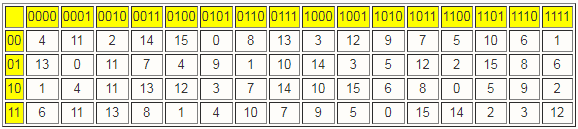
Tabel 11 S-BOX 5

S6



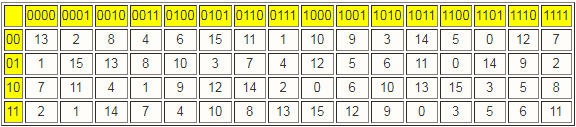
Tabel 12 S-BOX 6

S7



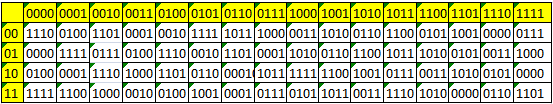
Tabel 13 S-BOX 7

S8



Tabel 14 S-BOX 8

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



Tabel 15 S-BOX 1 biner

Menghitung A1 dengan S-BOX dengan cara :

A1 : 01110000 10101001 10011000 00101000 11010001 00010100

Ambil 6 bit dari A1 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 :

B1 : 00001011 10011011 10101001 00100011

B2 : 10011010 00000010 00101111 01100001

B3 : 01100010 11000101 11000011 01111100

B4 : 01110001 11011100 10111011 10000101

B5 : 11100001 01100100 10101110 01001011

B6 : 01101000 01110111 10111101 01111000

B7 : 11110011 10100000 11000101 01110100

B8 : 00110001 10001011 01011110 00101101

B9 : 11111011 01001000 00010100 01010110

B10 : 01011000 01100001 00100100 11101011

B11 : 10101110 11000100 01110100 01011101

B12 : 11110001 00110011 10110110 01000001

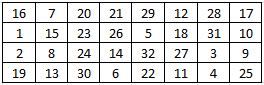
B13 : 10001010 11110000 11010011 10110000

B14 : 00101000 11110101 00101101 10010100

B15 : 00101001 00000100 01110110 00100001

B16 : 00011000 11001100 10110010 00011111

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



Tabel 16 P-BOX

Sehingga hasil yang diperoleh seperti berikut ini :

P(B1) : 11010101 01001010 011011011 1000000

P(B2) : 01010000 11111000 00101100 10001010

P(B3) : 11001011 00110101 10110111 00100000

P(B4) : 00110101 00100001 11111011 11100011

P(B5) : 00011001 10110011 11011010 10001100

P(B6) : 10111111 01011001 10110110 10001100

P(B7) : 01000011 10010100 11100111 00101110

P(B8) : 10111000 01100100 01001111 01101010

P(B9) : 01100010 10011011 11000010 01101010

P(B10) : 10001000 00011011 10001100 10001111

P(B11) : 01101010 10011101 00011011 10111000

P(B12) : 10100101 11110000 11001010 10001110

P(B13) : 01100111 10101101 00100101 00000101

P(B14) : 10010110 00001001 00110011 10101101

P(B15) : 00100000 00101100 01011110 10001000

P(B16) : 00101011 00101011 00011001 11100010

Hasil P(Bi) kemudian di XOR dengan Li-1 untuk mendapatkan nilai Ri, sedangkan nilai Li sendiri diperoleh dari nilai Ri-1 untuk nilai 1<=i<=16.

L0 : 11111111 10111000 01110110 01010111

R0 : 00000000 11111111 00000110 10000011

P(B1)      = 11010101 01001010 01101101 11000000  
L(1)-1    = 11111111 10111000 01110110 01010111  
------------------------------------------------------------------------------------------XOR  
R1    = 00101010 11110010 00011011 10010111

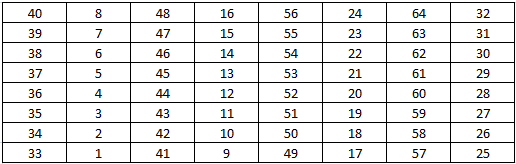
P(B2)      = 01010000 11111000 00101100 10001010  
L(2)-1    = 00000000 11111111 00000110 10000011  
------------------------------------------------------------------------------------------XOR  
R2    = 01010000 00000111 00101010 00001001

…………

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)



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

Gunakakan chiper (dalam hex) key1 sebagai plaintext dan dienkripsi 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.